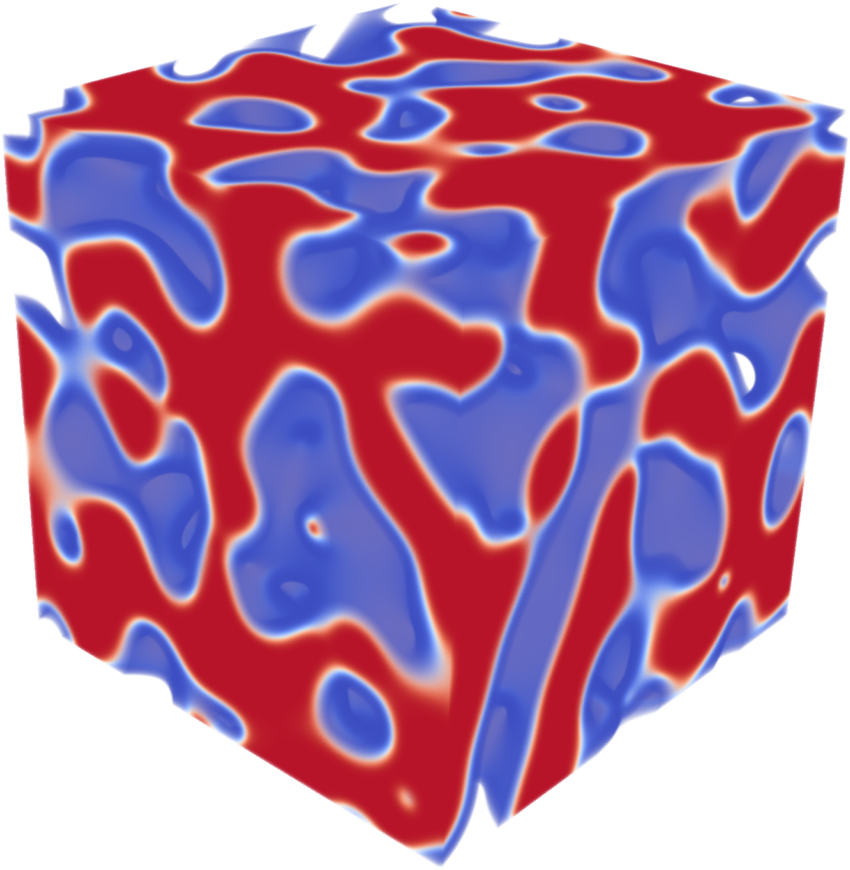


Simulation and analysis of spinodal decomposition using the Allen-Cahn equation

Master's thesis in Complex Adaptive Systems

WILLIAM WEST

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Simulation and analysis of spinodal decomposition using the Allen-Cahn equation

WILLIAM WEST

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Applied Mathematics and Statistics
Chalmers University of Technology
Gothenburg, Sweden 2023

Simulation and analysis of spinodal decomposition using the Allen-Cahn equation
WILLIAM WEST

Supervisor: Tobias Gebäck, Department of Mathematical Sciences
Examiner: Tobias Gebäck, Department of Mathematical Sciences

Master's Thesis 2023
Department of Mathematical Sciences
Applied Mathematics and Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Final moment from a simulation of spinodal decomposition in a box environment.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Simulation and analysis of spinodal decomposition using the Allen-Cahn equation
WILLIAM WEST
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Spinodal decomposition is a type of phase separation that can occur for certain compositions of mixtures. If one of the phases is then resolved, the result is a labyrinthine structure that can advantageously be used to transport material, such as medicine through tablets. Simulations of spinodal decomposition based on the Allen-Cahn equation were run with a variety of parameters to gather time-based data. Methods were then developed to analyse characteristics from the data, both in 2D and 3D. The results were compared to those from previous physical experiments, with some accuracy. Finally, the effects of boundary conditions on spatial variations and phase patterns were studied.

Keywords: Spinodal decomposition, phase separation, Allen-Cahn, simulation, modelling, characteristic length, mean curvature, drug release.

Acknowledgements

I want to thank my supervisor Tobias Gebäck for his invaluable guidance throughout the thesis work, which withouth it would not have been possible. I also want to thank Pierre Carmona and Alexander Rodin for laying the theoretical and practical groundwork for me to be able to build upon. Their experiments, simulations and code have been most useful. Finally, I will give big thanks to my supportive family, girlfriend, and friends.

William West, Gothenburg, June 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AC	Allen-Cahn
DFT	Discrete Fourier transform
EC	Ethylcellulose
FOV	Field of view
HPC	Hydroxypropylcellulose
LBM	Lattice Boltzmann method
Sc	Schmidt number
VTK	Visualisation Toolkit

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

In medicine, delivering drugs to a patient is often done in the form of a tablet. This is done to achieve both regularity of drug intake and control over the amount of drug released over time, the drug release rate. In some cases these tablets contain pellets that have the structure of a core made of microcrystalline cellulose surrounded by a layer of the drug intended for release. This drug-layer is in turn surrounded by an outer coating of a mixture of two types of cellulose and ethanol. One of these celluloses is hydrophilic and the other one is hydrophobic, for example hydroxypropylcellulose (HPC) and ethylcellulose (EC) respectively. The ethanol is used as a solvent to make the celluloses able to mix properly [1].

The purpose of this outer mixture is to deliver the drug inside at a controlled rate. This is accomplished in two main steps; the first is that the mixture undergoes spinodal decomposition, the mixture phase separates into a bicontinuous structure as the ethanol evaporates and this process halts when the ethanol is gone as the viscosity of the mixture left becomes too high for phase separation to spontaneously occur [3]. The second step happens when the tablet is inside the body, when the coating of the tablet comes into contact with water in the body the hydrophilic part of the coating is leached out and only the hydrophobic part remains. This creates a labyrinthine structure through which the drug in the inner layer can pass through [1]. It is desirable to achieve control over this microstructure by tuning variables and parameters in the creation process.

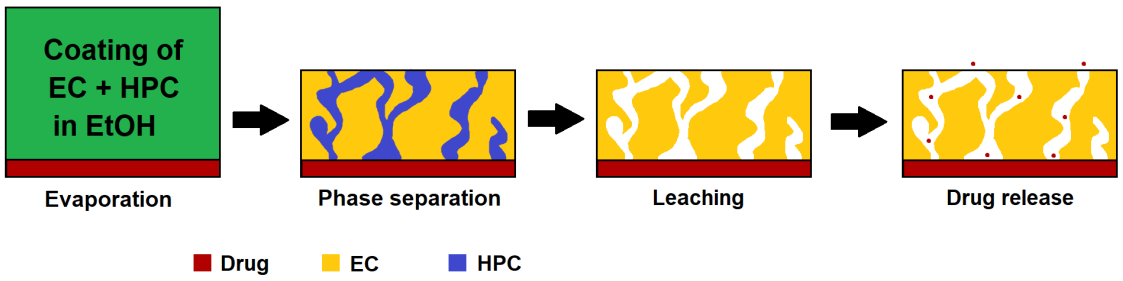


Figure 1.1: The mechanics of tablet drug release with the initial coating being made of a mixture of EC and HPC solved in ethanol which evaporates during phase separation. The HPC is leached out leaving paths through which the drug is released.

Designing the microstructures allows for a more stable release of the drug than

when compared to an immediate release system where the drug level in the body can vary wildly. For some drugs the drug level in the body needs to be stable and at approximately the same level over a longer period of time in order to make sure the drug level is not too low nor too high. Were it too low the drug would not be effective or not work at all and were the level too high there could be serious unwanted side effects [4], figure 1.2 showcases this. Keeping the drug level at the desired level could then be achieved by designing the microstructure in such a way that the drug passes from the inner layer to the body through the tunnels at a certain rate [5].

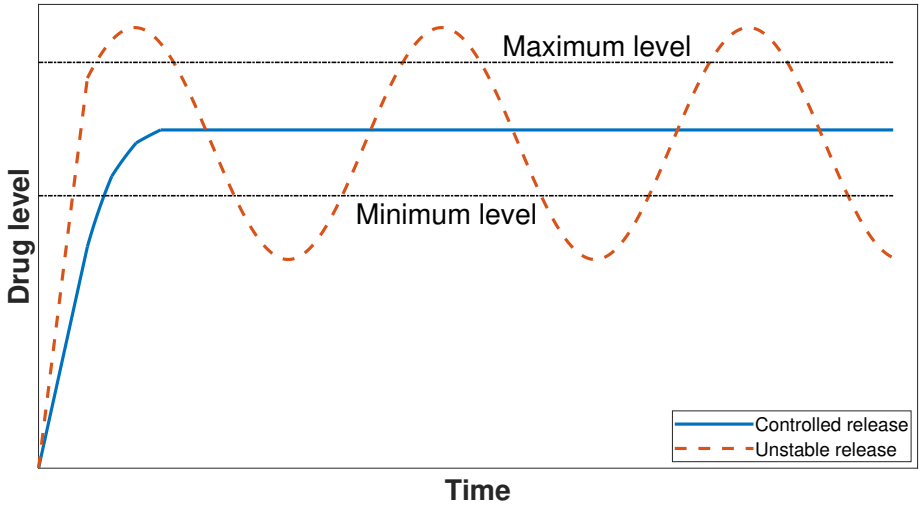


Figure 1.2: An example comparison between controlled release and repeated unstable release. The controlled release keeps the drug level in the body between the minimum and maximum levels while the unstable release sometimes brings the level too high or too low with repeated doses.

Different initial compositions of HPC and EC solved in ethanol gives rise to different types of microstructures when the 1-phase (1ϕ) transitions to 2-phase (2ϕ) [1]. This can happen through two different mechanisms; nucleation and growth, and spinodal decomposition [1]. Nucleation and growth is a two step process wherein the first step tiny droplets are formed which then act as nuclei for growth in the second step [6]. This eventually results in a discontinuous structure of irregularly sized beads of the less plenty phase surrounded by the more common phase [1]. The areas marked by a and e in the phase diagram in figure 1.3 are dominated by nucleation and growth.

Inside the spinodal curve, the dominating mechanism for phase separation is spinodal decomposition, during which small fluctuations in concentration lowers the free energy which leads to an amplification of the small changes in composition [3][7]. This process can create bicontinuous and labyrinthine structures but also discontinuous ones like shown in figure 1.3 [1].

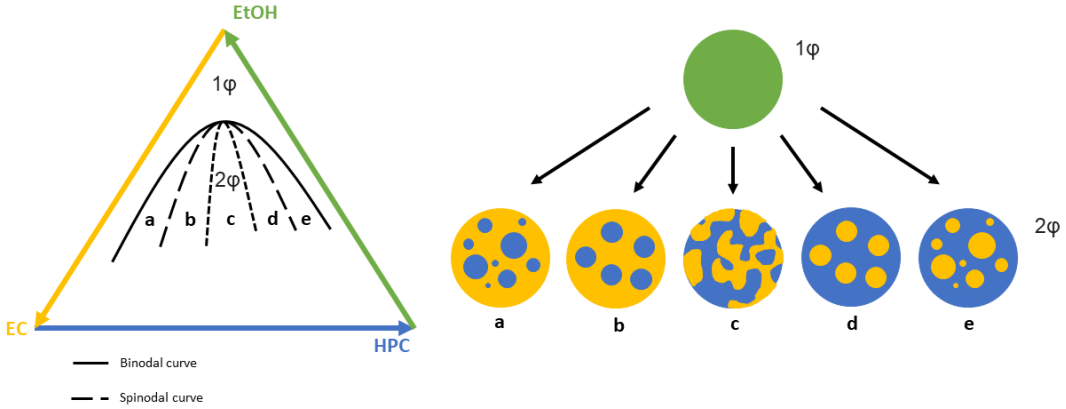


Figure 1.3: Ternary phase diagram of EC and HPC solved in ethanol showing phase separation and the possible resulting structures. Regions *a* and *e* are dominated by nucleation and growth while in regions *b*, *c* and *d*, spinodal decomposition is the main mechanism.

Traditionally, the Cahn-Hilliard equation is used to model multiphase flows but the Allen-Cahn equation, while less popular, also works well [8]. Because they are hard to solve, these equations are often solved numerically and they can be used together with a lattice-Boltzmann equation to computationally simulate the spinodal decomposition mechanism [8]. An aim of this thesis is to simulate spinodal decomposition specifically using the Allen-Cahn equation and different parameters to explore and gain a deeper understanding of the underlying mechanics.

Two works lay the groundwork for this thesis work. The first one is Pierre Carmona's Licentiate thesis where he sets up solutions of EC and HPC and studies the following phase separation and coarsening mechanisms using confocal laser scanning microscopy and image analysis techniques [1]. The results obtained by Carmona were then used to support Alexander Rodin's Master's thesis where he simulates spinodal decomposition phase separation using both the Cahn-Hilliard and the Allen-Cahn equations [3]. Rodin found that the AC models agreed better with Carmona's experiment data [3]. This thesis is based on both these previous works and continues to explore the mechanics and workings of phase separation through spinodal decomposition.

1.2 Aims

This work aims to examine spinodal decomposition further by simulating the behaviour on a powerful CPU-cluster and then analysing the resulting data. The goals in particular are listed below:

- Develop and apply methods to measure the length scale and curvature of the evolving patterns of both 2D and 3D data.
- Compare the measures of length scale and curvature from 3D structures to the 2D image data previously received from experiments.

- Study the effect of boundary conditions at the glass surface (bottom of the domain) and thickness of the mixture on the decomposition, especially regarding spatial variation and phase patterns depending on a variety of parameters.

2
Theory

2.1 Allen-Cahn equation

The Allen-Cahn equation is an equation that is used to model phase separation and ensures that the total free energy decreases over time [3] [9]. The code for the simulations uses a modified, conservative phase field version of the Allen-Cahn equation that has been changed from the original non-conservative model to achieve higher accuracy [10].

First, the general interface tracking phase field version was derived by rewriting the original Allen-Cahn equation using the free energy functional, a double-well function and a interpolating function [9]. By choosing appropriate forms for these functions and assuming no external advection, a general interface tracking phase field version of the Allen-Cahn equation is achieved [9].

Then, the mass conservative phase field version was developed by using the previously developed general phase field version and the same kernel functions as in the conservative level set method [11]. Assuming incompressible flow, a divergence free condition can be imposed and after reformulating, the conservative version was developed [11].

The version of conservative Allen-Cahn equation used in the code is the following one:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \boldsymbol{u}) = \nabla \cdot \left(M \left(\nabla \phi - \frac{4}{\xi} \phi (1 - \phi) \frac{\nabla \phi}{|\nabla \phi|} \right) \right) \tag{2.1}$$

Here, ϕ is the concentration of one of the phases in a two-phase solution at any given point. It has here been defined that ϕ in the equation is the second phase and set up such that the two phases sum to 1. Furthermore, $0 \leq \phi \leq 1$ where $\phi = 0$ means that only the first phase is present and $\phi = 1$ means that none of the first phase is present and instead, only the second phase is present.

In addition, $\boldsymbol{u}$ represents the velocity of the fluid at all points. M is the mobility constant to regulate how fast the liquid can move. Lastly, ξ determines the sharpness of the phase field by controlling the width of the diffusive interface like shown in figure 2.1

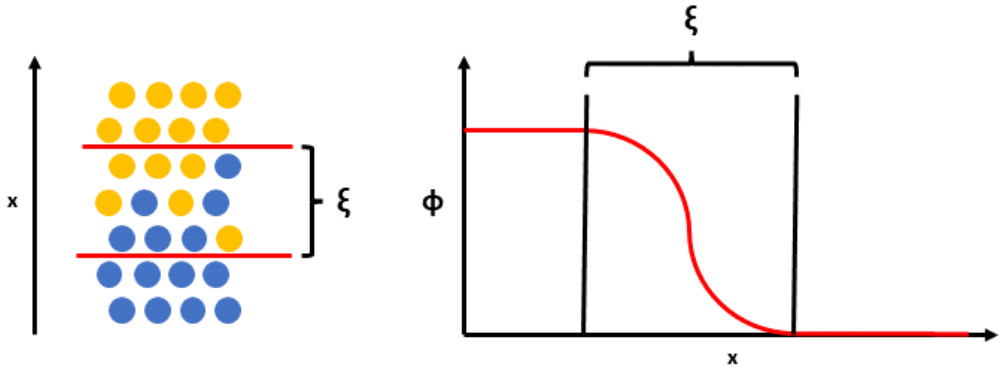


Figure 2.1: ξ is the width of the diffusive interface.

Further, $\boldsymbol{u}$ is obtained by solving the Navier-Stokes equations for conservation of mass and multiphase flows [12]. The conservation of mass equation is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \boldsymbol{u} = 0 \tag{2.2}$$

and the multiphase flow equation is:

$$\rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) = -\nabla p + \nabla \cdot [\mu(\nabla \boldsymbol{u} + \boldsymbol{u} \nabla)] + \boldsymbol{F}_s + \boldsymbol{F}_b \tag{2.3}$$

Here, ρ is the density, p is the pressure, μ is the dynamic viscosity, $\boldsymbol{F}_s$ is the surface tension and $\boldsymbol{F}_b$ is the body force.

2.2 Growth of patterns during phase separation

The phase separation mechanism of spinodal decomposition explored in the simulations makes a solution of liquids quickly unmix, transforming one homogeneous phase into two separate phases [1]. Spinodal decomposition is typically initiated by random fluctuations of concentration in the solution which lowers the free energy of mixing [7].

There are three stages that spinodal growth goes through. In the first stage, early in the process, the aforementioned fluctuations make the concentration differences grow at different speeds because they operate at different length scales themselves [13]. However, the concentration fluctuations with medium sized wavelengths dominate over the shorter and longer. This happens because the shorter wavelengths are suboptimal energy-wise and the longer wavelengths cause the associated fluctuations to grow slower than the medium ones [1].

During the intermediate stage, the amplitude of the concentration fluctuations reaches its maximum and the width of the diffusive interface decreases. Finally, during the late stage, equilibrium is reached and self-similar growth becomes the driving factor [1].

Like shown in figure 1.3, the composition of the mixture determines the outcome pattern which can either be bicontinuous or isolated droplets floating in the dominating phase. These outcomes have coarsening regimes associated with them, with bicontinuous structures appearing as a result of a hydrodynamic regime and discontinuous structures driven by a diffusive regime [14].

The hydrodynamic regime causes growth of structures through differences in Laplace pressure [15] which in turn is partially a result of the product between the surface tension and the curvature; $p_{Laplace} \propto \gamma/R$. The pressure is higher in the thinner arms and thus material flows inwards, growing the structures. The diffusive regime is instead driven by coalescence, causing smaller droplets to combine into larger ones [1]. The meeting of droplets is facilitated by Brownian motion [14].

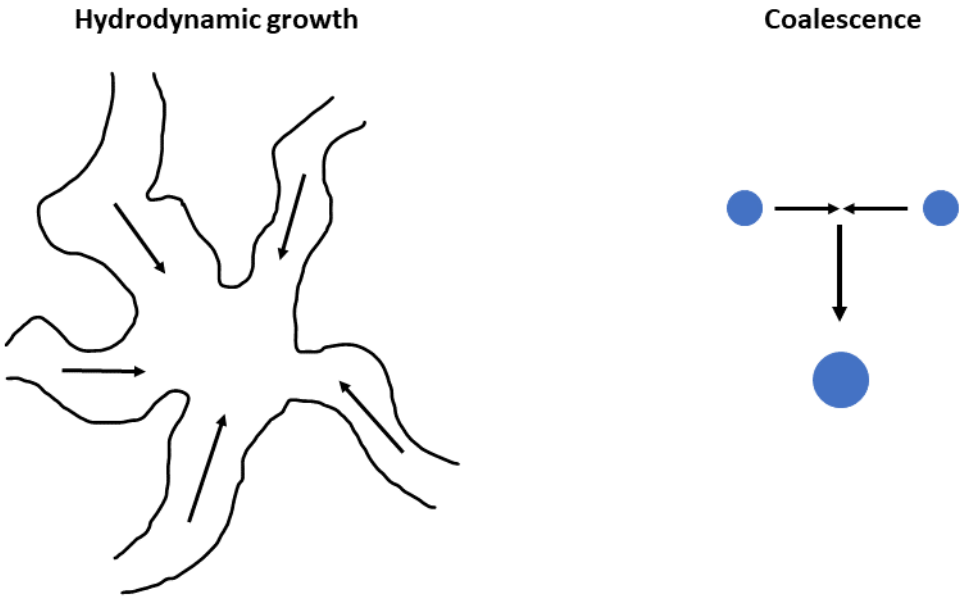


Figure 2.2: Figure showing two coarsening mechanisms, hydrodynamic growth and coalescence [1].

The characteristic length scale $L(t)$ of the forming patterns grow over time but at different speeds depending on the active regime. The hydrodynamic regime allows for faster, linear growth of the characteristic length scale while the coarsening for the diffusive regime grows slower [14].

$$L(t) \propto \begin{cases} t^{\frac{1}{3}} & \text{Diffusive regime} \\ t^{1} & \text{Hydrodynamic regime} \end{cases}$$

2.3 Lattice Boltzmann method

The lattice Boltzmann methods (LBM) are a collection of techniques used to simulate the flow of liquids. These techniques were inspired by lattice gas models and

developed these further, bringing with them their strengths and at the same time smoothing out their shortcomings [2]. More specifically, LBM solves the Navier-Stokes equations numerically at a set of discrete points to emulate a variety of systems such as turbulent flows, electromagnetic wave propagation and phase separation [2]. The code used for the simulations in this work uses LBM.

The LBM does not track every individual atom and molecule. Instead, it calculates distributions of some wanted quantities, the most central of which is the distribution function $f_i(\boldsymbol{x}, t)$. This function tracks the density of particles that have discrete velocities $\boldsymbol{c_i}$ at all positions $\boldsymbol{x}$ at time t [2]. From the distribution function, the mass density and momentum density can be found like so:

$$\rho(\boldsymbol{x}, t) = \sum_i f_i(\boldsymbol{x}, t), \qquad \rho\boldsymbol{u}(\boldsymbol{x}, t) = \sum_i \boldsymbol{c}_i f_i(\boldsymbol{x}, t) \tag{2.4}$$

When setting up an LBM environment it is possible to customise how many directions travel and interaction will be possible along. Fewer lanes means faster computations and a lower memory load but also lower simulation accuracy [2]. Specific setups are referred to as DdQq where d refers to the number of dimensions and q refers to the number of lanes connecting the nodes [2]. In figure 2.3, D2Q9 is shown as an example. Note that the stationary path, the self path to the current node, counts as a path too. As a compromise between performance and accuracy, the most used setup in 3D is D3Q19 [2].

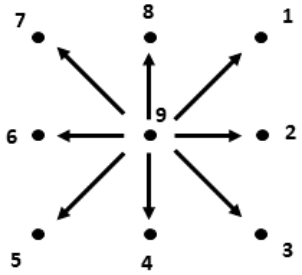


Figure 2.3: An LBM setup of D2Q9, meaning: 2 dimensions and 9 lanes of interaction.

In every iteration of LBM, the particle densities are relaxed towards an equilibrium f_i^{eq} and the speed at which this happens at is controlled by the relaxation time τ [2]. This equilibrium equation follows as:

$$f_i^{eq}(\boldsymbol{x}, t) = w_i \rho \left(1 + \frac{\boldsymbol{u} \cdot \boldsymbol{c}_i}{c_s^2} + \frac{(\boldsymbol{u} \cdot \boldsymbol{c}_i)^2}{2c_s^4} + \frac{\boldsymbol{u} \cdot \boldsymbol{u}}{2c_s^2}\right) \tag{2.5}$$

where w_i are weights associated with the specific lane setup and c_s is a constant, the speed of sound in the modelled environment, $c_s^2 = \Delta x^2 / 3\Delta t^2$ [2].

Then, there are two more important steps. They are the collision and propagation steps [2]. In the collision step, the particle densities are redistributed as collisions occur between the different grid points and the distribution function is here updated to be:

$$f_i^*(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t)\left(1 - \frac{\Delta t}{\tau}\right) + f_i^{eq}(\boldsymbol{x}, t)\frac{\Delta t}{\tau} \tag{2.6}$$

with f_i^* being the updated distribution function [2]. In the propagation step, also called the streaming step, particles are moved from every node to the surrounding nodes and a certain amount of particles stream back. This is represented as:

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i\Delta t, t + \Delta t) = f_i^*(\boldsymbol{x}, t) \tag{2.7}$$

One iteration of LBM is shown below in figure 2.4.

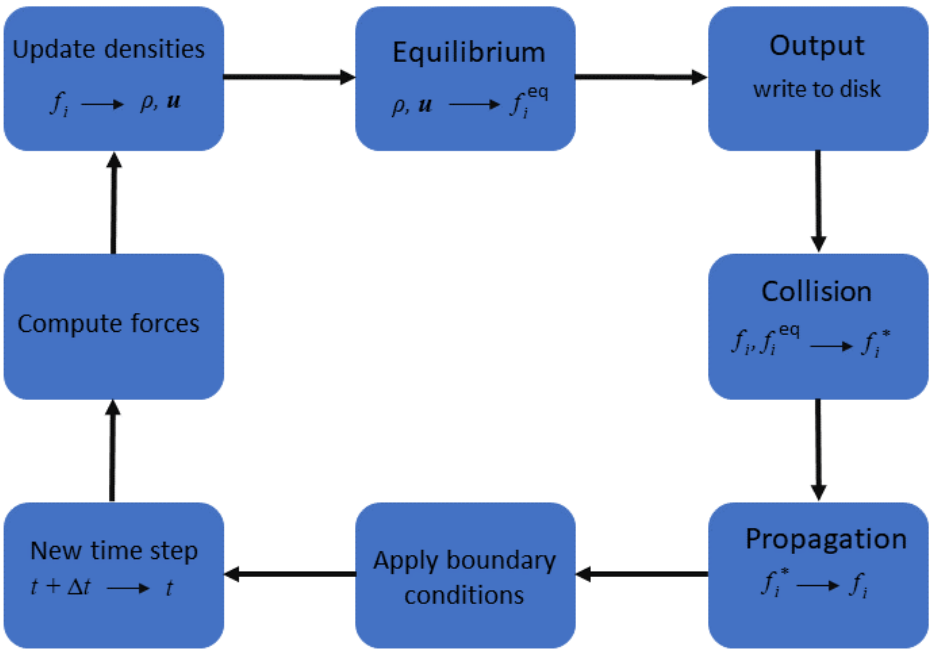


Figure 2.4: One iteration cycle of LBM, writing output to disk is optional and is generally not done for every iteration [2].

It is mainly ρ and $\boldsymbol{u}$ that is written to disk in the output step and rarely the populations f_i. One iteration represents one time step and this is repeated over and over to progress the simulation. In addition, boundary conditions are needed but are not discussed here. An extension of the scheme presented here may be used to simulate coupled Allen-Cahn and Navier-Stokes equations [12].

2.4 Schmidt number

Dimensionless numbers play an important part in modelling. Creating simulations where the dimensionless numbers match those of a real situation can be a good way of investigating the behaviour of a model without needing to build the potentially expensive and time consuming real product. The Schmidt number is a dimensionless number that describes the transport capabilities of a liquid. The number is defined as

$$Sc = \frac{\nu}{D} \tag{2.8}$$

where ν is the viscosity and D is the diffusion coefficient [16]. This ratio measures significance of momentum against mass diffusion in a liquid's velocity and concentration boundary layers [16]. When $Sc \approx 1$, neither momentum or mass diffusion dominate over the other and are comparable [16].

2.5 Discrete Fourier transform to determine characteristic length scale

In order to quickly calculate the characteristic length scale of the emerging patterns in the coarsening mixtures, the discrete Fourier transform can be used to transform the spatial 2D and 3D-concentration data into the frequency domain [17]. The characteristic length scale can then be extracted as the peak in the power spectrum [17].

The 2D-version of the DFT is presented here for a domain of size $N \times N$ data points:

$$F(k_1, k_2) = \frac{1}{N} \sum_{r_1=0}^{N-1} \sum_{r_2=0}^{N-1} f(r_1, r_2) e^{-i\frac{2\pi}{N}(k_1 r_1 + k_2 r_2)} \tag{2.9}$$

where (r_1, r_2) are the coordinates in real space and (k_1, k_2) are the coordinates in reciprocal space [18]. Then the power spectrum is found by $P(k_1, k_2) = |F(k_1, k_2)|^2$ [1].

Once the data is transformed, the radius is retrieved from $k = \sqrt{k_1^2 + k_2^2}$ [18]. In 2D, the characteristic length found with $L = \frac{\text{FOV}}{\mu}$ where μ is the wavenumber tied to the peak of the power spectrum given by the dominant radius k_{max} [1], and FOV is the field of view calculated as $\text{FOV} = N\Delta x$ with Δx being the physical distance between the data points. In 3D, the characteristic length is instead found through $L = \frac{N}{\mu}$.

2.6 Curvature

As spinodal decomposition occurs and the structures in the biphase liquid grow larger, the twists and turns grow smoother. To study how the mean curvature evolves over time, the data is measured for every output. The meaning of curvature

at any one given point is that the line the point sits on has a local curvature of $1/R$ where R is the radius of the circle that forms if the curvature is continuous [3].

It is specifically the curvature of the interface between the two phases that is interesting. This interface forms lines for which the mean curvature at every point is calculated by taking the divergence of the normal [19]:

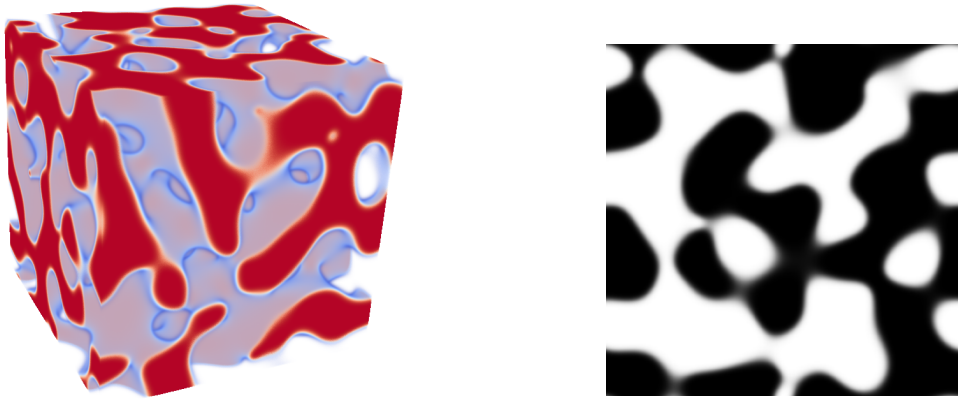
$$H = -\nabla \cdot \left(\frac{-\nabla \phi}{|\nabla \phi|} \right) \tag{2.10}$$

where ϕ is the phase concentration. This concept is general and works for both 2D and 3D-cases [19]. In the 3D-case however, H is the sum of the two principal curvatures and is divided by 2 to find the mean curvature.

3
Methods

3.1 Simulations

The simulations were run using pre-existing code on Chalmer's centre for high-performance computing, C3SE. The code itself consisted of an implementation of the Allen-Cahn equation in combination with an LBM algorithm, for which theory is presented in sections 2.1 and 2.3. As the simulations were running, VTK-files were output at regular intervals containing all variables at all points, but only the phase concentration was used. The data in these VTK-files could then be analysed in MATLAB and the concentration could be inspected in the visualisation program ParaView. Figure 3.1 illustrates an example situation from the unbounded environment after a full 30000 iterations simulating a low viscosity fluid.



(a) A 3D-view of the periodic cube environment.

(b) A 2D-slice of the same cube.

Figure 3.1: An example final situation after 30000 iterations. To the left, one phase is opaque red and the other has been set to invisible, to better show the types of tunnels that form.

3.1.1 Set up

In [3], the largest environments used for the simulations were boxes of size $200 \times 200 \times 200$ voxels. To allow the phase structures to grow for longer without their size starting to become comparable to the box size, cubes of side length 300 voxels were

studied. This larger environment was only feasible with the use of the CPU cluster. Also, this cube environment is periodic, there are no solid walls and the medium can freely loop around.

Additionally, two more environments were used. These environments differed from the first one in that they were not fully periodic. Instead, they have a free boundary condition at the top to mimic open air and a closed boundary condition at the bottom to imitate the glass containers in the physical experiments [1]. The difference between these two environments is that one has a height of 100 voxels and the other one has a height of 50. Both of them still have a length and width of 300 making them shorter than wide and long.

All simulations were run for 30000 iterations to allow the phase separation to start stagnating. Data storage became a factor when determining how often outputs would be written to disk. For the larger, unbounded environment output was given every 2000 iterations for a total of 15 outputs. The simulations in the smaller, bounded environments instead gave outputs every 1000 iterations as the smaller files allowed the freedom to have more of them for a higher resolution in time. But for the alternate contact angles (see 3.1.2) in the bounded environment the output was again for every 2000 iterations as this significantly sped up processing the data.

There is no inherent sense of time in the simulations so the time scale must be manually matched to previous results. With this in mind, the time scale was set up such that 500 iterations is equal to $6.667s$ which is in line with both previous simulations and experiments [1] [3]. This means that in total, the phase separation goes on for $400s$. Finding the length scale was done by comparing the experiments to the simulations. The first visible shapes that form after the initial noise goes away have a diameter of about 0.2 µm [1]. By running a test simulation with frequent outputs it was possible to count pixels and find a conversion factor between voxels and µm. The diameters in the simulation were found to be 2-3 pixels wide or 2.5 pixels wide on average. This means that 0.2 µm = 2.5 voxels and so 1 µm = 12.5 voxels or alternatively, 1 voxel = 0.08 µm. Finally, the phase field was initialised with a uniform random distribution centred on $\phi = 0.5$ with a width of 1, meaning a full range over the concentration.

3.1.2 Parameters

The input parameters that were used for the simulations are presented here, many of them are the same as in [3]. The viscosity ratio for the two phases was set to 3 for all simulations, but two different levels of viscosity were explored. In the result section, these viscosities are called high and low viscosity and the relaxation parameter of the light phase was set to 1.0 and 0.55 respectively. Furthermore, the density ratio between the phases was set to 1 during all simulations. The surface tension was $\sigma = 0.001$, the mobility was $M = 0.4$, the interface width was $\xi = 5$ and the relaxation parameter of the phase field was $\tau = 1.7$.

To study how the contact angle affected the results in the bounded environment, three of them were tried. This angle is the contact angle at equilibrium, meaning

the angle that is formed between the two phases and the floor. The lower phase ends up undermining the other one and creates a semi-covering layer depending on what degree the angle is set to. The default was set to 90° and this is the value used when no angle is mentioned but 70° and 50° were also studied.

3.2 Analysis methods

The processes for analysing the results in MATLAB are explained below. The code for analysing the characteristic length scale for 2D-data was provided. This code was extended to handle 3D-data in this work. The code for analysing the growth exponents, the average curvature and the average concentration was developed.

3.2.1 Length scale

The code for calculating the characteristic length scale of 2D-slices was provided, it is the same code as in [1]. It was then altered to also be able to measure the length scale in 3D. The length scale was acquired for every output to study the lengthening of the structures over time. In the unbounded 2D-case, the length result from many slices were combined to an average for every output with the idea that this would give a more accurate result. This was not done for the other cases. In the bounded case with alternate contact angles, the slice for which the length scale was calculated was taken from low in the environment with the idea that this would better display changes due to the contact angle. The field of view (FOV) was needed for the 2D-code to function and this can be calculated by $FOV = N\Delta x = 300 \cdot 0.08 = 24$µm.

A radial power spectrum was attained through the DFT and then squaring the absolute value of the result. The wavenumber and radius with the highest intensity can then be found. The length scales are then calculated like described in section 2.5. Figure 3.2 shows the 2D power spectrum and the radial distribution for 2D and 3D. Note that the dominant radius for 3D is shorter which when transformed back into normal space suggests that the length scale for 3D is longer.

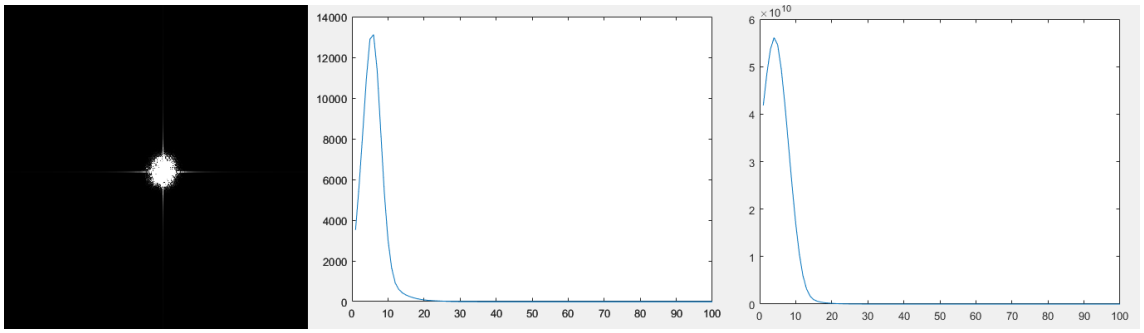


Figure 3.2: Left: 2D power spectrum. Middle: 2D radial distribution with the top in the sixth container. Right: 3D radial distribution with the top in the fifth container.

3.2.2 Growth exponent

The growth exponents, which reveal information about the behaviour of the growth in the system, were found by curve fitting. The data used was the calculated length scales over time. In theory, $L(t) \propto t^n$ so the function $L(t) = at^n + b$ was used to fit the length scale data against. The MATLAB function *lsqcurvefit* was used with restrictions put on a, n and b to force them to be positive and finite, $0 < a, n, b < \infty$.

This way, the parameters a, n and b were found along with the associated R^2-values for the fittings for both the 2D and 3D-length scale data. The growth exponents n and their R^2-values are presented in results chapter.

3.2.3 Average curvature

To find the average curvature of phase concentration in a 2D-slice, the curvature was first calculated for every point along the phase interface and then the average of these points was taken. The points on the interface were found by omitting all points that had a concentration too far from 0.5, an example of this is shown in figure 3.3. Thus the only points for which the curvature was calculated was ones that had a concentration in the range $[0.5 - x, 0.5 + x]$ for some value of x. Different values for x were tried and it was found that a larger value, like $x = 0.4$, was better since the interface line needed to be thicker for the calculations to work. The actual calculation was taking the divergence of the normalised gradient of the concentration like described in section 2.6. The mean was taken of the absolute value of all local curvatures. This was to prevent two or more curvatures from negating each other when the average was taken since only the magnitudes were sought and not the curve directions. This worked for 3D-data as well with minor adjustments to the code.

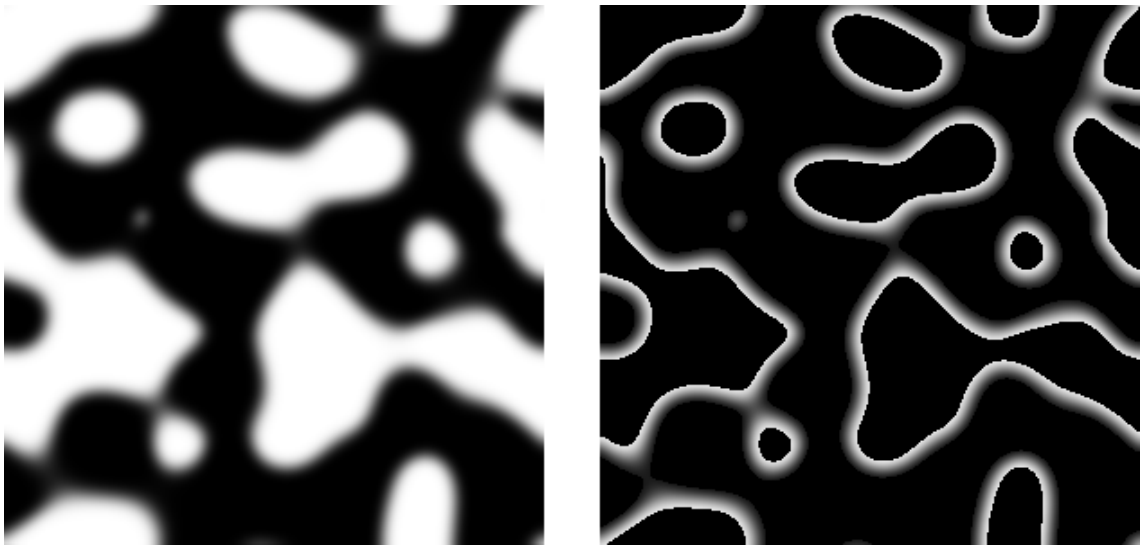


Figure 3.3: To the left is an example slice from a simulation using low viscosity with white and black representing the two phases. To the right is the same slice after points away from the interface have been removed.

3.2.4 Average concentration

The average concentration was studied per layer in the two bounded environments for three different contact angles. All concentration values in the bottom layer were summed together and then divided by the number of data points in that layer. The average concentration for every layer was found this way to see the fluctuations in the mixture. This was then repeated for vertical, standing layers to study the difference in average concentration from bottom to top and side to side.

4
Results

There is a desire to study how the biphasic structures evolve over time for different parameters. Periodically measuring the characteristic length and the average curvature are two meaningful ways to do this. This section presents the results as they have been achieved. There are three environments used. The first one is the box of size $300 \times 300 \times 300$ voxels with periodic boundaries in all directions. The second one is the domain which has size $300 \times 300 \times 100$, being shorter in height and having a glass surface at the bottom of the domain at z = 2.5 as well as being in contact with air at the top. The third environment is a variation of the second, it has a height of 50 instead of 100.

The simulations were run with both high and low viscosity with $\tau = 1$ and $\tau = 0.55$ respectively, as described in section 3.1.2. For the cube of side 300, the characteristic length with accompanying growth exponents and curvature is presented. These, along with the average concentration in two directions is presented for the two shorter environments. Both the 2D and 3D cases are considered for characteristic length and curvature. Both these cases use the same 3D simulation data but the 2D case measures the length scale and curvature in a 2D-slice, or several slices for the periodic length case as described in section 3.2.1.

4.1 Periodic Environment

Here the characteristic length scale, the growth exponents and the average curvature is presented for the periodic environment of size $300 \times 300 \times 300$.

4.1.1 Characteristic Length Scales

The characteristic length scales over time for the unbounded periodic environment is presented in figure 4.1. The length scale is not defined for the initial randomly generated landscape as it is just noise with no structures yet. The graph therefore starts at $t = 26.668$ s which is equal to 2000 iterations, this is the first output after the initial generation. For the 2D-case both the low viscosity and the high viscosity focused simulations start around 0.6 µm and then start to diverge. The characteristic length for the high viscosity case reaches about 1 µm after 30000 iterations which represents 400 s. For low viscosity, this length measurement reaches 1.66 µm by the end of the simulation. The five results for each viscosity agree well with each other.

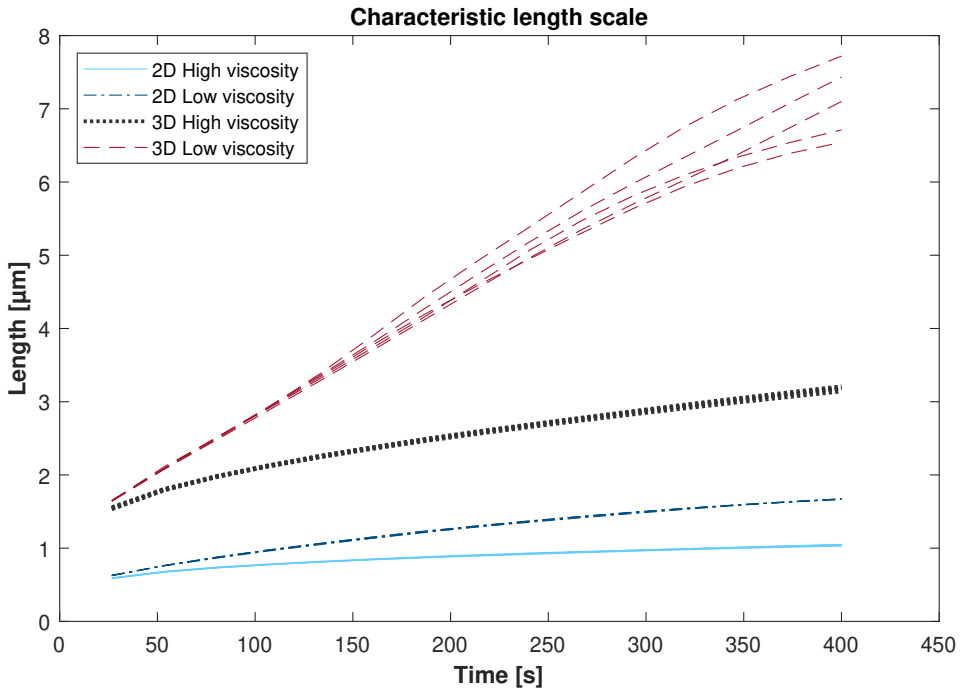


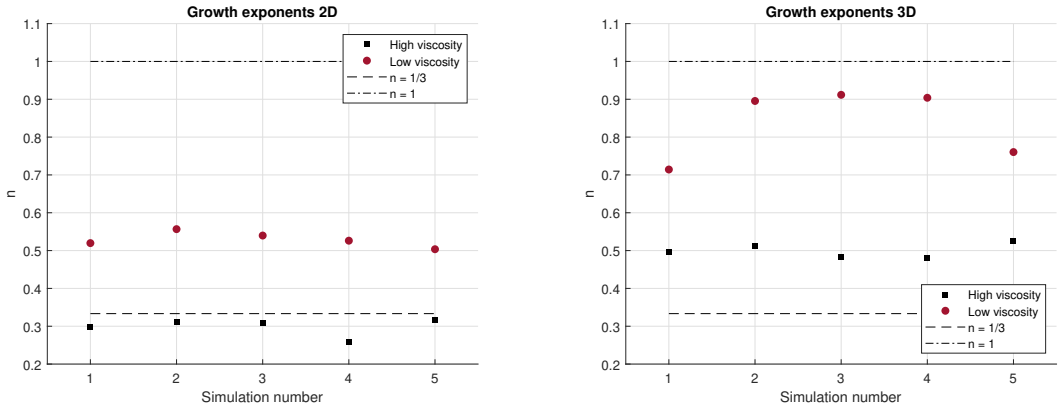
Figure 4.1: Length scale in the periodic environment for 2D and 3D-images for high and low viscosity.

When measuring 3D-structures, the length starts at around 1.6 µm for both high and low viscosity and for the high viscosity case continues to grow to about 3.2 µm whereas for the low viscosity case it grows strongly, almost linearly for almost 300 seconds for the growth to then start flattening out and end up around 7 µm. Worth mentioning is that the five simulations for the high viscosity case follow each other closely but the low viscosity simulations start separating and they end up further from each other.

4.1.2 Growth Exponents

The length scales were fitted to the function $L(t) = at^n + b$ where n is the growth exponent. This was done for all simulations in figure 4.1 and the results are presented in figures 4.2a and 4.2b. The figures show lines for $n = 1$ and $n = \frac{1}{3}$ which are the theoretical values for hydrodynamic growth and diffusive growth respectively.

The resulting growth exponents for the 2D-case agree quite well with each other for both high and low viscosity. In the 3D-case, the same low variation appears for high viscosity but the low viscosity simulations yielded results that vary more. The 3D measurements of the length scales yielded larger growth exponents for both viscosity cases.



(a) **(b)**

Figure 4.2: Growth exponents for 2D and 3D-length scales for high and low viscosity.

Tables 4.1 and 4.2 show the growth exponents and their accompanying R^2-values. The top two tables show the 2D-case with high viscosity to the left and low viscosity to the right. The lower two tables in turn show growth exponents for the 3D-case. All R^2-values from the simulations in the periodic environment are close to one indicating good curve fits.

Table 4.1: 2D high viscosity left, 2D low viscosity right

Simulation number	n	R^2
1	0.2990	0.9997
2	0.3121	0.9999
3	0.3085	0.9999
4	0.2595	0.9999
5	0.3167	0.9998

Simulation number	n	R^2
1	0.5197	0.9991
2	0.5567	0.9995
3	0.5397	0.9990
4	0.5262	0.9985
5	0.5036	0.9998

Table 4.2: 3D high viscosity left, 3D low viscosity right

Simulation number	n	R^2
1	0.4959	0.9999
2	0.5129	0.9998
3	0.4824	0.9998
4	0.4814	0.9997
5	0.5253	0.9999

Simulation number	n	R^2
1	0.7143	0.9969
2	0.8955	0.9996
3	0.9118	0.9976
4	0.9040	0.9992
5	0.7604	0.9949

4.1.3 Average Curvature

The average curvatures for the 2D and 3D structures are presented in figure 4.3. The graphed curves start at 26.668 seconds and end at 400 seconds which represents 2000 iterations and 30000 iterations respectively. This is the same situation as for the length scale in figure 4.1. The mean for the curvature being measured in the 2D

cross sections starts out at around $1.1\frac{1}{\mu m}$ for both high and low viscosity. However, these measurements quickly separate and the average curvature for the low viscosity case shrinks and drops faster than in the high viscosity case. At the end of the simulation these curvature values end up at around $0.9\frac{1}{\mu m}$ for the high viscosity case and around $0.5\frac{1}{\mu m}$ for the low viscosity case, albeit with more variance.

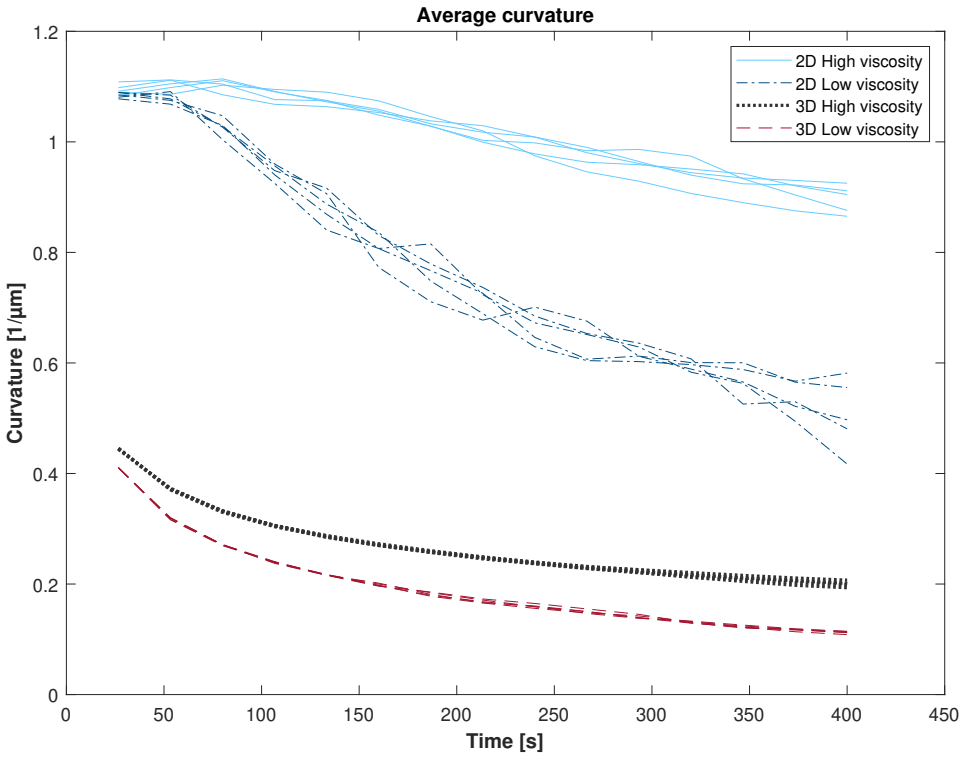


Figure 4.3: Average curvature in the periodic environment for 2D and 3D-images for high and low viscosity.

When measuring the curvature for the 3D-structures, the result starts out significantly lower than in the 2D-case; $0.44\frac{1}{\mu m}$ and $0.41\frac{1}{\mu m}$ for high and low viscosity respectively. The curvature for both cases then sharply decline in an exponential decay-like manner before finishing at simulation end at around $0.2\frac{1}{\mu m}$ and $0.11\frac{1}{\mu m}$ with the higher value corresponding to the high viscosity case and the lower value belonging to the low viscosity case. Note that the variation between the simulations is smaller for the 3D-calculations than for the 2D-calculations.

4.2 Bounded Environments

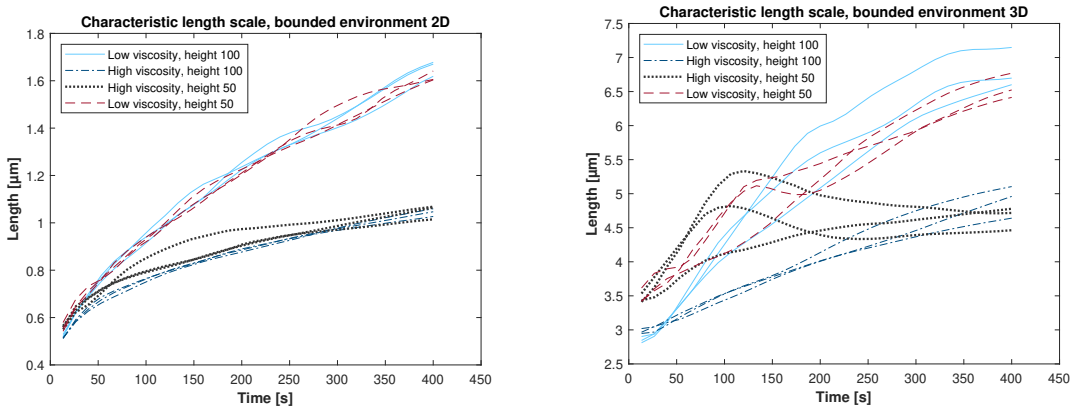
In this section, the characteristic length scales and their growth exponents, the average curvature and the average concentration is presented for the two bounded environments with the three different contact angles 50°, 70°, 90°. The two bounded environments have a height of 50 and 100 voxels. The results from the simulations

using 70° and 50° are shown separately from the results using 90° which is here considered the default and displayed first.

4.2.1 Characteristic Length Scales

Starting with the length scales in figure 4.4, the result from the 2D-analysis is presented to the left and the result from the 3D-analysis is presented to the right. Due to the smaller domain, these simulations output data every 1000 iterations while still running for 30000 iterations. This means that they start at $t = 13.334$ s while still ending at $t = 400$ s.

The 2D-lengths for all four cases start at about half a micrometer and then start growing with the low viscosity cases growing faster, reaching 1.6 µm by the end compared to the high viscosity cases reaching 1 µm by the end. For the most part, these simulations agree well with each other with one simulation differing more.



(a) **(b)**

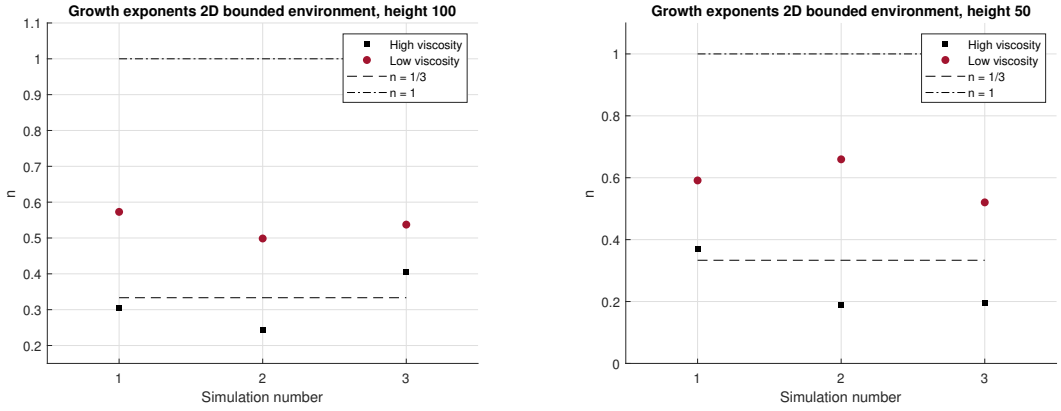
Figure 4.4: The characteristic length scales for the bounded environments using 2D and 3D analysis for high and low viscosity.

The 3D-lengths start at a higher value in comparison to the 2D-lengths and after a while they have a lot more variance. The environment with the shorter height of 50 have longer initial lengths than the taller environment, including the simulations using higher viscosity. The lengths of the low viscosity simulations of both heights then grow faster and quite soon the simulations having higher initial length but high viscosity gets passed. Note that three of the results have humps after which the length uncharacteristically shrinks.

4.2.2 Growth Exponents

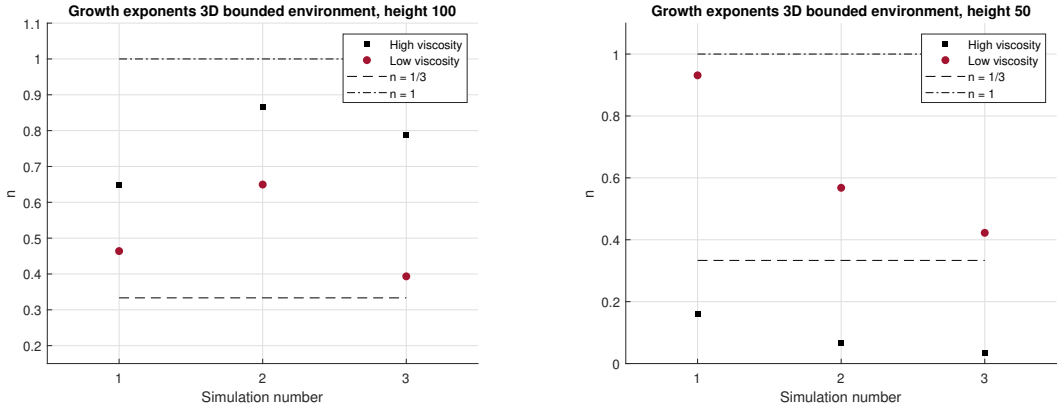
Figure 4.5 shows the growth exponents related to the 2D bounded environments while figure 4.6 shows growth exponents of the 3D bounded environments. The greater variance seen in the plots of figure 4.4b naturally also appear for their growth exponents. In figure 4.6a the high viscosity growth exponents are larger and closer to 1 than their low viscosity counterparts as their corresponding lines are straighter

and more linear. In figure 4.6b there are two small growth exponents, these belong to two simulations with humps and their R^2-values are small, indicating a poor fit to the function.



(a) **(b)**

Figure 4.5: Growth exponents for 2D bounded environment for different heights and viscosities compared.



(a) **(b)**

Figure 4.6: Growth exponents for 3D bounded environment for different heights and viscosities compared.

The following tables list the R^2-values of the shown growth exponents. Most of the values are close to 1, with two values specifically being much lower implying bad fits.

Table 4.3: 2D high viscosity with height 100 left, 2D low viscosity with height 100 right.

Simulation number	n	R^2
1	0.3053	0.9997
2	0.2428	0.9990
3	0.4058	0.9993

Simulation number	n	R^2
1	0.5729	0.9976
2	0.4986	0.9982
3	0.5374	0.9966

Table 4.4: 2D high viscosity with height 50 left, 2D low viscosity with height 50 right.

Simulation number	n	R^2
1	0.3681	0.9977
2	0.1903	0.9745
3	0.1938	0.9968

Simulation number	n	R^2
1	0.5913	0.9980
2	0.6594	0.9939
3	0.5203	0.9984

Table 4.5: 3D high viscosity with height 100 left, 3D low viscosity with height 100 right.

Simulation number	n	R^2
1	0.6485	0.9939
2	0.8672	0.9988
3	0.7877	0.9958

Simulation number	n	R^2
1	0.4639	0.9794
2	0.6496	0.9961
3	0.3933	0.9904

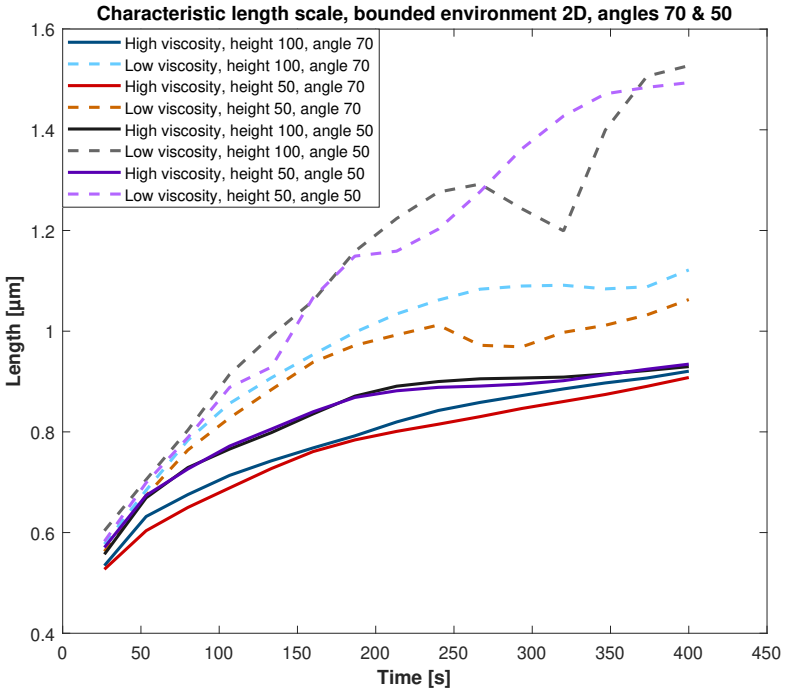
Table 4.6: 3D high viscosity with height 50 left, 3D low viscosity with height 50 right.

Simulation number	n	R^2
1	0.1618	0.9892
2	0.0659	0.3502
3	0.0331	0.2148

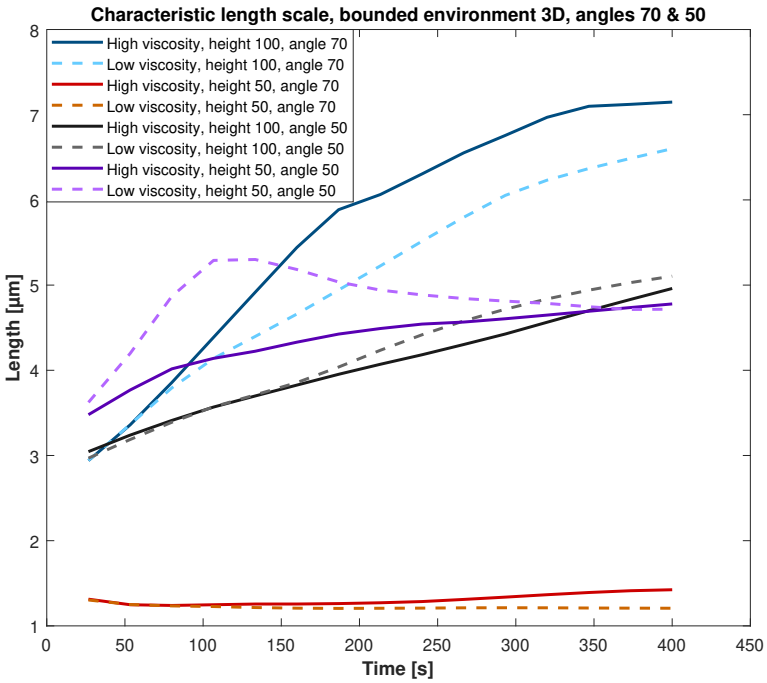
Simulation number	n	R^2
1	0.9312	0.9893
2	0.5680	0.9650
3	0.4224	0.9807

4.2.3 Characteristic Length Scales for Contact Angles 70 & 50

To see how different contact angles affect the length scales, the results from simulations using 70° and 50° contact angles with the boundary floor are presented in figure 4.7. The behaviour is mostly unchanged in the 2D analysis with only the length scales of the simulations using a contact angle of 70° and low viscosity growing slower. The 3D-behaviour is also mostly the same and even includes one hump. However, the two simulations using a height of 50 and a contact angle of 70° stand out. They initialise with shorter lengths and experience little to no growth throughout the entire simulation.



(a)



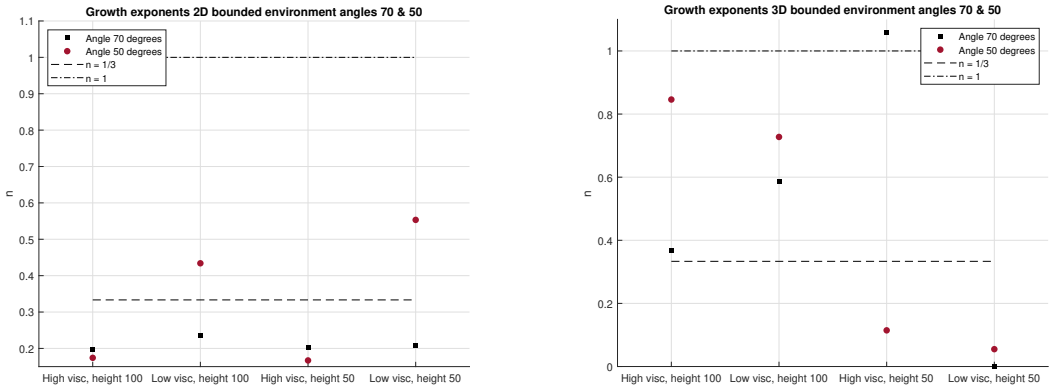
(b)

Figure 4.7: Length scale in the bounded environment for 2D and 3D-images with contact angles of 70 and 50° and a variety of parameters.

4.2.4 Growth Exponents for Contact Angles 70 & 50

The growth exponents for the varying heights, angles and viscosities are presented graphically in figure 4.8 below and then listed more detailed in tables 4.7 and 4.8 below the graphs. For 2D and 70°, varying the height of the bounded environment does not affect the growth exponent much in this case as the 2D-layer is taken from down low as described in section 3.2.1. Varying the viscosity more surprisingly also does not change the exponents much but the accompanying R^2-values are slightly lower. However, when the contact angle is 50° the growth exponent increases for low viscosity.

Three of the eight fitted growth exponents for the 3D-structures have lower R^2-values but it appears that for a contact angle of 50°, lowering the height of the domain also lowers the growth exponent.



(a) **(b)**

Figure 4.8: Growth exponents for the bounded environment for alternate angles.

Tables 4.7 and 4.8 list the growth exponents and their R^2-values for varying parameters. Most of them are close to 1 but three in particular are further away.

Table 4.7: 2D growth exponents and R^2-values for angles 70° and 50°.

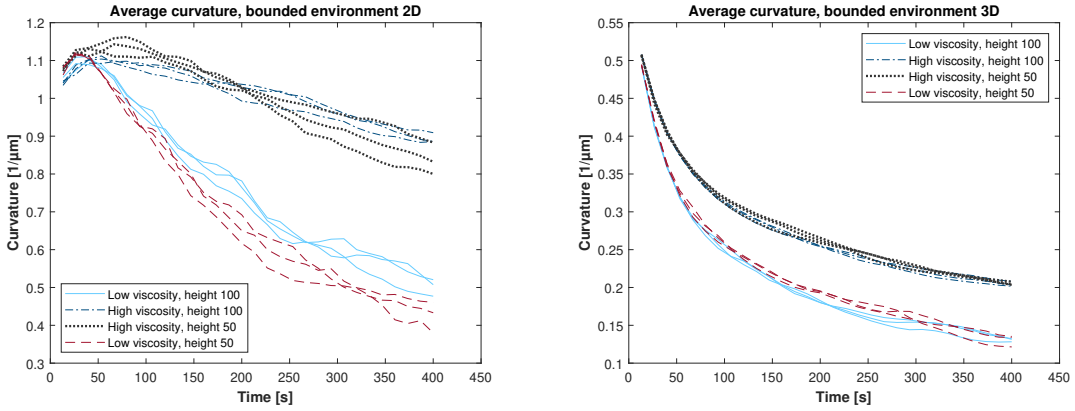
Angle	Height	Viscosity	n	R^2
70	100	High	0.1963	0.9982
70	100	Low	0.2359	0.9667
70	50	High	0.2014	0.9984
70	50	Low	0.2078	0.9347
50	100	High	0.1742	0.9592
50	100	Low	0.4340	0.9563
50	50	High	0.1670	0.9680
50	50	Low	0.5533	0.9922

Table 4.8: 3D growth exponents and R^2-values for angles 70° and 50°

Angle	Height	Viscosity	n	R^2
70	100	High	0.3690	0.9842
70	100	Low	0.5863	0.9967
70	50	High	1.0588	0.6958
70	50	Low	0.0000	0.0000
50	100	High	0.8461	0.9988
50	100	Low	0.7275	0.9963
50	50	High	0.1146	0.9956
50	50	Low	0.0550	0.2465

4.2.5 Average Curvature

To see how the curvature of the structures evolves over time in the bounded environments, two plots were produced and are shown in figure 4.9. The average curvature of the 2D-images in the bounded environment behaves mostly similar to how it does in the free, unbounded environment. There are two main exceptions. The first one is that in the beginning of the simulation, during the first 50-100 seconds, the average curvature increases before it starts to decrease. The second one is that when the parameters are set to a height of 50 and a low viscosity, the average curvature decreases faster and reaches a lower result by the end of the simulation compared to what happens in the periodic environment.



(a) **(b)**

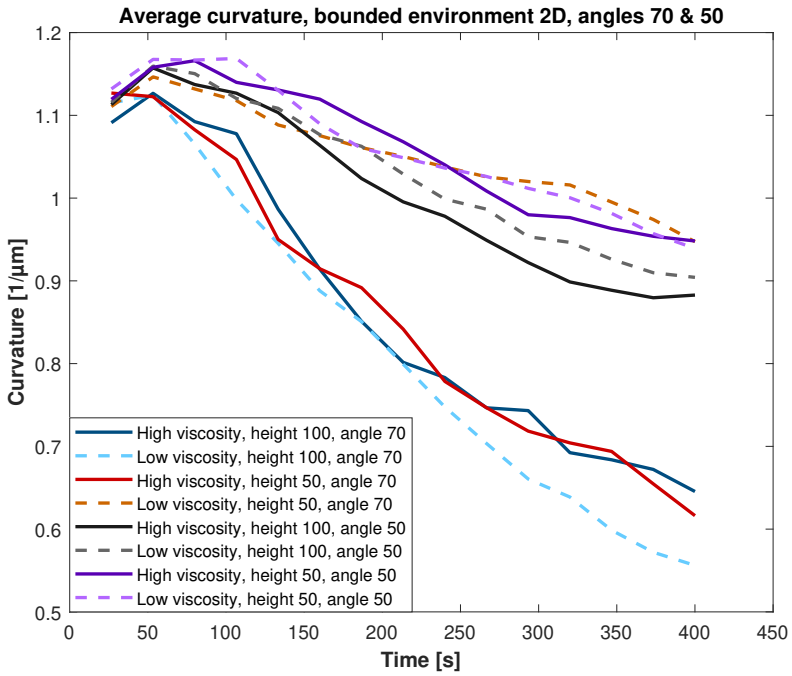
Figure 4.9: Average curvature in the bounded environments for 2D and 3D-images for high and low viscosity.

For the 3D-structures, the average curvature starts higher at around $0.5\frac{1}{\mu m}$ rather than $0.44\frac{1}{\mu m}$ and $0.41\frac{1}{\mu m}$ in the free environment. The behaviour of the curvature is then very similar to the behaviour in the unbounded case with the exception that for low viscosity and both heights, the average curvature finishes slightly higher than in the periodic environment. Note that for the simulations using the same parameters, the variance is much lower than in the 2D-case.

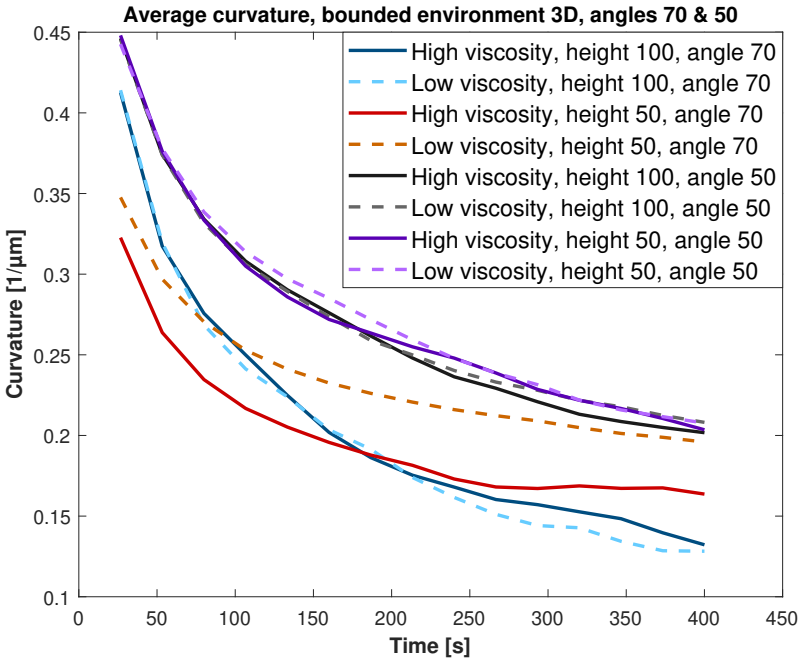
4.2.6 Average Curvature for Contact Angles 70 & 50

The average curvature was again calculated but with the contact angle instead being 70° and 50° in the interest of seeing how this affects the behaviour. Interestingly, the viscosity of the liquid now seems to matter less and instead it is the angle that primarily determines the rate of curvature shrinkage. For an angle of 70° the shrinkage is faster and more significant than for an angle of 50° and this pattern appears for both 2D and 3D-analysis. The one exception to this rule in both cases is that the simulation using low viscosity, a height of 50 and and angle of 70 has a slower shrinkage.

There are two other main differences to the behaviour. These include that when the contact angle is 90° the low viscosity simulations reach a lower value for the average curvature than for the alternate angles. The second other main difference is that the curvature in figure 4.10b starts lower and more spread out compared to figure 4.9b where the curvature for the different simulations all start at about $0.5\frac{1}{\mu m}$.



(a)



(b)

Figure 4.10: Average curvature in the bounded environments for 2D and 3D-images with contact angles of 70 and 50° and a variety of parameters.

4.2.7 Average Concentration

In order to explore the phase buildup at the bottom of the environment volumes caused by different contact angles, a collection of visualisations of the phenomena are presented followed by graphs displaying the average concentration of every layer both along the x and z-axis.

Figures 4.11-4.14 showcase ParaView-generated graphics from the side based on the final iteration of different simulations. It is possible to see that the red phase content is higher at the bottom for the lower angles and highest for the angle of 50°. For 90° there is no effect and the bottom has a similar pattern to the rest of the box. For low viscosity the structures are predictably larger and the bottom layer is thicker. The height of the environment does not seem to have an impact on the aggregated bottom phase.

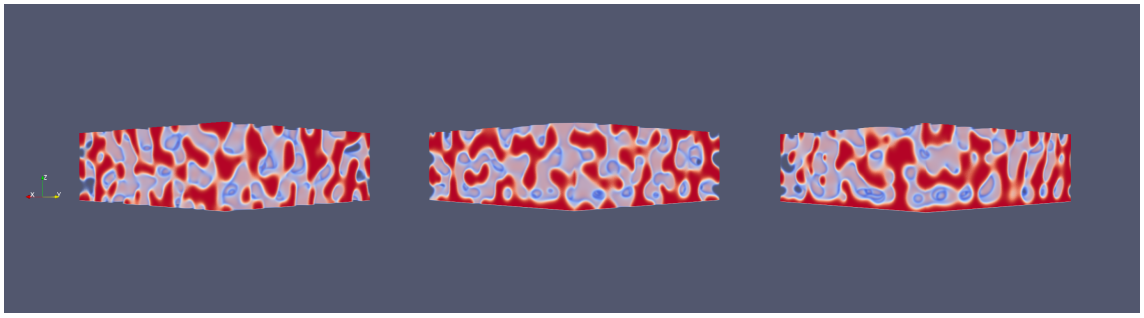


Figure 4.11: Height 100, high viscosity, angles: 90° - 70° - 50°

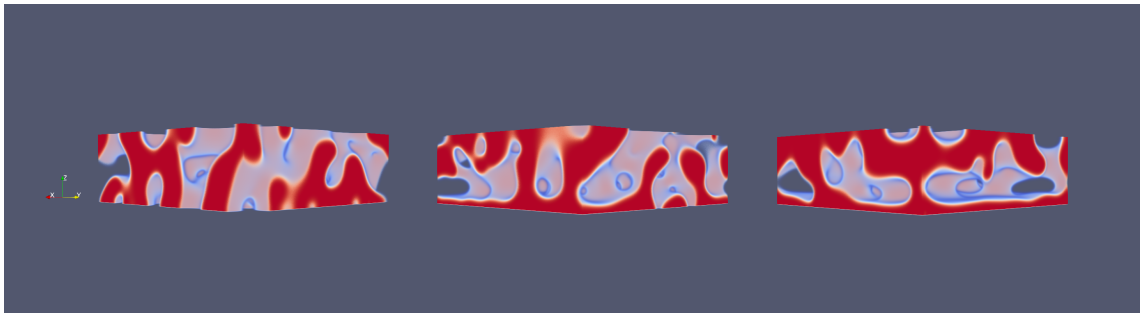


Figure 4.12: Height 100, low viscosity, angles: 90° - 70° - 50°

Figure 4.13: Height 50, high viscosity, angles: 90° - 70° - 50°

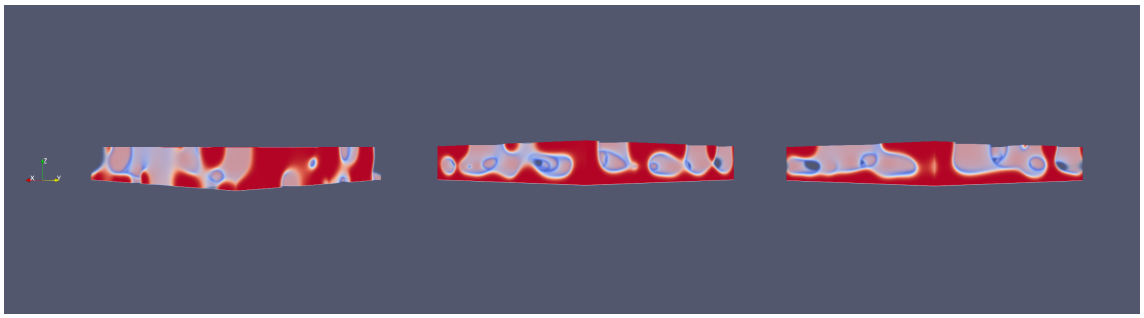
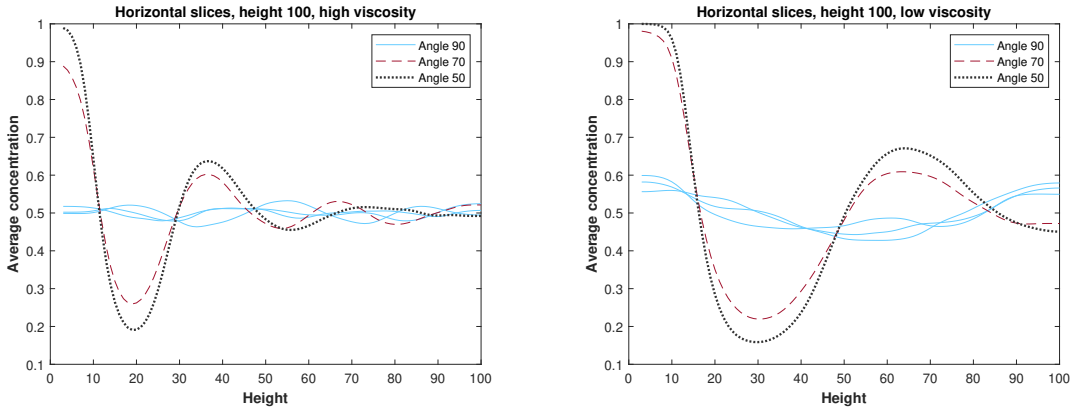


Figure 4.14: Height 50, low viscosity, angles: 90° - 70° - 50°

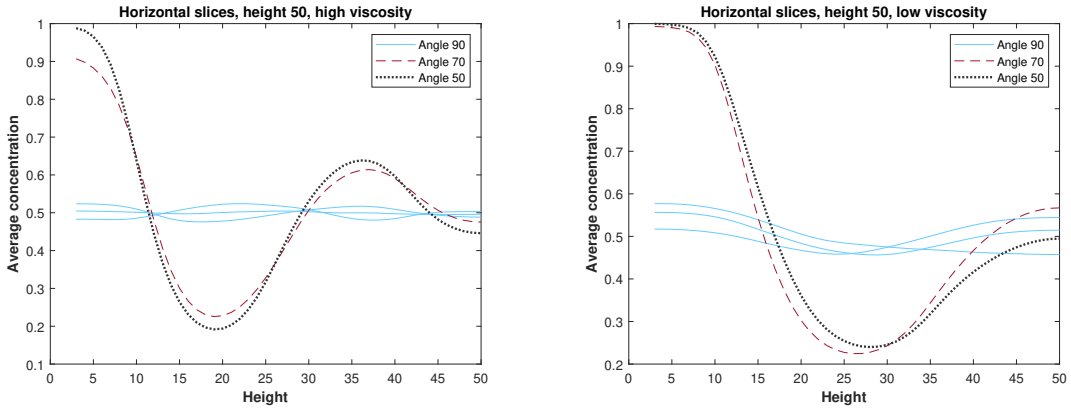
To explore the average concentration and the shape of the bottom formations deeper, the mean concentration of every layer was calculated and these results are presented in figures 4.15a-4.18. For the first two sets of graphs, the layers are stacked horizontally and thus the mean concentration along the z-axis is shown. This is done for two heights, high and low viscosity, and for three angles of contact; 90°, 70° and 50°.

For high viscosity, a contact angle of 50° gives a higher concentration at the bottom while for low viscosity, this effect is less pronounced or missing. The phase layers are also thicker for low viscosity. A neutral angle of 90° gives no pattern and makes the average concentration seemingly random with little variance.



(a) **(b)**

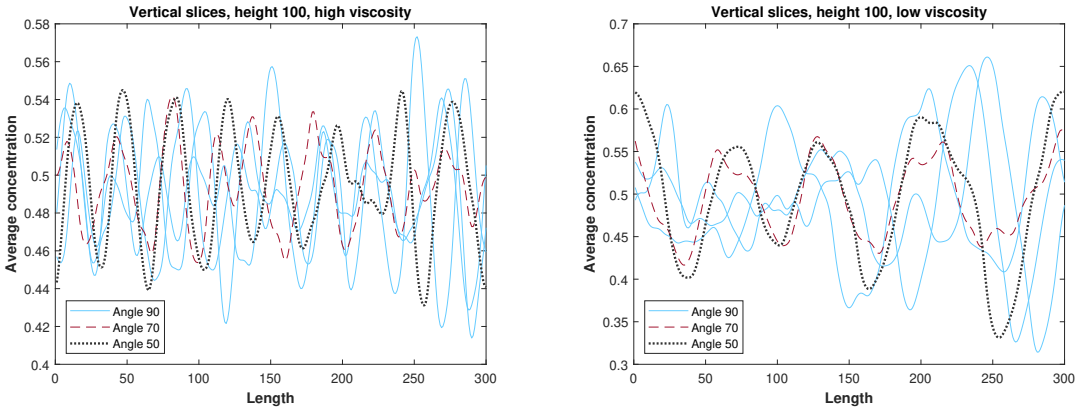
Figure 4.15: The average concentration in horizontal slices for a height of 100 voxels, high and low viscosity and three different angles.



(a) **(b)**

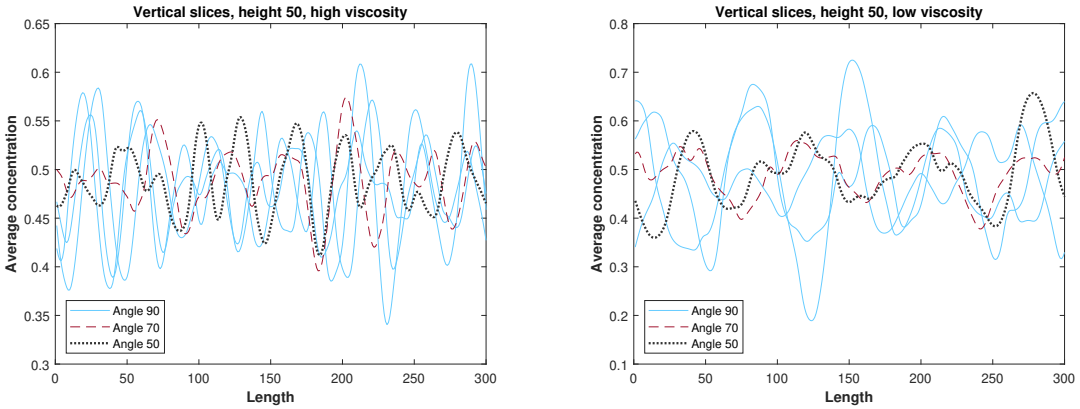
Figure 4.16: The average concentration in horizontal slices for a height of 50 voxels, high and low viscosity and three different angles.

The second two sets of graphs, figures 4.17-4.18, instead have vertical layers stacked along the x-axis. These graphs stand in stark comparison to the ones produced by the horizontal layers, the data is very noisy and there appears to be no pattern for any combination of height, viscosity and angle.



(a) **(b)**

Figure 4.17: The average concentration in vertical slices for a height of 100 voxels, high and low viscosity and three different angles.



(a) **(b)**

Figure 4.18: The average concentration in vertical slices for a height of 50 voxels, high and low viscosity and three different angles.

4.3 Comparison with experiment data

The experimental data from Carmona's licentiate thesis can be compared to the measurements of the data from the simulations to check the accuracy of the simulations. Since the experimental data is gathered from 2D-images, it is compared to the 2D-measurements from the simulations.

The experimental data for the evolution of the characteristic length scale for the bicontinuous phase structure is based on a mixture of EC/HPC with 37 wt% HPC where the same parameter is 50 wt% for the simulations [1]. A comparison of the results still shows somewhat similar behaviour, but the characteristic lengths of the physical experiments are shown to be significantly longer; 2.0 µm after 256s, increasing to 4.9 µm by 328s and plateauing at 5.0 µm after 495s. For the low viscosity simulation in the free environment, the same first two moments have lengths of about 1.4 µm and 1.57 µm [1], respectively. For the bounded environment, the same lengths at these two moments are about 1.35 µm and 1.5 µm. It is worth mentioning that the length for the 3D-low viscosity case grows faster and is more similar to the experiment length.

The experimental solution with 50 wt% plateaus at about 1.3 µm which is closer to the results from the simulations [1]. However, experimentally this solution does not produce a bicontinuous phase but rather a discontinuous one [1]. This makes for a bad comparison and it is why the 50 wt% solution is compared to instead.

The accompanying experimental growth exponents for the bicontinuous cases (25 wt%, 30 wt%, 35 wt%, 37 wt%) are all close to one compared to the 2D-low viscosity exponents that are all a little over 0.5 [1]. For 40, 45 and 50 wt%, the exponents are about 2/3 but again, these measure discontinuous structures and are thus less comparable [1].

The average curvatures, on the other hand, are more comparable. The 2D-low viscosity simulations in the unbounded environment all start slightly under $1.1\frac{1}{\mu m}$, then decline and reach $0.6\frac{1}{\mu m}$ by 300 seconds. The 37 wt% solution from the experiments started at $0.9\frac{1}{\mu m}$ and finishes over $0.55\frac{1}{\mu m}$, also after 300s [1]. The 2D-low viscosity simulations in the bounded environment is also quite comparable, starting at $1.05\frac{1}{\mu m}$ and reaching $0.6\frac{1}{\mu m}$ by 300s.

5
Discussion

The aims are here brought up again. The first one was to develop methods to measure the patterns over time. Previous code was used to find the characteristic length from 2D-data and this code was altered to also work for 3D-data. Two methods were developed to find the average curvature, both from 2D and 3D-data. The growth exponents were found using curve fitting and the average concentration through the domain was recorded.

The second aim was to compare the simulation results to the results from previous experiments. The measurements from the simulations were converted from general units to µm by comparing the characteristic lengths when visible structures start to appear. In these comparisons it was seen that the 2D-lengths were significantly shorter than in experiments for both high and low viscosity and in the bounded and unbounded environment. These characteristic lengths are larger for 3D and more in line with the results from the experiments. The reason for why the length scales are larger in 3D compared to 2D is not entirely clear. It is seen in figure 3.2 that the radial distribution peaks for a smaller value in the Fourier space and thus has a larger length scale in real space. It is possible this happens because the additional dimension of depth allows for more angles and paths to be searched along and thus a longer distance will often be found.

The growth exponents for the unbounded case are more stable and well agreeing with each other. In general, the results from the bounded environments seem to be less predictable. This is possibly the lower heights in these environments causing unexpected behaviour and making anomalies more likely. Most of the time, the lower viscosity liquid yielded higher growth exponents, likely because the length keeps growing for longer before stagnation occurs. This can be seen in the function that the data was fitted against; $L(t) = at^n + b$. The function is linear for $n = 1$ and so straighter lines will have larger exponents. Figure 4.6a is an exception to this. Here the high viscosity simulations had larger growth exponents than the low viscosity ones. This can be partially explained by studying the relevant lines in the accompanying graph in figure 4.4b. The growth stagnation is much less pronounced for high viscosity, resulting in a straighter line which yields a larger growth exponent. More simulations are needed to find why this happens for specifically these parameters.

The 2D-average curvatures, specifically for low viscosity, were quite comparable to the physical experiments. They generally behave the same, for both bounded and unbounded environments. The mean curvatures measured in 3D are all much smaller

and have less variance between the different simulations. One possible reason for why this happens comes from the difference in how the curvature is calculated in 2D and 3D. In 2D, the mean curvature is calculated from the curvature at every point on the interface. This means that areas with larger curvature have more points associated with them and thus contribute more heavily to the overall average. In 3D, it is instead the principal curvatures that are found and averaged. These are the minimum and maximum curvatures which avoids the problem with the weighted average.

The third aim was to study the effect of the glass boundary. It seemed to create more chaotic results for the characteristic length scale and average curvature but the average concentration followed expected behaviour; damped oscillations along the z-axis and random noise along the x-axis. The humps appearing in figures 4.4b and 4.7b are possibly appearing because of the gathering of phase at the bottom of the bounded environments making the measured structures grow differently in the z-direction compared to the x and y-directions. It is also worth considering the short height; 50 voxels is 4µm which is comparable to the characteristic length scale of the humps in the graphs. It could be that the length scale is being limited by the height.

Concerning using many 2D-slices to further average the result, it appears to be better as it produces less variation and likely gives a more accurate result since the characteristic length from one slice could be an outlier.

5.1 Future work

Some length results have uncharacteristic humps or are completely flat with no growth at all, in particular figures 4.4b and 4.7b. To find out why this is happening, more work is needed. It is possibly a quirk stemming from the shorter bounded environment combined with the method of calculation. Perhaps the result varies greatly from simulation to simulation; this could be tested by running more of the bounded simulations using the same parameters. It would also be interesting to try out a taller bounded environment.

A big limitation of the simulation is that it does not model the evaporation of ethanol that happens in the physical experiments. Normally, this would increase the viscosity of the remaining liquid over time and eventually lead to growth stagnation by kinetic trapping [1]. This evaporation and reduction in boundary height could be modelled with a moving boundary and an increasing space between the voxels to treat the moving boundary as stationary [3]. Doing so would avoid recalculating the LBM-domain in every time step, something that would take too much time to be realistically feasible. A possible transformation to make this possible was presented but not implemented in [3].

Bibliography

[1] P. Carmona, *Structure evolution of spin-coated phase-separated EC/HPC films*. Licentiate thesis, Chalmers University of Technology, Gothenburg, Sweden, May 2021.

[2] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen, "The lattice boltzmann method," *Springer International Publishing*, vol. 10, no. 978-3, pp. 4–15, 2017.

[3] A. Rodin, "Computational study of spinodal decomposition in thin layers," Master's thesis, Chalmers University of Technology, Gothenburg, Sweden, June 2022.

[4] C. T. Huynh and D.-S. Lee, "Controlled release," *Encyclopedia of Polymeric Nanomaterials, 2014*, pp. 1–12, 01 2014.

[5] G. Tiwari, R. Tiwari, B. Sriwastawa, L. Bhati, S. Pandey, P. Pandey, and S. K. Bannerjee, "Drug delivery systems: An updated review," *International journal of pharmaceutical investigation*, vol. 2, no. 1, p. 2, 2012.

[6] N. Das, "Phase behaviour and separation kinetics of polymer blends," *Journal of Microscopy*, vol. 253, no. 3, pp. 198–203, 2014.

[7] R. A. Jones and R. W. Richards, *Polymers at surfaces and interfaces*. 1999.

[8] A. Fakhari, T. Mitchell, C. Leonardi, and D. Bolster, "Improved locality of the phase-field lattice-boltzmann model for immiscible fluids at high density ratios," *Physical Review E*, vol. 96, no. 5, p. 053301, 2017.

[9] Y. Sun and C. Beckermann, "Sharp interface tracking using the phase-field equation," *Journal of Computational Physics*, vol. 220, no. 2, pp. 626–653, 2007.

[10] M. Geier, A. Fakhari, and T. Lee, "Conservative phase-field lattice boltzmann model for interface tracking equation," *Physical Review E*, vol. 91, no. 6, p. 063309, 2015.

[11] P.-H. Chiu and Y.-T. Lin, "A conservative phase field method for solving incompressible two-phase flows," *Journal of Computational Physics*, vol. 230, no. 1, pp. 185–204, 2011.

[12] A. Fakhari, M. Geier, and T. Lee, "A mass-conserving lattice boltzmann

method with dynamic grid refinement for immiscible two-phase flows," *Journal of Computational Physics*, vol. 315, pp. 434–457, 2016.

[13] T. Hashimoto, "Dynamics in spinodal decomposition of polymer mixtures," *Phase Transitions: A Multinational Journal*, vol. 12, no. 1, pp. 47–119, 1988.

[14] A. Bray, "Coarsening dynamics of phase-separating systems," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1805, pp. 781–792, 2003.

[15] A. P. Solon, J. Stenhammar, M. E. Cates, Y. Kafri, and J. Tailleur, "Generalized thermodynamics of motility-induced phase separation: phase equilibria, laplace pressure, and change of ensembles," *New Journal of Physics*, vol. 20, no. 7, p. 075001, 2018.

[16] Y. A. Cengel, A. J. Ghajar, and M. Kanoglu, *Heat and mass transfer: fundamentals and applications*. 2011.

[17] T. Fujita and M. W. Chen, "Characteristic length scale of bicontinuous nanoporous structure by fast fourier transform," *Japanese journal of applied physics*, vol. 47, no. 2R, p. 1161, 2008.

[18] J. Zhu, R. Balieu, X. Lu, and N. Kringos, "Microstructure evaluation of polymer-modified bitumen by image analysis using two-dimensional fast fourier transform," *Materials & design*, vol. 137, pp. 164–175, 2018.

[19] J. E. Taylor, "Ii—mean curvature and weighted mean curvature," *Acta metallurgica et materialia*, vol. 40, no. 7, pp. 1475–1485, 1992.

A

Appendix

Here, some of the used MATLAB code is presented. This includes the functions that calculate and measures the length scales and curvatures in 2D and 3D, the code that calculates the growth exponents, and the code that calculates the average concentration. The code that calls the presented functions and saves the results has been left out along with the code for plotting figures.

Length Scale 2D

```
function [L,avpixsmooth] = Lengthscale(I,FOV)
I=double(I);
I=I/255; %normalize the pixels

% for 2d characteristic length

%remove the noise background
G=imgaussfilt(I,100);
I=I-G;

% smooth the image
I=imgaussfilt(I,2);

% imshow(I)

%Fast Fourier transform
I=I-mean(I(:));
F=abs(fftshift(fft2(I))).^2;
%figure
%imagesc(F,[0,1E6])
imshow(F)

%Create coordinate with matrix x for x direction and y for y direction
[x,y]=ndgrid(1:1024,1:1024); %the image is 1024x1024 pixel
x=x-513;
y=y-513; %put the center in the center of the FFT
```

```
q=sqrt(x.^2+y.^2); %creation of matrix with radius (distance)

D=0:99;
avpix=zeros(size(D));

for i=1:numel(D) %i is the increment of the radius range
                 %where to take the average of pixels
    dmin=D(i)-0.5; %center the donut on increment i
    dmax=D(i)+0.5; %the thickness of one donut is 1
    indic=q>=dmin & q<dmax ;
    avpix(i)=mean(F(indic)); %take the average of the pixel value
                             %within the donut

end

%smoothing the avpix curve with a gaussian filter
avpixsmooth=smoothdata(avpix,'gaussian',10);
avpixsmooth2=avpixsmooth/trapz(D,avpixsmooth);
%Caluculate the integral by the area under the curve (trapeze approach)
av=trapz(D,D.*avpixsmooth2);
%Transfrom the fonction in L
L=FOV./av ;
end
```

Length Scale 3D

```
function L = Lengthscale3d(I,N,length,width,height)

    I = double(I);

    %remove the noise background
    G=imgaussfilt3(I,100);
    I=I-G;

    % smooth the image
    I=imgaussfilt(I,2);

    %Fast Fourier transform
    I=I-mean(I(:));
    F=abs(fftshift(fftn(I))).^2;

    %Create coordinate-matrix
    [x,y,z]=ndgrid(1:length,1:width,1:height); %the image is
                                               %300x300x300 pixel
    x=x-length/2+1; %151;
```

```
    y=y-width/2+1; %151; %put the center in the center of the FFT
    z=z-height/2+1; %151;

    q=sqrt(x.^2 + y.^2 + z.^2); %creation of matrix with
                                %radius (distance)

    D=0:99;
    avpix=zeros(size(D));

    for i=1:numel(D) %i is the increment of the radius range
                     %where to take the average of pixels
    dmin=D(i)-0.5; %center te donut on increment i
    dmax=D(i)+0.5; %the thickness of one donut is 1
    indic= q >= dmin & q < dmax;
    avpix(i)=mean(F(indic)); %take the average of the pixel
                             %value within the donut
    end

    %smoothing the avpix curve with a gaussian filter
    avpixsmooth=smoothdata(avpix,'gaussian',10);
    %Normalize
    avpixsmooth2=avpixsmooth/trapz(D,avpixsmooth);
    %Caluculate the integral by the area under the curve,
    %(trapeze approach)
    %average distance just equal to the integral now because
    %we already normalized
    averageDistance=trapz(D,D.*avpixsmooth2);
    %Transfrom the function in L
    L = N/(averageDistance);
end
```

Growth Exponents

The 2D case for high viscosity is presented here without the other cases as they are the same.

```
clear
clc

load('results2dLength.mat')

growth_exponent_2d_high_visc_lsq = zeros(1, nSimulations);
rsquare_2d_high_visc_lsq = zeros(1, nSimulations);

x = (1:15)'.*6.667*4;
f = @(c,x) c(1)*x.^c(2)+c(3);
```

```
x0 = [1 1 1];
lb = [0 0 0];
ub = [inf inf inf];

% 2d high viscosity
for iSimulations = 1:5
    y = results_length_2d_high_viscosity(iSimulations,:)';
    c = lsqcurvefit(f,x0,x,y,lb,ub);
    fittedy = c(1)*x.^c(2)+c(3);
    growth_exponent_2d_high_visc_lsq(iSimulations) = c(2);
    RSS = sum((y-fittedy).^2);
    TSS = sum((y-mean(y)).^2);
    rsquare_2d_high_visc_lsq(iSimulations) = 1-RSS/TSS;
end
```

Curvature 2D

```
function averageCurvature = calculateCurvature2dGradDiv(data_vtk, N)

% find the normal at all points by taking the gradient
and then normalize
data_vtk(data_vtk > 0.9) = NaN;
data_vtk(data_vtk < 0.1) = NaN;
[fx, fy] = gradient(data_vtk);
for i = 1:N
    for j = 1:N
        normFactor = sqrt( fx(i,j)^2 + fy(i,j)^2 );
        fx(i,j) = fx(i,j)/normFactor;
        fy(i,j) = fy(i,j)/normFactor;
    end
end

% curvature = divergence of the normal
curvature = divergence(fx, fy);

curvature(abs(curvature) > 0.3) = 0;

averageCurvature = mean(abs(curvature),'all','omitnan');

end
```

Curvature 3D

```
function averageCurvature = calculateCurvature3d(data_vtk,
length, width, height)

% find the normal at all points by taking the gradient
and then normalize
data_vtk(data_vtk > 0.9) = NaN;
data_vtk(data_vtk < 0.1) = NaN;
[fx, fy, fz] = gradient(data_vtk);
for i = 1:length
    for j = 1:width
        for k = 1:height
            normFactor = sqrt( fx(i,j,k)^2 + fy(i,j,k)^2 +
            fz(i,j,k)^2 );
            fx(i,j,k) = fx(i,j,k)/normFactor;
            fy(i,j,k) = fy(i,j,k)/normFactor;
            fz(i,j,k) = fz(i,j,k)/normFactor;
        end
    end
end

% curvature = divergence of the normal
curvature = divergence(fx, fy, fz);

curvature(abs(curvature) > 0.3) = 0;

averageCurvature = (1/2) * mean(abs(curvature),'all','omitnan');

end
```

Average Concentration

Again, only one case is presented as the other cases uses the same code with other data.

```
clear
clc

textfile = fopen('filepaths_boundary_1-6.txt');
textcell = textscan(textfile,'%s','delimiter','\n','TextType','string');
filepathsStrings = textcell{1};

boxlength = 300;
width = 300;
bigHeight = 100;
```

```
% 90 degree angle
averageConcentrationHorizontal100 = zeros(6, bigHeight);
averageConcentrationVertical100 = zeros(6, width);

% 90 degrees, height 100
for iSimulations = 1:6

        [data_vtk_cell, info] = vtk_read_volume(strcat(
'D:\Skola\Mastersarbete\Resultatdata\boundary',num2str(iSimulations),
'\',filepathsStrings(end)));
        data_vtk = cell2mat(data_vtk_cell(1, 1));

        for i = 1:bigHeight

            averageConcentrationHorizontal100(iSimulations, i) =
            mean(data_vtk(:,:,i), 'all');

        end

        for j = 1:300

            averageConcentrationVertical100(iSimulations, j) =
            mean(data_vtk(j,:,:), 'all');

        end

        disp(strcat("Simulation: ",num2str(iSimulations)))

end
```

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY