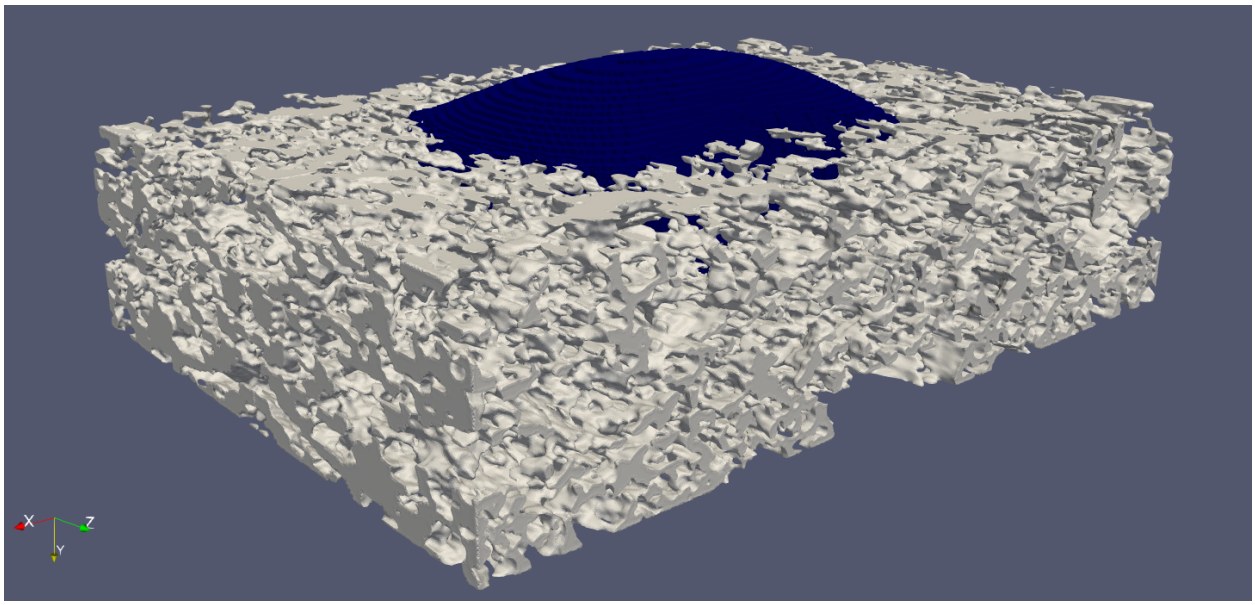
Simulation of the Spreading and Penetration of Droplets into a Paper Coating Layer using the Volume of Fluid Method with OpenFOAM

B. V. van der Hoek

May 6, 2019



First supervisor: A. Raoof, Second supervisor: S.M. Hassanizadeh

Contents

Acknowledgements

Firstly, I would like to thank Hamed Aslannejad for the many useful discussions we had and for reviewing the first version of this thesis. Furthermore, I would like to thank Eva Schoonderwoerd for the fruitful collaboration running the OpenFOAM simulations. I would also like to thank Guang Yang from the University of Stuttgart for his help with the OpenFOAM software during my stay there. Big thanks to Enno de Vries for his help with technical matters such as the workstation and the hydrology cluster. I'd also like to thank Annuska Exel for her help with many practical matters. Fianlly, I'd like to thank Vincent Hoogenboom for the many discussions we had and the creative ideas he has shared.

1 Introduction

Inkjet printing is a way of printing liquids onto porous surfaces that has a wide range of applications. It can be applied to printing of ink on paper, but it also has many industrial applications, including the printing of solder in micro-electronics soldering (De Gans et al., 2004), printing of polymers for organic light-emitting diodes (Calvert, 2001) and transistor circuits (Sirringhaus et al., 2000), the microseeding of living cells (Nakamura et al., 2005), 3-D printing (Le, 1998) and printing Active Pharmaceutical Ingredients (APIs) on substrates (Scoutaris et al., 2016). In the case of inkjet printing on paper, a distinction can be made between printing on uncoated paper and printing on coated paper. Uncoated paper can be characterised as a fibrous layer, consisting of cellulose fibers (Aslannejad and Hassanizadeh, 2017). In the case of coated paper, this fibrous layer is coated with a thin layer of very fine mineral particles, commonly calcium carbonate or kaolin. A binder is used to make these mineral particles stick together (Aslannejad et al., 2017). This is done to confine the ink penetration during the printing process. One of the goals in inkjet printing is to prevent the coalescence of adjacent droplets, as this leads to unexpected and detrimental results (Daniel and Berg, 2006), thus the spreading of the ink droplet on top of the surface that is being printed on should be minimised. During the settling of a droplet on top of a porous substrate, a variety of processes take place that influence the spreading of the droplet. Firstly, the inertia of the droplet influences how the droplet interacts with the surface. Furthermore there are gravitational and capillary effects due to the porous structure of the underlying substrate. Finally, evaporation and diffusion play a role in the settling of the ink droplet on the porous substrate. Many experimental and computational studies have been done on the spreading and penetration of liquid droplets on solid and porous substrates. Von Bahr et al. (1999) have performed experiments on the spreading behaviour of microdrops of surfactant solution on solid surfaces, taking into consideration factors like surface tensions and drop lifetime prior to impact. Bacri and Brochard-Wyart (2000) conducted experiments of small and large liquid droplets on prewetted porous substrates, with focus on the effect of the dynamic contact angle of the droplets. Schoelkopf et al. (2000) present experiments in which a wetting fluid was applied to a porous cube of calcium carbonate to study the rate of uptake of the wetting fluid. Furthermore, they used the Pore-Cor software package to approach the same problem computationally. Starov et al. (2002a), Starov et al. (2002b), and Starov et al. (2002c) have conducted experiments on the simultaneous spreading and penetration of a liquid droplet into saturated, and thick and thin dry porous substrates. Dam and Le Clerc (2004) performed experiments of the impact of inkjet printed droplets with radii in the order of tens of micrometers on solid substrates. Tan (2017) did experimental and numerical studies of the absorption of small droplets into porous substrate in which the impacting, spreading and imbibing of droplets on the porous substrate was visualized, and a computational fluid dynamics solver was applied to simulate the droplet dynamics. They report good agreement between the numerical and experimental data. Yin et al. (2018) present modeling work in which they employ pore-network modeling to simulate the imbibition of a liquid droplet into a paper coating layer. Aslannejad et al. (2018) have performed both experimental observations and pore-scale modeling of the spreading and penetration of a droplet on and into the fibrous layer of paper. Furthermore, work has been done to characterize the hydraulic properties of uncoated paper (Aslannejad and Hassanizadeh, 2017), coated paper (Aslannejad et al., 2017) and of the interface between the coated and fibrous layers of coated paper (Aslannejad et al., 2019a).

Due to its importance on printing quality in the application of inkjet printing, this thesis aims at performing pore-scale simulations of droplet spreading and penetrating into the coating layer of paper. Pore-scale simulations have the potential of revealing processes at the pore-scale that would otherwise be missed in the case of macro-scale modelling. Furthermore, the thickness of coating layer of the printing paper is in the range of 1–25 $[\mu m]$ (Aslannejad et al., 2019a) and mean pore-size of 200 $[nm]$. Taking into account that the REV size of the coating layer is $4 \cdot 4 \cdot 4$ $[\mu m]$ (Aslannejad et al., 2017), then in some locations along the coating layer there are not enough REVs across the cross section of the layer and doing macro-scale simulation for such a case is less meaningful.

The OpenFOAM (Open source Field Operation And Manipulation) modelling package includes solvers for many types of computational fluid dynamics (CFD) problems, such as (heat) transport, turbulence modeling and multi-phase flow problems. Since the problem of liquid droplet spreading is inherently a two-phase flow problem, the latter application of OpenFOAM is of particular interest in this thesis. The main solver for approaching two-phase flow problems with OpenFOAM is the *interFoam* solver. This solver

assumes incompressible, immiscible and isothermal flow. The *interFoam* solver applies the Volume of Fluid (VoF) method, in which the two phases under consideration are tracked by employing a phase fraction indicator function. The Volume of Fluid method has been applied in many studies on microfluidics and porous media. Ashish Saha and Mitra (2009) applied the VoF method alongside experimental methods to study the impact of different dynamic contact angle models on the flow in a microfluidic capillary. Malgarinos et al. (2014) simulated the spreading of droplets on a solid surface with dynamic contact angle effects using the VoF method. Ferrari et al. (2015) present a study of two-phase flow experiments in porous micromodels, employing the VoF method to test numerical simulations with experimental results. Linder et al. (2015) used the VoF method to study the motion of a liquid droplet on a solid surface, taking into account dynamic contact angles. Arias and Montlaur (2018) have studied the effect of a contact angle boundary condition in the case of bubble generation inside of a capillary T-junction, also employing the VoF method. Shams et al. (2018) present a study of the application of the VoF method on two-phase flow in porous media at the micro-scale.

1.1 Research Question & Hypothesis

In order to simulate the spreading and penetration of liquid droplets into the coating layer of printing paper, the VoF method was used in this thesis. The *interFoam* solver of the OpenFOAM software package was employed on a paper coating geometry obtained by Focused Ion Beam Scanning Electron Microscopy (FIB-SEM) imaging (Aslannejad et al., 2017). In particular, the effect of the equilibrium contact angle of the liquid droplet was studied. The primary aim of this thesis was to answer the following research question: *"What is the effect of the equilibrium contact angle on the spreading and penetration of a small liquid droplet on a paper coating layer?"* The hypothesis was that although a lower equilibrium contact angle would result in faster penetration, it would also contribute positively to the spreading of the droplet on the surface, due to the higher wettability of the coating layer. The goal of this work was to find out the effect of the contact angle on the resulting penetration depth and spreading of ink on the surface of printing paper. Since the penetration and spreading extend directly dictates the quality of the print, pore-scale modelling of ink movement into the coating layer of paper yields knowledge needed to optimize ink and substrate properties to reach the highest quality of print with the lowest amount of ink.

2 Theory

2.1 Inkjet Printing

There are two commonly used methods of forming droplets in inkjet printing: the continuous inkjet (CIJ) technique and the drop-on-demand (DOD) technique (Kettle et al., 2010). In continuous inkjet printing, droplets are continuously being created. However, since droplets are not constantly needed during the printing process, most of the droplets are deflected back to the ink circulations or into a waste reservoir. The remaining droplets are let through and end up on the substrate. In the drop-on-demand method of inkjet printing, droplets are only released when they are actually required on the substrate (Le, 1998).

During inkjet printing on a porous substrate, several phenomena occur that are important for the evolution of the ink droplet on and into the substrate (Kettle et al., 2010). When the ink droplet first reaches the surfaces, the droplet wets the surface and inertial forces dominate the spreading and penetration of the droplet (Schoelkopf et al., 2000). During these first moments of contact, the contact angle of the ink droplet with the porous substrate oscillates. The dampening of this oscillation depends on the droplet's viscosity and size, but this period of oscillation generally lasts about 8 ms (Von Bahr et al., 2003). After some time, a change will occur from a regime where inertial flow dominates to one where capillary flow becomes more important. When this change occurs depends strongly on the radius of the pores, ranging from only 0.01 μs for a pore radius of 0.1 μm to 0.01 s for a pore radius of 1 mm (Schoelkopf et al., 2000). As capillary penetration continues, the separation of ink components begins around a few milliseconds after the droplet arrives at the surface. Adsorption of the ink components onto the surface of the porous substrate start becoming increasingly significant about 1 second after the arrival of the droplet. Experimental studies by Von Bahr et al. (1999) indicate that spreading of microdrops of surfactant solution can be subdivided into two regimes with different characteristics, a non-diffusive regime and a diffusive regime. The non-diffusive regime was found to be dominated by inertial, gravitational and capillary forces, where droplet spreading occurs very fast. The diffusive regime is dominated by diffusion of surfactants towards the interface, and starts around two seconds after the droplet has arrived on the surface. The complete drying of the ink takes much longer, and can take multiple hours to finalise (Kettle et al., 2010).

2.2 Spreading and Penetration of Droplets

The spreading and penetration of ink droplets on and into a paper coating are very important for the print quality achieved with inkjet printing. Coated papers are used to produce high print quality by reducing penetration of ink into the underlying fibrous layer (Aslannejad et al., 2019a). Furthermore, coalescence of ink droplets during the process of inkjet printing can cause the formation of large pools of liquid on the printing surface, which is detrimental to the printing results (Daniel and Berg, 2006). Therefore, this section will provide a short review of the mechanisms behind the spreading and penetration of liquid droplets into porous substrates.

2.2.1 Droplet Spreading

When a liquid droplet, surrounded by gas, lies on top of a solid surface, this system contains three surfaces where the three different phases are in contact with each other: the liquid-solid interface, the liquid-gas interface, and the gas-solid interface. This system will attempt to minimize the combined surface free energy of these three surfaces, given a constant volume of the liquid droplet. The Helmholtz surface free energy of the droplet is given as:

$$F = \gamma_{\mathrm{ls}} A_{\mathrm{ls}} + \gamma_{\mathrm{lg}} A_{\mathrm{lg}} + \gamma_{\mathrm{gs}} A_{\mathrm{gs}} \tag{1}$$

where γ_{ls}, γ_{lg} and γ_{gs} are the surface energies of the three interfaces, and A_{ls}, A_{lg} and A_{gs} are the surface areas of the interfaces. Assuming that the shape of the droplet can be approximated with that of a spherical cap whose base has a radius of a, and that the droplet has a constant volume, the derivative of the Helmholtz surface free energy with respect to a becomes (Daniel and Berg, 2006):

$$\frac{dF}{da} = 2\pi a \left(\cos(\theta) - \frac{\gamma_{\mathrm{gs}} - \gamma_{\mathrm{ls}}}{\gamma_{\mathrm{lg}}} \right) \gamma_{\mathrm{lg}} \tag{2}$$

where θ is the angle between the liquid-solid and liquid-gas interfaces at the contact line. For the case of minimum free surface energy the right-hand-side of equation (2) should equal 0, and this is thus achieved when:

$$\cos(\theta_{\mathrm{eq}}) = \frac{\gamma_{\mathrm{gs}} - \gamma_{\mathrm{ls}}}{\gamma_{\mathrm{lg}}} \tag{3}$$

Equation (3) is called the Young-Dupré equation, which expresses the equilibrium contact angle of the droplet in terms of surface energies. The equilibrium contact angle between a solid and liquid, given a particular third phase (often air), gives an indication of how wettable the solid is for that particular liquid. The lower the contact angle, the higher the wettability of the solid, as the liquid has a higher tendency to spread out over the solid in order to reach a state of minimum free surface energy.

The equilibrium contact angle is, however, not enough to fully describe the behaviour of a droplet on a solid surface. In reality, the contact angle of a droplet has a dynamic nature, varying in value within a range of different values, from the advancing contact angle, to the receding contact angle. The difference between the advancing and receding contact angles is called contact angle hysteresis (Johnson Jr. and Dettre, 1964), (Huh and Scriven, 1971). This contact angle hysteresis is due to the fact that besides the surface tensions, the three-phase contact line also has an energy associated with it which needs to be considered (Tadmor, 2004). This contact line energy arises from roughness and heterogeneities of the solid surface, leading to 'pinning' of the contact line (De Gennes, 1985).

2.2.2 Droplet Penetration

If the surface that a liquid droplet is introduced to is not solid, but has a porous structure underneath, not only will the droplet spread over the surface, but it will also start penetrating into the porous substrate due to inertial, gravitational and capillary effects. The relevance of the aforementioned processes to the penetration of a liquid droplet into a porous substrate can be expressed by some dimensionless numbers. The Bond number describes the magnitude of gravitational forces with respect to surface tension forces (Tan, 2017) and can be expressed as:

$$Bo = \frac{\rho g r^2}{\gamma_{\mathrm{lg}}} \tag{4}$$

where ρ is the density of the liquid, g is the gravitational acceleration, and r is the droplet radius. A Bond number smaller than 1 is an indication that capillary forces are dominating with respect to gravitational forces. Inertial forces can be related to surface tension forces by the Weber number (Kettle et al., 2010), which can be calculated as:

$$We = \frac{\rho u^2 l}{\gamma_{\mathrm{lg}}} \tag{5}$$

where u is the droplet's impact velocity and l is the characteristic length of the fluid flow. A small Weber number is an indication that inertial effects are relatively small compared to capillary forces.

The Young-Laplace equation can be used to obtain

the capillary pressure in a vertical capillary tube with radius r (Schoelkopf et al., 2000):

$$P_c = \frac{2\gamma_{\text{lg}} \cos\theta_{\text{eq}}}{r} \tag{6}$$

where θ_{eq} is the equilibrium contact angle between the fluid meniscus and the wall of the capillary tube. The capillary pressure is the pressure difference over this meniscus when the liquid in the capillary tube is in equilibrium. This capillary pressure can be used to find the height z of a meniscus in a capillary of radius r when the wetting force and gravity are in equilibrium. The wetting force is given as $F_c = \pi r^2 P_c$, while the gravitational force is $F_g = mg$, where $m = \rho V$ and g is the gravitational acceleration. For a cylindrical capillary, V can be approximated as $\pi r^2 z$, where z is the height of the meniscus. Thus, when the two forces are in equilibrium, the following equivalence holds:

$$F_c = F_g \tag{7}$$

$$2\pi r\gamma_{\text{lg}} \cos(\theta_{\text{eq}}) = \rho\pi r^2 zg \tag{8}$$

$$z = \frac{2\gamma_{\text{lg}} \cos(\theta_{\text{eq}})}{\rho g r} \tag{9}$$

Lucas (1918) & Washburn (1921) have presented a way of expressing the position of a wetting meniscus in a capillary with respect to time, ignoring gravitational effects:

$$x(t) = k\sqrt{t} \tag{10}$$

where k is given as:

$$k = \sqrt{\frac{r\gamma_{\text{lg}} \cos(\theta_d)}{2\mu}} \tag{11}$$

for Poiseuille flow in a cylindrical and horizontally orientated capillary with radius r, and where θ_d is the dynamic contact angle, μ is the dynamic viscosity. The Lucas-Washburn equation is, however, not enough to accurately describe the processes on short time scales and with fine pore sizes which are encountered when considering penetration into paper coating. Generally this is due to the fact that the Lucas-Washburn equation does not take inertial effects into consideration (Ridgway et al., 2002). Furthermore, the equation predicts an infinite velocity at $t = 0$ (Sorbie et al., 1995). Therefore, several modifications to the Lucas-Washburn equation have been proposed to account for this. Bosanquet (1923) modified the equation to include an inertial wetting term. Szekely et al. (1971) extended the Lucas-Washburn equation by incorporating a macroscopic energy balance in which the total kinetic and potential energy within the system are taken into consideration.

2.2.3 Simultaneous Spreading & Penetration

During the settling of a liquid droplet on a porous substrate, the spreading and penetration of the droplet are two competing processes occurring simultaneously. Therefore, in order to fully understand the processes of droplet spreading and penetration on and into coated paper, the two processes should not only be understood separately, but the interplay between the two processes should also be examined. Starov et al. (2002a) present a hydrodynamic model describing simultaneous spreading and penetration on a porous substrate, however, in their consideration, the porous layer into which penetration takes place is much thinner than the height of the droplet. This is not the case in inkjet printing on coated paper, where droplet volumes range from 10–20 pL (Calvert, 2001), which translates to 13.37–16.84 μm assuming a spherical droplet, while Aslannejad et al. (2017) report the thickness of a paper coating layer as 12.8 μm. Starov et al. (2002a) divide the motion of the droplet (here defined as the part of the liquid that resides on top of the surface of the porous layer, and thus has not yet penetrated the substrate) into two parts, the droplet spreading over the parts of the porous substrate that is saturated, and the shrinking of the droplet due to imbibition into the substrate. This results in the following equation describing the change in droplet base radius a over time:

$$\frac{da}{dt} = v_+ - v_- \tag{12}$$

where v_+ and v_- are the expansion and shrinkage velocities of the droplet base, respectively. From their assumptions they derive the following expression for the dynamic contact angle θ_d:

$$\theta_d = \frac{4V_d}{\pi a^3} \tag{13}$$

where V_d is the droplet volume. The radius of the base of the droplet can therefore be expressed as a function of droplet volume and dynamic contact angle:

$$a = \left(\frac{4V_d}{\pi\theta_d}\right)^{\frac{1}{3}} \tag{14}$$

Since both the dynamic contact angle and the droplet volume vary through time (since droplet volume is lost as the liquid penetrates into the porous substrate), the derivative of the droplet base radius over time can be expressed as:

$$\frac{da}{dt} = -\frac{1}{3}\left(\frac{4V_d}{\pi\theta_d^4}\right)^{\frac{1}{3}} \frac{d\theta_d}{dt} + \frac{1}{3}\left(\frac{4}{\pi V_d^2 \theta_d}\right)^{\frac{1}{3}} \frac{dV_d}{dt} \tag{15}$$

From this equation, combined with equation (12), expressions for the expansion and shrinkage velocities of the droplet base can be derived:

$$v_+ = -\frac{1}{3}\left(\frac{4V_d}{\pi\theta_d^4}\right)^{\frac{1}{3}}\frac{d\theta_d}{dt} \tag{16}$$

$$v_- = -\frac{1}{3}\left(\frac{4}{\pi V_d^2\theta_d}\right)^{\frac{1}{3}}\frac{dV_d}{dt} \tag{17}$$

According to Starov et al. (2002a), the droplet spreading process can be subdivided into two stage, a fast stage where there is primarily spreading and penetration into the porous substrate can be ignored, and where the droplet volume is thus approximately constant, followed by a slower stage where there is only little spreading occurring and penetration becomes the dominating process. Under the assumption of the first stage of spreading they find the following expression for the expansion of the droplet base radius:

$$v_+ = 0.1\left(\frac{4V_d}{\pi}\right)^{0.3}\left(\frac{10\gamma_{\mathrm{lg}}\omega}{\mu}\right)^{0.1}(t+t_0)^{-0.9} \tag{18}$$

where t_0 is the duration of the initial stage of spreading and ω is an effective lubrication parameter. The volume of the droplet on top of the porous substrate can be expressed through time as:

$$V_d(t) = V_0 - \phi\pi h l(t)^2 \tag{19}$$

where the volume of wetted porous substrate is expressed as a cylinder with height h and radius $l(t)$, and where ϕ is the porosity of the substrate. This expression can be substituted into equation (17) in order to express the velocity of the droplet base shrinkage in terms of only the radius of wetted substrate:

$$v_- = \frac{2\pi^{\frac{2}{3}}\phi h l}{3}\left(\frac{4}{(V_0-\pi\phi h l^2)^2\theta_d}\right)^{\frac{1}{3}}\frac{dl}{dt} \tag{20}$$

For the spreading of the fluid within the porous substrate $\frac{dl}{dt}$ the Darcy equation was solved and the following expression was found:

$$\frac{dl}{dt} = \frac{K_p P_c}{\mu l\frac{l}{a}} \tag{21}$$

where K_p is the permeability of the porous substrate (assumed to be homogeneous and isotropic). From these expressions for the evolution of the droplet radius on top of and inside of the porous substrate, they conclude that the first stage of the spreading process is very short, and that capillary spreading is more dominant in this stage than shrinkage of the droplet base due to imbibition. Furthermore, they find that in the second stage the spreading of the droplet stops almost completely, and here the shrinkage of the droplet base is determined by imbibition from the droplet into the substrate.

Clarke et al. (2002) presented a theory of simultaneous spreading and absorption of droplets into porous substrates based on the molecular-kinetic theory. Here, the radial velocity of the wetting line can be related to the dynamic contact angle, and Darcy's law is used to incorporate the process of absorption into the substrate.

Another way of describing simultaneous spreading and penetration is that of Daniel and Berg (2006), who used an energy based approach. Firstly, the Helmholtz free energy balance of the system is considered:

$$\frac{dF}{dt} = \gamma_{\mathrm{lg}}\frac{dA_{\mathrm{lg}}}{dt} + \gamma_{\mathrm{ls}}\frac{dA_{\mathrm{ls}}}{dt} + \gamma_{\mathrm{gs}}\frac{dA_{\mathrm{gs}}}{dt} \tag{22}$$

Since evaporation is not taken into account, the total volume of fluid in the system remains constant, and thus:

$$V_i(t) = V_0 - V_d(t) \tag{23}$$

where V_i is the imbibed volume, V_0 is the volume of the droplet before penetration starts, and V_d is the volume of the droplet. The overall evolution of the droplet base is described in a similar fashion as seen in equation (12), however, here the shrinkage velocity v_- is allowed to be either negative or positive depending on its mathematical expression:

$$\frac{da}{dt} = v_+ + v_- \tag{24}$$

where a is again the droplet base radius. In their approach, Daniel and Berg (2006) assume that the shape of the droplet is that of a spherical cap, and that during the penetration and spreading of the droplet it remains its shape. Bacri and Brochard-Wyart (2000) have shown experimentally that this assumption is reasonable, at least for droplets of approximately mm size. In order to maintain the spherical cap geometry of the droplet. the assumption is made that all points along the drop sink at the same rate. From this it follows that:

$$\frac{dh}{dt} = \frac{1}{\pi a^2}\frac{dV_d}{dt} \tag{25}$$

where h is the height of the droplet. The volume of a spherical cap can be expressed in terms of its base radius and its height:

$$\frac{6}{\pi}V_d = h(3a^2+h^2) \tag{26}$$

And thus the change in droplet volume through can be expressed purely in terms of its base radius and height and the change of those variables through time:

$$\begin{aligned}\frac{6}{\pi}\frac{dV_d}{dt} &= \frac{d}{dt}\left(3ha^2+h^3\right)\\ &= 6ah\frac{da}{dt}+3\left(a^2+h^2\right)\frac{dh}{dt}\end{aligned} \tag{27}$$

By neglecting spreading effects, and thus associating $\frac{da}{dt}$ directly with v_-, they find the following expression for the shrinkage velocity:

$$v_- = \frac{\cot\left(\theta_d\right)}{\pi a^2}\frac{dV_d}{dt} \tag{28}$$

The interfacial areas found in equation (22) can be expressed as follows:

$$A_{\mathrm{lg}} = \pi\left(a^2+h^2\right)+\phi f_L \pi a^2 \tag{29}$$

$$A_{\mathrm{ls}} = \pi\left(1-\phi\right)a^2+\phi f_S\pi a^2+\Lambda\left(V_0-V_d\right) \tag{30}$$

$$\begin{aligned}A_{\mathrm{gs}} = {} & A_{\mathrm{tot}}-\pi\left(1-\phi\right)a^2-\phi f_S\pi a^2\\ & +\Lambda\left(V_{\mathrm{tot}}-V_0+V_d\right)\end{aligned} \tag{31}$$

where ϕ is the porosity of the porous substrate, f_L and f_S are factors correcting the interfacial areas for the curvature of the meniscus, and Λ is the ratio of pore surface are to void volume, which if the pores are assumed to be cylindrical can be expressed in terms of the pore radius r_c as:

$$\Lambda = \frac{2}{r_c} \tag{32}$$

Under the assumption that the meniscus area scales directly to the base area of the drop, as opposed to equating it to the total meniscus are in the porous substrate, substituting the time derivatives of equations (29-31) into equation (22) gives:

$$\begin{aligned}\frac{dF}{dt} = {} & 2\pi a\gamma_{\mathrm{lg}}\left(\cos\left(\theta_d\right)-\cos\left(\theta_{\mathrm{eq}}^{\mathrm{eff}}\right)\right)v_+\\ & +\left(\frac{2\gamma_{\mathrm{lg}}\sin\left(\theta_d\right)}{a}+\Lambda\gamma_{\mathrm{lg}}\left(\bar{s}+1\right)\right)\frac{dV_d}{dt}\end{aligned} \tag{33}$$

where $\frac{da}{dt}$ has been associated with the spreading velocity v_+, $\bar{s}$ is a normalized spreading coefficient:

$$\bar{s} = \frac{\gamma_{\mathrm{gs}}-\gamma_{\mathrm{ls}}-\gamma_{\mathrm{lg}}}{\gamma_{\mathrm{lg}}} \tag{34}$$

and where $\theta_{\mathrm{eq}}^{\mathrm{eff}}$ is the equilibrium contact angle in the case of no penetration. For a non-porous substrate it holds that $\cos\left(\theta_{\mathrm{eq}}^{\mathrm{eff}}\right) = \bar{s}+1 = \cos\left(\theta_{\mathrm{eq}}\right)$. Since $\cos\left(\theta_{\mathrm{eq}}\right) = \bar{s}+1$, equation (33) can be slightly modified to:

$$\begin{aligned}\frac{dF}{dt} = {} & 2\pi a\gamma_{\mathrm{lg}}\left(\cos\left(\theta_d\right)-\cos\left(\theta_{\mathrm{eq}}^{\mathrm{eff}}\right)\right)v_+\\ & +\left(\frac{2\gamma_{\mathrm{lg}}\sin\left(\theta_d\right)}{a}+\Lambda\gamma_{\mathrm{lg}}\cos\left(\theta_{\mathrm{eq}}\right)\right)\frac{dV_d}{dt}\end{aligned} \tag{35}$$

The change in Helmholtz free energy can also be described in terms of the dissipative processes in the system. It is assumed that during droplet spreading and penetration, the total dissipation of Helmholtz free energy can be summed up as the dissipation due to the radial spreading of the droplet and due to the capillary flow inside the porous substrate. This is expressed as:

$$\frac{dF}{dt} = -\mu\int\Phi_d dV-\mu\int\Phi_p d\left(V_0-V_d\right) \tag{36}$$

where $\mu\Phi_d$ and $\mu\Phi_p$ are the viscous dissipation in the droplet (on top of the surface) and inside of the porous substrate, respectively. Under the assumption that the droplet spreading can be approximated as a cylindrical pancake of fluid, the viscous dissipation of the droplet can be described as:

$$\mu\int\Phi_d dV = \frac{6\pi\omega\mu a^2 v_+^2}{gh} \tag{37}$$

where ω is a viscosity multiplier, and g is a factor defined as:

$$g = \left(\frac{1}{2}+\frac{1}{6}\left(\frac{h}{a}\right)^2\right) \tag{38}$$

Under Poiseuille flow, the viscous dissipation inside a single pore of the porous substrate is expressed as:

$$\mu\int\Phi_{\mathrm{pore}}dV_{\mathrm{pore}} = 4\pi\mu k^2 w\left(t-\tau\right) \tag{39}$$

where τ is the time at which the liquid first enters the particular pore, k is the Washburn coefficient as seen in equation (10) and where:

$$w\left(t\right) = \frac{dz\left(t\right)}{dt} \tag{40}$$

where $z\left(t\right)$ is the depth of penetration into the capillary in accordance with the Lucas-Washburn equation. If it assumed that the total number of pores into which the liquid flows is determined by the radius of the base of the droplet a, then the total viscous dissipation inside of the porous substrate can be found by integrating over all pores that are in contact with the droplet:

$$\mu\int\Phi_p d\left(V_0-V_d\right) = \frac{4\mu k^2}{r_c^2}\left(2\pi\phi\int_0^a xw\left(t-\tau\right)dx\right) \tag{41}$$

Given equation (23), an approximation of the volume of the droplet can be given as:

$$V_d(t) = V_0-V_i = V_0-2\pi\phi\int_0^a xz\left(t-\tau\right)dx \tag{42}$$

under the assumption that the process of penetration is axisymmetric. It then follows that:

$$\frac{dV_d}{dt} = -2\pi\phi \int_0^{a(t)} xw(t-\tau)dx \tag{43}$$

Substituting the equation above and equation (32) into equation (41), and substituting for the Washburn coefficient, the following expression for the viscous dissipation in the porous substrate is found:

$$\mu \int \Phi_p d\left(V_0 - V_d\right) = -\Lambda\lambda_{\mathrm{lg}} \cos\left(\theta_{\mathrm{eq}}\right) \frac{dV_d}{dt} \tag{44}$$

Substituting the expressions for the viscous dissipation of the droplet and in the porous substrate into equation (36) expressing the change in Helmholtz free energy in terms of dissipation, and combining it with equation (35), the following equality is found:

$$\frac{6\pi\omega\mu a^2 v_+^2}{gh} - 2\pi a\gamma_{\mathrm{lg}}\left(\cos\left(\theta_{\mathrm{eq}}^{\mathrm{eff}}\right) - \cos\left(\theta_d\right)\right) v_+ - \frac{2\gamma_{\mathrm{lg}}\sin\left(\theta_d\right)}{a}\frac{dV_d}{dt} = 0 \tag{45}$$

which is a quadratic equation that can be solved to find the spreading velocity v_+:

$$v_+ = \frac{\gamma_{\mathrm{lg}} gh}{3\mu\omega a}\left(\cos\left(\theta_{\mathrm{eq}}^{\mathrm{eff}}\right) - \cos\left(\theta_d\right)\right) + \left[\left(\frac{\gamma_{\mathrm{lg}} gh}{3\mu\omega a}\left(\cos\left(\theta_{\mathrm{eq}}^{\mathrm{eff}}\right) - \cos\left(\theta_d\right)\right)\right)^2 - \frac{2\gamma_{\mathrm{lg}} gh\sin(\theta_d)}{3\mu\omega\pi a^3}\frac{dV_d}{dt}\right]^{\frac{1}{2}} \tag{46}$$

This expression for the droplet spreading velocity, combined with equation (17) describing the droplet shrinkage velocity and equation (24), describes the change in droplet base radius through time due to simultaneous droplet spreading and penetration. Here, the rate of growth of the droplet base radius is dependent only on the droplet base radius, the droplet volume, and the rate of volume loss due to imbibition. Therefore, Daniel and Berg (2006) argue that the mode of penetration that is used to calculate the rate of penetration should not influence results significantly, as long as they approximate the rate of penetration well enough. They distinguish between two modes of penetration: vertical penetration and bi-directional penetration. The theory of vertical penetration they present is based on the penetration equation of Clarke et al. (2002), where the porous structure is approximated as a collection of vertical capillaries. The theory of bi-directional penetration is based on Darcy's law. In the case of vertical penetration, radial wicking is ignored, however it is argued that this approach is reasonable since droplet spreading occurs simultaneously with radial wicking, especially during early times when spreading is the dominating process.

2.3 Governing Equations of Flow on the Pore Scale

When looking at a domain Ω with a volume of $\mathbb{V}$ and bounded by a surface S, one can describe the balance of any extensive quantity E in this domain as follows: the rate at which E accumulates within Ω must be equal to the sum of the net influx of E into Ω through the bounding surface S and the rate at which E is produced within Ω. These three terms can also be described mathematically. The rate of accumulation of E within Ω then becomes:

$$\frac{\partial}{\partial t}\int_{\mathbb{V}} e' d\mathbb{V} = \int_{\mathbb{V}} \frac{\partial e'}{\partial t} d\mathbb{V} \tag{47}$$

where e' is the amount of E per unit volume (also known as the *density* of E). The net influx of E into Ω can be described as:

$$-\int_S e'\vec{U}^E \cdot \vec{\nu} dS \tag{48}$$

where $\vec{\nu}$ is the outward facing unit normal vector of the infinitesimally small areas dS which make up the entire surface of S, and $\vec{U}^E$ is the velocity of E. Consequently $e'\vec{U}^E$ is then the flux of E (velocity of E multiplied with its density, resulting in unit of E per unit area per unit time), and $e'\vec{U}^E \cdot \vec{\nu}$ the component of the flux going out of Ω (hence the minus sign). Finally, the production of E within Ω can be mathematically described by:

$$\int_{\mathbb{V}} \rho\Gamma^E d\mathbb{V} \tag{49}$$

where Γ^E is the rate of internal production of E per unit mass of the phase, and ρ is the mass density of the phase. Combining these three components then gives the entire balance of E within Ω in mathematical form:

$$\int_{\mathbb{V}} \frac{\partial e'}{\partial t} d\mathbb{V} = -\int_S e'\vec{U}^E \cdot \vec{\nu} dS + \int_{\mathbb{V}} \rho\Gamma^E d\mathbb{V} \tag{50}$$

Using Gauss' Theorem the surface integral of the net influx of E into Ω can be rewritten into a volume integral as follows:

$$\int_S e'\vec{U}^E \cdot \vec{\nu} dS = \int_{\mathbb{V}} \vec{\nabla} \cdot e'\vec{U}^E d\mathbb{V} \tag{51}$$

Substituting this into (50) gives:

$$\int_{\mathbb{V}} \left(\frac{\partial e'}{\partial t} + \vec{\nabla} \cdot e' \vec{U}^E - \rho \Gamma^E \right) d\mathbb{V} = 0 \qquad (52)$$

Finally, the volume $\mathbb{V}$ can be shrunk to zero around some Eulerian coordinate $\vec{x}$ within Ω to get the *microscopic level E-balance equation* for this point $\vec{x}$:

$$\frac{\partial e'}{\partial t} + \vec{\nabla} \cdot e' \vec{U}^E - \rho \Gamma^E = 0 \qquad (53)$$

Splitting up the flux ($\vec{j}^E = e' \vec{U}^E$) into an *advective* ($\vec{j}^E_{\text{adv}}$) and a *diffusive* ($\vec{j}^E_{\text{dif}}$) part, the balance equation can be rewritten to:

$$\frac{\partial e'}{\partial t} = -\vec{\nabla} \cdot \left(\vec{j}^E_{\text{adv}} + \vec{j}^E_{\text{dif}} \right) + \rho \Gamma^E \qquad (54)$$

The advective flux is defined as the flux of E due to being carried by a phase with a certain (mass-weighted) velocity ($\vec{U}$) and can be expressed as $\vec{j}^E_{\text{adv}} = e' \vec{U}$. The diffusive flux is defined as the flux of E relative to the advective flux of E, thus any part of the flux that is not due to movement of the phase. This can be expressed as $\vec{j}^E_{\text{dif}} = e' \left(\vec{U}^E - \vec{U} \right)$.

The microscopic level E-balance equation can be used to obtain balances for specific extensive quantities of a fluid, such as the mass balance equation and the Navier-Stokes (linear momentum balance) equations.

2.3.1 The Microscopic Mass Balance Equation

When speaking of the mass (m) of a fluid phase, E can be substituted with m in the above described E-balance equation. From $E = m$ it follows that $e' = \frac{E}{\mathbb{V}} = \frac{m}{\mathbb{V}} = \rho$, thus the density of m is the mass density of the phase. The *specific value* (e), which is defined as the amount of unit E per unit mass, becomes $e = \frac{m}{m} = 1$. Since mass cannot be created or vanish, the internal production rate must be zero: $\Gamma^E = \Gamma^m = 0$. Furthermore, for the mass, $\vec{U}^E = \vec{U}^m$, which is also the mass-weighted velocity ($\vec{U}$) of the phase. Therefore the diffusive flux becomes zero:

$$\vec{j}^m_{\text{dif}} = e' \left(\vec{U}^m - \vec{U} \right) = \rho \left(\vec{U} - \vec{U} \right) = 0 \qquad (55)$$

The resulting mass balance equation is then:

$$\frac{\partial \rho}{\partial t} = -\vec{\nabla} \cdot \rho \vec{U} \qquad (56)$$

For incompressible flow the density is constant and the above expression simplifies to:

$$\vec{\nabla} \cdot \vec{U} = 0 \qquad (57)$$

2.3.2 The Navier-Stokes Equation for Incompressible Flow

The microscopic level E-balance equation can also be applied to the linear momentum ($\vec{M} = m\vec{U}$) of the fluid phase. From $E = m\vec{U}$ it follows that $e = \frac{E}{m} = \vec{U}$ and $e' = \frac{E}{\mathbb{V}} = \frac{m\vec{U}}{\mathbb{V}} = \rho \vec{U}$. The source of linear momentum in a unit volume is the total force acting on that volume, thus $\Gamma^E = \Gamma^M = -\vec{F}_b$, where $\vec{F}_b$ is the body force working on the fluid phase per unit mass [$L\ T^{-2}$].

The diffusive flux of linear momentum can be described as:

$$\vec{j}^M_{\text{dif}} = \rho \vec{U} \left(\vec{U}^M - \vec{U} \right) \qquad (58)$$

Since the diffusive flux of linear momentum is caused by energy dissipation due to stresses, the diffusive flux can also be described through the *stress tensor*:

$$\vec{j}^M_{\text{dif}} = -\boldsymbol{\sigma} \qquad (59)$$

with

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \qquad (60)$$

where σ_{ij} is the stress component with direction i working on the surface normal to j.

The advective flux of the linear momentum is:

$$\vec{j}^M_{\text{adv}} = e' \vec{U} = \rho \vec{U} \otimes \vec{U} \qquad (61)$$

Combining these terms into a complete momentum balance equation gives:

$$\frac{\partial \rho \vec{U}}{\partial t} = -\vec{\nabla} \cdot \left(\rho \vec{U} \otimes \vec{U} - \boldsymbol{\sigma} \right) + \rho \vec{F}_b \qquad (62)$$

The stresses working on a fluid can be divided into two parts, the *shear stresses* (σ_{ij} or τ_{ij}) with a direction parallel to the surfaces that they act upon (i.e. $i \neq j$), and the *normal stresses* (σ_{ij}) with a direction normal to the surfaces that they act upon (i.e. $i = j$). The normal stresses are the three diagonal components of the stress tensor $\boldsymbol{\sigma}$, and the *mean stress* is defined as a third of the sum of the normal stresses:

$$\sigma_m = \frac{1}{3} \left(\sigma_{xx} + \sigma_{yy} + \sigma_{zz} \right) \qquad (63)$$

This mean stress is equal to minus the pressure working on the fluid:

$$p = -\sigma_m = -\frac{1}{3} \left(\sigma_{xx} + \sigma_{yy} + \sigma_{zz} \right) \qquad (64)$$

In the absence of shear stresses, and assuming equal normal stresses (i.e. $\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = \sigma_m$), a *normal stress tensor* can then be defined as:

$$\boldsymbol{\sigma}_{\mathrm{n}} = \begin{bmatrix} -p & 0 & 0 \\ 0 & -p & 0 \\ 0 & 0 & -p \end{bmatrix} = -p\boldsymbol{\delta} \tag{65}$$

where p is the hydrostatic pressure and $\boldsymbol{\delta}$ is the *Kronecker delta*, or identity matrix. Similarly a *shear stress tensor* can be defined:

$$\boldsymbol{\tau} = \begin{bmatrix} 0 & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & 0 & \tau_{yz} \\ \tau_{zx} & \tau_{yz} & 0 \end{bmatrix} \tag{66}$$

The stress tensor can then be divided up into a shear stress and a normal stress part:

$$\boldsymbol{\sigma} = \boldsymbol{\tau} - p\boldsymbol{\delta} \tag{67}$$

Substituting equation (67) into equation (62) gives:

$$\frac{\partial \rho \vec{U}}{\partial t} = -\vec{\nabla} \cdot \rho \vec{U} \otimes \vec{U} - \vec{\nabla} \cdot \boldsymbol{\tau} + \vec{\nabla} p + \rho \vec{F}_b \tag{68}$$

since $\vec{\nabla} \cdot \boldsymbol{\delta} = \vec{\nabla}$.

To reduce the amount of variables that are needed to solve this momentum balance equation, the constitutive equation for the shear stress of a Newtonian fluid can be applied. This constitutive equation relates the shear stress to the dynamic viscosity of the fluid (μ) and the fluid's velocity, and reads as follows:

$$\boldsymbol{\tau} = \mu \left(\vec{\nabla} \otimes \vec{U} + \left(\vec{\nabla} \otimes \vec{U} \right)^T \right) \tag{69}$$

Given that the dynamic viscosity is constant, this then introduces the following term to the momentum balance equation:

$$-\mu \vec{\nabla} \cdot \left(\vec{\nabla} \otimes \vec{U} + \left(\vec{\nabla} \otimes \vec{U} \right)^T \right) \tag{70}$$

With some vector calculus it can be shown that:

$$\vec{\nabla} \cdot \left(\vec{\nabla} \otimes \vec{U} + \left(\vec{\nabla} \otimes \vec{U} \right)^T \right) = \vec{\nabla}^2 \vec{U} + \vec{\nabla} \left(\vec{\nabla} \cdot \vec{U} \right) \tag{71}$$

Thus, since for incompressible flow we have $\vec{\nabla} \cdot \vec{U} = 0$, the momentum balance equation can be rewritten in the following form:

$$\rho \frac{\partial \vec{U}}{\partial t} = -\rho \vec{\nabla} \cdot \vec{U} \otimes \vec{U} + \mu \vec{\nabla}^2 \vec{U} - \vec{\nabla} p + \rho \vec{F}_b \tag{72}$$

which is known as the Navier-Stokes equation for incompressible flow.

For two-phase flow, an additional term is required to account for the interfacial force ($\vec{F_\sigma}$) concentrated on the interface between the two phases (Tryggvason et al., 2006), (Deshpande et al., 2012), (Shams et al., 2018). One way that this interfacial force can be represented mathematically is presented by Deshpande et al. (2012), where it is expressed as:

$$\vec{F}_\sigma = \int_\Gamma \sigma \kappa \delta \left(\vec{x} - \vec{x}_s \right) \vec{n} \, d\Gamma \left(\vec{x}_s \right) \tag{73}$$

with Γ as the interface, σ the interfacial tension (assumed to be constant), κ the local interface curvature, $\delta \left(\vec{x} - \vec{x}_s \right)$ the three-dimensional Dirac delta function (equal to 1 on the interface and equal to 0 elsewhere), and $\vec{n}$ the unit vector normal to the interface. With this extra term added to include interfacial forces, the governing momentum balance equation then becomes:

$$\begin{aligned} \rho \frac{\partial \vec{U}}{\partial t} = & -\rho \vec{\nabla} \cdot \vec{U} \otimes \vec{U} + \mu \vec{\nabla}^2 \vec{U} - \vec{\nabla} p + \rho \vec{F}_b \\ & + \int_\Gamma \sigma \kappa \delta \left(\vec{x} - \vec{x}_s \right) \vec{n} \, d\Gamma \left(\vec{x}_s \right) \end{aligned} \tag{74}$$

2.4 The Volume of Fluid method

There are several methods to simulate two-phase flow. One of the most commonly used methods is the finite volume method (FVM) (Linder et al., 2015). In the finite volume method the governing equations are integrated over a small but finite control volume, after which the equations can be discretized (Versteeg and Malalasekera, 2007). The volume of fluid (VOF) method is an Eulerian approach to modeling two-phase flow that falls in the category of finite volume methods.

In the volume of fluid method, the interface between two phases is not explicitly tracked. Instead, an indicator function $\mathbb{I}(\vec{x}, t)$ is used, where in a domain containing an α-phase region (R_α) and a β-phase region (R_β) (Deshpande et al., 2012):

$$\mathbb{I}(\vec{x}, t) = \begin{cases} 1 & \text{for } \vec{x} \in R_\alpha \text{ at time } t \\ 0 & \text{for } \vec{x} \in R_\beta \text{ at time } t \end{cases} \tag{75}$$

This indicator function can then be used to assign a volume fraction (γ) of phase α to each computational cell Ω_i of the model, by integrating it over the corresponding cell:

$$\gamma(\vec{x}_i, t) = \frac{1}{|\Omega_i|} \int_{\Omega_i} \mathbb{I}(\vec{x}, t) d\mathbb{V} \tag{76}$$

From the expressions of the indicator function $\mathbb{I}$ and the volume fraction γ it follows that the volume fraction that is assigned to each cell must be equal to 1 if the cell is completely occupied by phase α, while conversely it must be equal to 0 if the cell is completely occupied by phase β. In cells where $0 < \gamma(\vec{x}_i, t) < 1$, an interface between phases α and β must be present, since both phases are present in the cell.

2.5 The *interFoam* Solver

In this section the specifics of the *interFoam* solver will be discussed. The standard *interFoam* solver assumes incompressible, immiscible, isothermal two-phase flow conditions, however, modified versions of the solver are available that support compressible and non-isothermal flow (*compressibleInterFoam*), miscible fluids (*interMixingFoam*), and n-phase flow (*multiphaseInterFoam*). Here, only the standard *interFoam* solver will be discussed.

2.5.1 Transport Equation of the Volume Fraction

The *interFoam* solver applies the volume of fluid method to solve the momentum balance and mass balance equations for two-phase flow. Every cell is assigned a volume fraction γ that determines whether or not an interface is present inside that cell. Since both phases are assumed to be incompressible, and thus have constant densities ρ_α and ρ_β, the density at every point in the domain can be expressed as:

$$\rho(\vec{x}, t) = \rho_\alpha \mathbb{I}(\vec{x}, t) + \rho_\beta (1 - \mathbb{I}(\vec{x}, t)) \tag{77}$$

Thus, although incompressible flow is assumed, the density in the domain is not constant but depends on the phase that is present at a certain point. This density can then be substituted in the mass balance equation (56) to get:

$$\begin{aligned} \frac{\partial(\rho_\alpha - \rho_\beta)\mathbb{I}(\vec{x}, t)}{\partial t} + \vec{\nabla} \cdot \left((\rho_\alpha - \rho_\beta)\mathbb{I}(\vec{x}, t)\vec{U} \right) \\ + \frac{\partial \rho_\beta}{\partial t} + \rho_\beta \vec{\nabla} \cdot \vec{U} = 0 \end{aligned} \tag{78}$$

Due to the assumption of incompressibility, the terms $\frac{\partial \rho_\beta}{\partial t}$ and $\rho_\beta \vec{\nabla} \cdot \vec{U}$ both equal zero, and thus what remains is:

$$(\rho_\alpha - \rho_\beta)\frac{\partial \mathbb{I}(\vec{x}, t)}{\partial t} + (\rho_\alpha - \rho_\beta)\vec{\nabla} \cdot \left(\mathbb{I}(\vec{x}, t)\vec{U} \right) = 0 \tag{79}$$

$$\frac{\partial \mathbb{I}(\vec{x}, t)}{\partial t} + \vec{\nabla} \cdot \left(\mathbb{I}(\vec{x}, t)\vec{U} \right) = 0 \tag{80}$$

Integrating equation (80) over a single computational cell Ω_i results in:

$$\int_{\Omega_i} \frac{\partial \mathbb{I}(\vec{x}, t)}{\partial t} d\mathbb{V} + \int_{\Omega_i} \vec{\nabla} \cdot \mathbb{I}(\vec{x}, t)\vec{U} d\mathbb{V} = 0 \tag{81}$$

which can be rewritten as

$$\frac{\partial}{\partial t} \int_{\Omega_i} \mathbb{I}(\vec{x}, t) d\mathbb{V} + \vec{\nabla} \cdot \int_{\Omega_i} \mathbb{I}(\vec{x}, t)\vec{U} d\mathbb{V} = 0 \tag{82}$$

Finally, substituting equation (76) into the equation above gives:

$$|\Omega_i| \frac{\partial \gamma(\vec{x}_i, t)}{\partial t} + |\Omega_i| \vec{\nabla} \cdot (\gamma(\vec{x}, t)\vec{U}) = 0 \tag{83}$$

or simply:

$$\frac{\partial \gamma}{\partial t} + \vec{\nabla} \cdot (\gamma \vec{U}) = 0 \tag{84}$$

The equation above describes the transport of the volume fraction γ.

2.5.2 The Continuum Surface Force (CSF) Model

Since the standard *interFoam* solver assumes incompressible flow, it makes use of the incompressible Navier-Stokes equation, modified to take into account forces at the interface (equation (74)). The method that the *interFoam* solver uses to solve the interfacial forces is the continuum surface force (CSF) model presentedby Brackbill et al. (1992). Here, instead of treating the interface as a discontinuous jump from one phase to the other, the interface is modelled as a continuous transition with a finite thickness h. The interfacial force then becomes a volume force for which holds that it must be equal to the surface force per unit area when $h \to 0$:

$$\lim_{h \to 0} \int_{\Delta V} \vec{F}_{\sigma,\mathrm{V}} dV = \int_{\Delta A} \vec{F}_{\sigma,\mathrm{A}} dA \tag{85}$$

For two fluids that are inviscid and have a constant surface tension, the surface force per unit area is:

$$\vec{F}_{\sigma,\mathrm{A}} = \sigma \kappa \vec{n} \tag{86}$$

Outside of the interfacial transition region, it should hold that $\vec{F}_{\sigma,\mathrm{V}} = 0$.

The CSF model is used to rewrite the expression of surface force over the interface in a computational cell as a volume integral of the whole cell:

$$\int_{\Gamma \cap \Omega_i} \sigma \kappa \vec{n} \; d\Gamma(\vec{x}_s) = \int_{\Omega_i} \sigma \kappa \vec{\nabla} \gamma \; dV \tag{87}$$

2.5.3 Discretisation of Governing Equations

The equations described above are used by the *interFoam* solver to calculate the velocities, pressures and volume fractions in each computational cell in the model for each timestep. However, as they are written above, the equations are not yet suitable to apply to the discrete cells that make up the spatial domain of a model applying the finite volume method, nor to the discrete timesteps over which these quantities are to be calculated. Therefore, the governing equations that are applied in *interFoam* and other OpenFOAM solvers are actually discretised versions of these equations. How exactly the equations are discretised depends very much on the discretisation schemes and solver algorithms that are used to solve a particular problem. An in depth formulation of the discretisation of the governing equations in the *interFoam* solver is described by Deshpande et al. (2012).

3 Methods

3.1 Fluid Properties

For the simulation of droplet spreading and penetration on a paper coating layer, two main types of liquid were used: water and an ink-like liquid. The properties that were used for these two liquids are shown in Table 1. The table also shows the properties of the surrounding air phase, used for both types of liquid droplets. The contact angles between the liquid droplets and the paper coating were varied in the simulations, but as a basis, a contact angle of 45 degrees was picked based on the measurements of Järnström et al. (2010).

Property	**Water**	**Ink-like liquid**	**Air**
Density	1000 [kg m^{-3}]	1050 [kg m^{-3}]	1 [kg m^{-3}]
Kin. viscosity	10^{-6} [m^2 s^{-1}]	$2 \cdot 10^{-6}$ [m^2 s^{-1}]	$1.48 \cdot 10^{-5}$ [m^2 s^{-1}]
Interfacial tension w/ air	0.0707106 [N m^{-1}]	0.03571 [N m^{-1}]	

Table 1: *Fluid properties of the water and ink-like droplets, and the properties of the surrounding air.*

3.2 Geometry and Mesh Generation

As the basis of the paper coating geometry to be used in the simulations, a piece of coated paper was imaged by Aslannejad et al. (2017), using focused ion beam scanning electron microscopy (FIB-SEM) imaging. They determined the porosity of the piece of coating layer that was imaged to be approximately 34%, and the permeability was determined to be 0.1 [mDarcy]. The representative elementary volume of the coating layer was found to be $4 \times 4 \times 4$ [μm^3]. The resulting images were then analysed using the Avizo Fire software, and the geometry was simplified by removing disconnected pores and smoothing its surfaces, in order to increase the quality of the mesh that will be used to run the simulations on. Therefore, it is possible that the porosity of the geometry that was used to create the mesh for the simulations had a slightly lower porosity than the value reported above. The geometry resulting from these steps is shown in Figure 1.

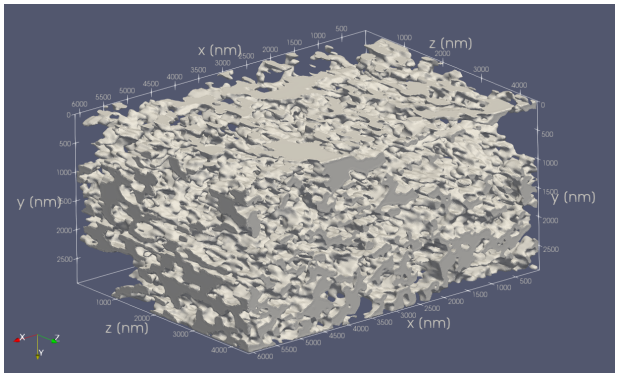


Figure 1: *Stereolithography file of the paper coating layer obtained by FIB-SEM imaging (Aslannejad et al., 2017).*

The droplets that were to be used in the simulations were to have a minimum radius of 2 [μm], however, due to the inclusion of a very large grain in the original geometry that was created from the FIB-SEM images, the geometry had to be cropped to a size of approximately $6 \times 2.9 \times 4.4$ [μm^3]. Some preliminary simulations indicated that this domain size was not large enough to contain the spreading and penetration of droplets with a radius of 2 [μm]. Since the REV size of the coating layer was $4 \times 4 \times 4$ [μm^3], then one can mirror the domain over its axes

to make the modelling geometry bigger. Therefore the domain was mirrored over the two axes that were aligned with the coating surface. This resulted in a geometry with a size of approximately $12.2 \times 2.9 \times 8.9$ [μm^3], which was sufficient to model the droplets with a radius of 2 [μm]. However, for the case of a droplet with a radius of 3 [μm], the depth of this geometry (2.9 [μm]) was not sufficient to contain the penetration of the droplet, and thus for the mesh of this simulation the geometry was mirrored once more in the vertical direction to obtain a size of 5.9 [μm]. This geometry, which was mirrored from Figure 1 in all three axes is shown in Figure 2.

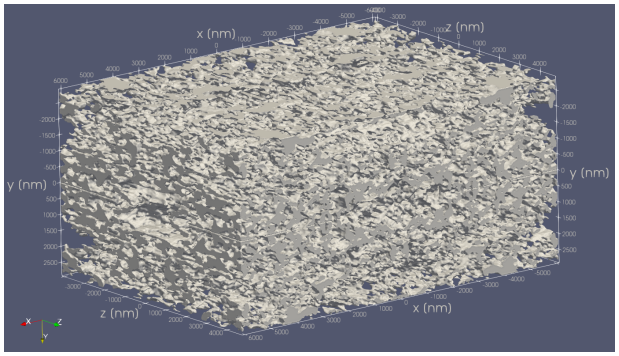


Figure 2: *Stereolithography file of the paper coating layer, mirrored over all three axes to account for droplets of at least 3 [μm] in radius.*

To convert this geometry into a mesh that could be used as the domain over which *interFoam* solves the governing flow equations, the *blockMesh* and *snappyHexMesh* utilities were used. The *blockMesh* utility can be used to create meshes with relatively simple geometries based on a set of blocks, each with 8 vertices. The edges of these blocks can be either straight lines, arcs or splines. For the purpose of this thesis, the *blockMesh* utility was employed to create a cubic mesh of which the grid cells were a constant size. This simple cubic mesh then served as the basis for the more complex mesh made to represent the micro-model shown above. This more complex mesh was generated using the *snappyHexMesh* utility.

The *snappyHexMesh* tool takes as input the geometry (in the form of a stereolithography (STL) or wavefront object (OBJ) files) and a base mesh. Depending on the selected options it then goes through four stages. Firstly, the base mesh is optionally refined around the features of the input geometry and/or in user specified regions. Parameters such as the level of refinement and the amount of cells between each refinement level can be specified by the user. Secondly, the cells in the mesh that are located inside of the geometry are removed from the mesh. After this, the cells of the mesh are "snapped" to the surface of the geometry, such that these surfaces are represented by the mesh as accurately as possible. As a final optional stage, the *snappyHexMesh* utility can add a layer of hexahedral cells along the boundary surface of the mesh to get rid of potential irregular cells created by the snapping process.

3.2.1 Mesh Quality Metrics

During all of its mesh generation stages, *snappyHexMesh* allows for certain mesh quality controls to be applied, which check if a generated mesh conforms to user specified parameters. If during a change in the mesh a cell or face is introduced that is of poor quality according to the specified quality controls, this change in the mesh is reverted. A poor quality mesh can have a huge influence in the obtained solution of a simulation, mainly through an increase in numerical diffusion, and it is thus important that a proper mesh is created before trying to run a model. Some important metrics that influence the quality of a mesh are its non-orthogonality, skewness, aspect ratio and smoothness (Holzinger, 2018), which are explained below.

Mesh Non-Orthogonality The non-orthogonality of a mesh is determined by the angle between the normal vector of a face connecting two cells (at the face centre) and the vector between the two cell centres, shown in Figure 3. For a boundary face (i.e. a face that does not connect two cells, but connects a cell with the outisde of the model), the non-orthogonality is determined by the angle between the line connecting the cell centre and the face centre and the normal vector of the face at the face centre. The higher this angle, the more non-orthogonal the mesh is said to be at this location. A mesh with a lot of non-orthogonal faces results in an increase of numerical diffusion in the solution, mainly of the diffusive terms of the solved equations.

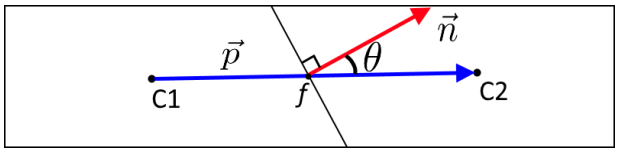


Figure 3: *Illustration of non-orthogonality, where $\vec{p}$ is the vector between the two cell centres $C1$ and $C2$, $\vec{n}$ is the normal vector of the face connecting the two cells at the face centre f, and θ, the angle between these two vectorrs, is the degree of non-orthogonality.*

Mesh Skewness The mesh skewness is defined as the deviation of the vector connecting two cell centres from the centre of the face connecting these two cells, divided by the distance between the two cell centres, shown in Figure 4. As well as adding numerical diffusion to both the convective and diffusive terms of the solved equations, cells with a high skewness also affect the interpolation of cell values to face centres.

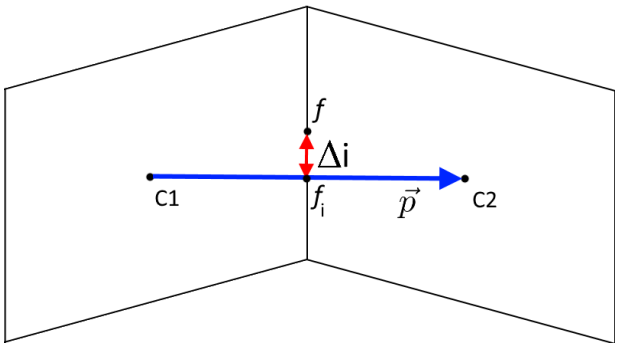


Figure 4: *Illustration of skewness, where $\vec{p}$ is the vector connecting the two cell centres $C1$ and $C2$, intersecting the face with face centre f connecting the two cells at point f_i, and Δi is the distance between points f and f_i. The skewness is then defined as Δi divided by the magnitude of vector $\vec{p}$.*

Mesh Aspect Ratio The aspect ratio of a mesh is given by the ratio of the longest side of its cells over their shortest side. Thus for a rectangular cell with sides $\Delta x > \Delta y$ (shown in Figure 5), the aspect ratio is defined as $\frac{\Delta x}{\Delta y}$. High aspect ratios tend to smear gradients in the solution, however, if these gradients are small over the long direction of the cells, a high aspect ratio is not necessarily problematic.

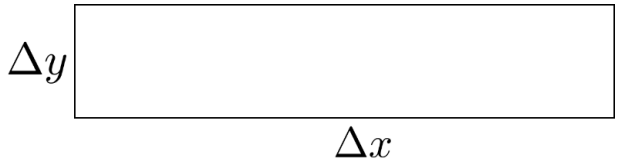


Figure 5: *Illustration of a cell with a relatively high aspect ratio* $\left(\frac{\Delta x}{\Delta y}\right)$.

Mesh Smoothness The mesh smoothness is determined by the transition in size between contiguous cells. For a mesh where two neighbouring cells have lengths of Δx_1 and Δx_2 (as shown in Figure 6), the smoothness is defined as the ratio $\frac{\Delta x_2}{\Delta x_1}$. A smooth mesh has a very gradual transition in cell sizes, whereas a mesh with a lower smoothness can contain sudden jumps in cell size from one cell to another, neighbouring cell. These sudden jumps in cell size tend to add numerical diffusion to the solution.

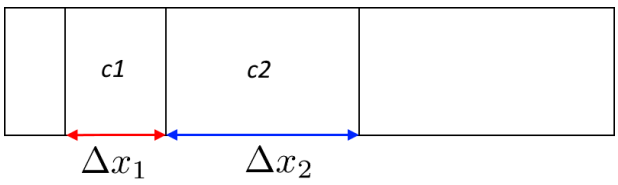


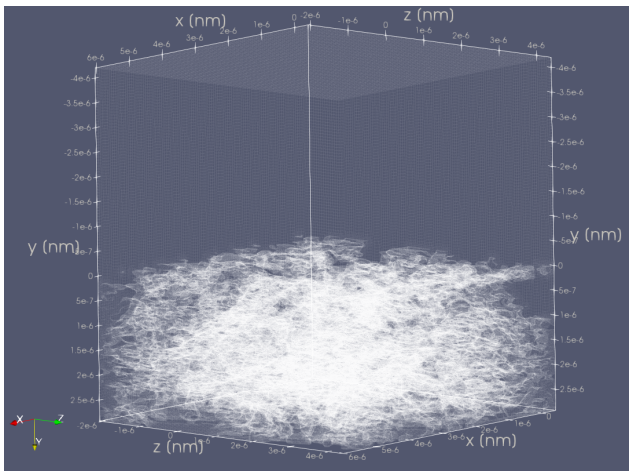
Figure 6: *Illustration of a mesh with a smoothness of* $\frac{\Delta x_2}{\Delta x_1}$ *between cells* $c1$ *and* $c2$.

The *checkMesh* utility provided by OpenFOAM can give a reasonably detailed account of many mesh quality metrics, as well as show other mesh information, such as the number of cells and the number of total and internal faces of the mesh. However, in reality the above mentioned metrics can be very hard to control, since the quality controls of *snappyHexMesh* are not able to give full control of all of these metrics, and so one often has to resort to trial and error in order to get the mesh quality metrics to their desired values. Furthermore, especially for complex geometries like the one used in this thesis, some concessions have to be made to also enable the mesh to accurately represent the input geometry. In particular, due to limitations in computational time, it is not practical to use a mesh with a very high number of cells in a simulation. However, since the coating geometry is very complex and features many small pores, a fine line has to be found between having a mesh that is too coarse, causing it to misrepresent the geometry in a major way, and having a mesh that is too fine, resulting in excessive computational time, making it impractical to run a simulation. This is characterized by the Courant number, which in its one dimensional form is expressed as:

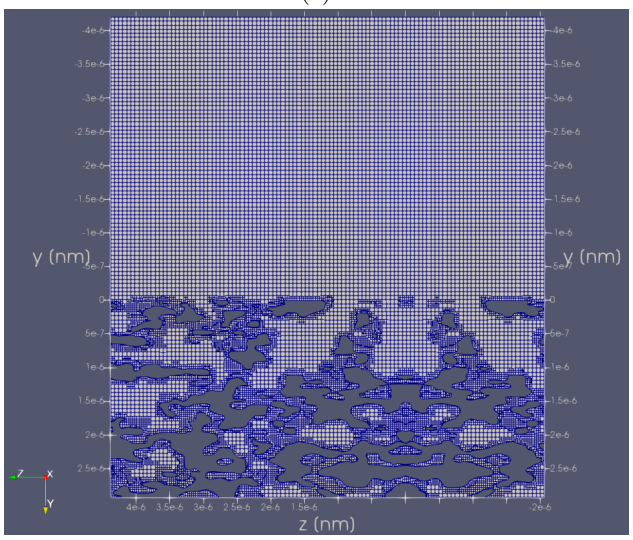
$$Co = \frac{u\Delta t}{\Delta x} \tag{88}$$

This number gives an indication of how well the simulation has converged. The upper part of the fraction represents the distance that a particle with velocity u would travel in one time step of length Δt in the simulation, while the lower part represents the size of the grid near the particle. If the Courant number becomes larger than one, this means that the particle is skipping cells of the mesh with each time step, which generally leads to numerical diffusion. Consequently, if a mesh is very fine, simulations on this mesh generally also need a very small time step size in order to keep the Courant number below a value of 1. This, added to the fact that the more cells are in a mesh, the longer it takes to compute a single time

step, means that one has to be careful not to make their mesh too fine.



(a)



(b)

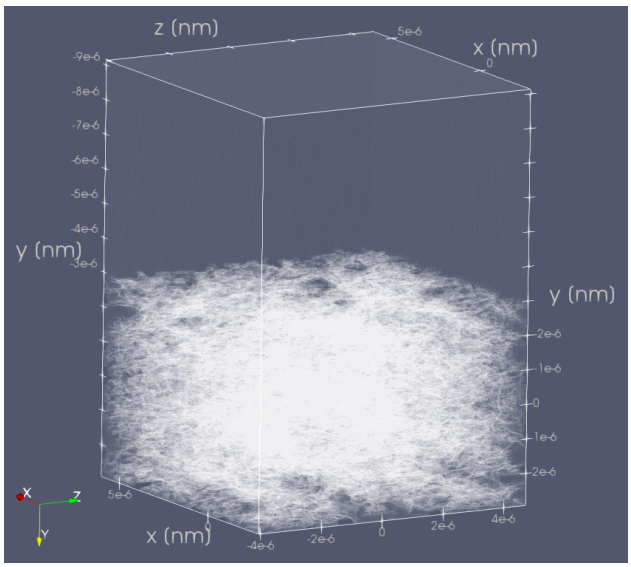
Figure 7: *Wireframe (a) and cross sectional (b) representation of the mesh based on the coating geometry with a depth of 2.9 [μm].*

3.2.2 Meshes Used in the Simulations

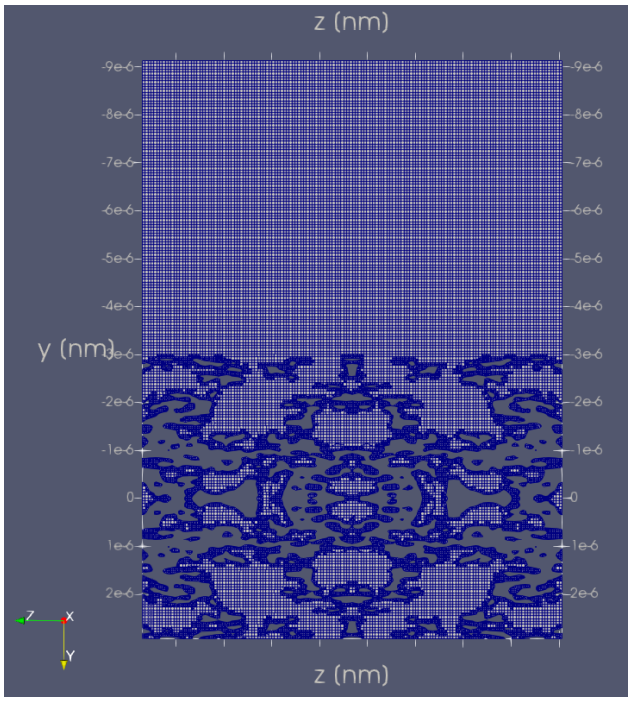
The meshes that were used in the simulations were created using the *blockMesh* and *snappyHexMesh* utilities with the aforementioned coating layer geometries as input. The settings for the *snappyHexMesh* utility are given in appendix A.

For the mesh based on the geometry with a depth of 2.9 [μm], which was prepared for the case of a droplet of 2 [μm] in radius, not the entire domain was used to generate the mesh. This was done since mirroring the original geometry resulted in a domain size which was way larger than the expected maximum droplet spreading for droplets of this size, and thus computational time could be saved by reducing the amount of cells in the mesh. The final size of this mesh was $6.44 \times 7.13 \times 6.44$ [μm^3], and it is shown in Figure 7. The mesh had a total of 2,747,477 cells and 8,742,037 faces. The maximum non-orthogonality of this mesh was 57.3, while the average non-orthogonality was 9.8. The maximum skewness of the mesh was 3.9, while the maximum aspect ratio was 8.2. The *checkMesh* output for this mesh is shown in appendix B.1.

The mesh based on the mirrored geometry with a coating layer depth of approximately 5.9 [μm], prepared for the simulation with a droplet radius of 3 [μm], had a total number of 9,510,158 cells and 30,308,873 faces. The final mesh size was $8.88 \times 12.05 \times 8.77$ [μm^3], and it is shown in Figure 8. The maximum non-orthogonality of the mesh was 58.38, and the average non-orthogonality was 10.12. The maximum skewness and aspect ratio of the mesh were 3.93 and 8.22, respectively. The *checkMesh* output for this mesh can be found in appendix B.2.



(a)



(b)

Figure 8: *Wireframe (a) and cross sectional (b) representation of the mesh based on the coating geometry with a depth of 5.9 [μm].*

3.3 Initial Conditions

To initialize the droplet in the domain, the volume fraction of liquid is set to 1 inside of a sphere with the desired center position and radius using the *setFields* utility of OpenFOAM. To save computational time, the spreading and penetration of the droplet was assumed to be symmetrical over the two axes parallel to the coating's surface. Therefore, only a quarter of the actual droplet was put into the corner of the mesh for each simulation. In all cases, the droplet was initialized such that it slightly touched the paper coating in the initial time step, in order for capillary effects to take over immediately. Figure 9 shows the initial position of the droplet for the case of a droplet with radius 2 [μm].

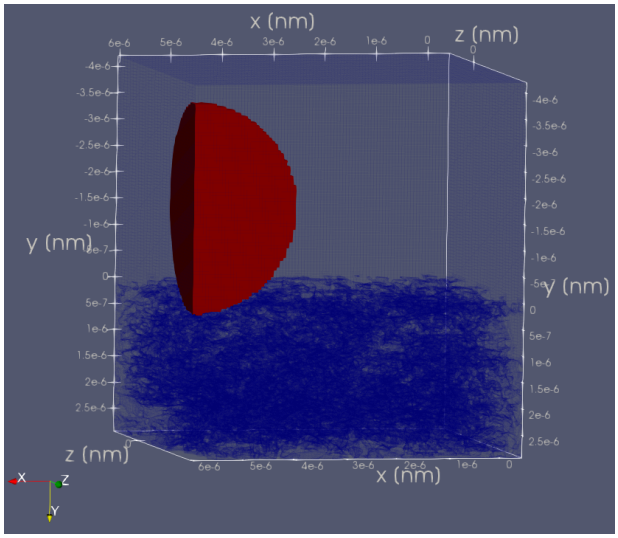


Figure 9: *Initial position of the droplet (red) for the case with a droplet radius of 2 [μm].*

The velocity values inside of the domain were set to zero, i.e.:

$$\vec{u_0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Similarly, the pressure values in the entire domain were set to zero, i.e.:

$$P - \rho g h = 0$$

3.4 Boundary Conditions

3.4.1 Atmospheric Boundaries

The lower boundary and the boundaries at the sides of the model were given an atmospheric boundary condition. For the pressure this entails a *totalPressure* boundary condition with an initial value set to 0. The *totalPressure* boundary condition sets determines the pressure as follows (Greenshields, 2018):

$$P = \begin{cases} P_0 & \text{for outflow} \\ P_0 - \frac{1}{2}|\vec{u}|^2 & \text{for inflow} \end{cases} \tag{89}$$

The value for P_0 was set to 0. The velocity boundary condition at these boundaries was set to the *pressureInletOutletVelocity*, with an initial value of $\vec{u} = [0\ 0\ 0]$. This boundary conditions assigns a *zeroGradient* boundary condition at all times, except for the tangential component, where for inflow a *fixedValue* boundary condition is assigned, which by default is zero. The volume fraction boundary condition for the atmospheric boundaries was set to *inletOutlet*. Similarly to the *pressureInletOutletVelocity* boundary for the velocity, this boundary condition assigns a *zeroGradient* for outflow, but switches to a *fixedValue* boundary condition when there is inflow at the boundary. Both the initial value and the inflow value for the volume fraction were set to 0.

3.4.2 No-Slip Boundaries

For the parts of the domain that needed to represent a solid surface, no slip boundary conditions were applied. For the pressure this meant assigning a *fixedFluxPressure* boundary condition, again with an initial value of 0. This boundary condition is similar to a *zeroGradient* boundary condition, but it adjusts the gradient to gravity and surface tension effects (Greenshields, 2018). For the velocity, the boundary condition at the no-slip boundaries was simply set to a *fixedValue* boundary condition with value $\vec{u} = [0\ 0\ 0]$. On the grains of the coating, the volume fraction boundary condition was set to *constantAlphaContactAngle*, which assigns an equilibrium contact angle restraint on the volume fraction near the surface. Apart from the grains, the upper boundary of the domain was also made solid, to prevent the droplet from escaping the domain through the top. For this upper boundary the same pressure and velocity boundary conditions were used, but for the volume fraction the *zeroGradient* was used.

3.4.3 Symmetry Plane Boundaries

As mentioned above, only a quarter of the actual droplets were simulated, under the assumption that the behaviour of the droplets is symmetrical over the two axes that run parallel to the surface of the coating. To do this, two of the sides of the domain were given the *symmetryPlane* boundary condition for the pressure, velocity and the volume fraction. This boundary condition assumes that the domain is mirrored over that particular boundary and assigns its values accordingly.

The OpenFOAM files specifying the pressure, velocity and volume fractions can be found in appendices C.1, C.2 and C.3, respectively.

4 Results & Discussion

4.1 Sensitivity Analysis

There are many parameters that determine the precise way the *interFoam* find its solution for each time step. The discretisation schemes are defined in the *fvSchemes* dictionary file, while the equation solvers and algorithms are specified in the *fvSolution* dictionary file. Preliminary to running the main simulations, a sensitivity analysis was performed on the settings in the *fvSolution*, to determine the effect of different parameters on the simulation results and the required computational time.

4.1.1 Simulation Setup

The simulations were done on two meshes. The first set of simulations were performed on a relatively coarse mesh without surface refinement. In later runs this mesh was replaced with a more detailed mesh including surface refinement. Both meshes were the same size: $1.25 \times 1.1 \times 1$ [μm^3]. Note that this is much smaller than the meshes that were used in the actual simulations, discussed in section 3.2.2. The former mesh contained 299,758 cells with 917,872 faces, and had a maximum and average non-orthogonality of 65 and 5.7, respectively, while the latter mesh contained 669,058 cells with 2,160,019 faces, with a maximum and average non-orthogonality of 55 and 10.8, respectively. Both meshes are shown in Figures 10 & 11.

For the fluid properties of the droplet, the properties of water were used, as described above in section 3.1. An equilibrium contact angle of 45 degrees was assigned. The radius of the droplet in these simulations was set to 0.4 [μm].
The initial position of the water droplet in the simulations was partially embedded into the coating to speed up the simulation a bit. This means that the results themselves are not very representative of the actual process of droplet imbibition into coating since this initial penetration likely has a large influence on the results, but for the purpose of this sensitivity analysis this was not deemed to be a problem since it was the difference between the simulations that was of interest, not the absolute results.

All of the simulations on the coating were run up until a time of $2.5 \cdot 10^{-6}$ seconds, at which point the droplet had infiltrated completely and become approximately static, although it should be noted that due to the droplet's proximity to the model boundaries, there was some mass transfer of the droplet through the boundary.

Initially, the following *fvSolution* settings were used:

```
nAlphaCorr:        2
nAlphaSubCycles:   1
cAlpha:            1
MULESCorr:         yes
nLimiterIter:      3

Alpha solver:      smoothSolver
Alpha smoother:    symGaussSeidel
Alpha tolerance:   1e-8
Alpha relTol:      0

Pressure solver:      GAMG
Pressure smoother:    GaussSeidel
Pressure tolerance:   1e-6
Pressure relTol:      0

Velocity solver:      smoothSolver
Velocity smoother:    symGaussSeidel
Velocity tolerance:   1e-6
Velocity relTol:      0

momentumPredictor:        yes
nOuterCorrectors:         1
nCorrectors:              3
nNonOrthogonalCorrectors 2
```

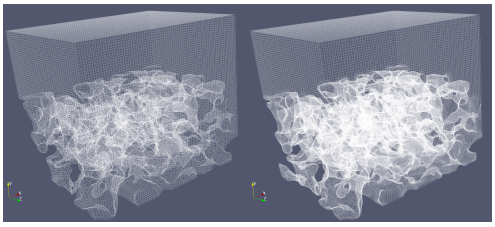


Figure 10: *Wireframe representation of the unrefined mesh (left) and the refined mesh (right) used in the simulations.*

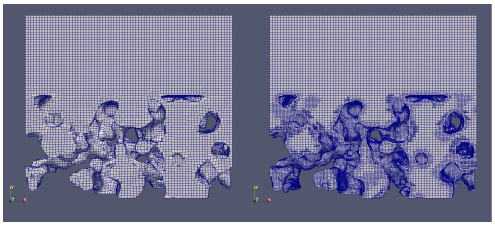


Figure 11: *Slices showing the internal structure of the unrefined mesh (left) and the refined mesh (right) used in the simulations.*

4.1.2 Sensitivity Analysis on Unrefined Mesh

The following simulations were performed on the mesh without surface refinement (left side of Figures 10 & 11).

Pressure solvers (GAMG & PCG with DILU preconditioner) There was no noticable difference in droplet spreading when using the GAMG (Geometric Agglomerated Algebraic Multigrid) solver with the GaussSeidel smoother compared to the same simulation but with the PCG (Preconditioned Conjugate Gradient) solver with the DIC (Simplified Diagonal-based Incomplete Cholesky) preconditioner for the pressure equation. The initial residuals for p_rgh for both simulations were also very similar and satisfactory (of the order of 10^{-5}). The initial residuals for the velocity were however quite high for both the pressure solvers. There was a significant difference in computational time between the two solvers, with the PCG solver needing approximately 12500 seconds to complete $2 \cdot 10^{-6}$ of simulation time, while the GAMG needed approximately 59500 seconds to complete the same amount of simulation time. Since the solutions for both solvers were very similar, and the PCG solver was much less computationally demanding, it was decided to use this pressure solver for most of the following simulations.

Velocity Boundary Conditions (pressureInletOutletVelocity & zeroGradient) The droplet spreading in the simulations using the pressureInletOutletVelocity and the zeroGradient boundary conditions for the velocity very similar. The pressureInletOutletVelocity boundary condition resulted in slightly better initial residuals for p_rgh, but the initial resiuald for the velocity were slightly better in the simulations with the zeroGradient boundary condition. In terms of computational time, the pressureInletOutlerVelocity boundary seemed to be slightly less demanding than the zeroGradient boundary condition (12500 seconds versus 13500 seconds to compute $2 \cdot 10^{-6}$ seconds of simulation, respectively). This is why it was decided to keep the pressureInletOutletVelocity boundary condition in later simulations.

Momentum Predictor Two simulations were run with the PCG solver for the pressure and the pressureInletOutletVelocity boundary condition, but in one of the two the momentum predictor was turned off (it was on in preceding simulations). Both simulations had very similar spreading of the droplet. The initial residuals for p_rgh were slightly better with the momentum predictor turned off. Since the residuals for the velocity are not logged when the momentum predictor is turned off, these could not be compared. With the momentum predictor turned

on there were some strange velocity oscillations in the top part of the model, which were not seen when the momentum predictor was turned off. The solution for the simulation with the momentum predictor turned on also featured some small areas in the model with very high pressures, which were not as clearly present when it was turned off. The computational time for the simulation with the momentum predictor turned off was much shorter (approximately 5300 seconds for a simulation time of $2 \cdot 10^{-6}$) than when it was turned on (approximately 12500 seconds for the same simulation time). Due to the slightly better pressure residuals, the strange velocity oscillations found with the momentum predictor on, and the much shorter computational time with the momentum predictor turned off, it was decided that it was likely best to turn the momentum predictor off in actual simulations.

However, to see if turning the momentum predictor off had a similar effect for the GAMG solver for the pressure, a simulation was also done with this solver but with the momentum predictor off. Compared to the GAMG simulation with the momentum predictor on, there were some minor differences in the spreading of the droplet. The initial residuals for p_rgh were much better when the momentum predictor was turned off, and this smulation was also much less computationally demanding, but still more so than the simulations with the PCG solver.

Velocity Solvers (smoothSolver & PBiCG) Here, two different solvers for the velocity equation were compared, the smoothSolver (which was also used in prior simulations) and the PBiCG (Preconditioned Bi-Conjugate Gradient) solver with DILU (Simplified Diagonal-based Incomplete LU) smoother. For these simulations the momentum predictor was also still turned on. Again the spreading of the droplet in the coating was very similar with both solvers. Initial residuals for p_rgh were slightly better when using the PBiCG solver, while the initial residuals for the velocity were very similar for both simulations. There were no large differences in the pressure fields for both simulations. The velocity fields for both simulations were very similar as well, however, a single written timestep in the simulation using the PBiCG solver had much higher velocities throughout the entire domain of the model. Since this was similar to the oscilaltions found when comparing the momentum predictor turned on and off, the same simulation was run again but with the momentum predictor off, and for this simulation this timestep with high velocities did not occur, so it was likely that this was related to the momentum predictor and not the PBiCG solver. The computational time for the simulation using the PBiCG solver was shorter than that of the simulation using the smoothSolver (approx. 10800 seconds and approx. 12500 seconds for $2 \cdot 10^{-6}$ seconds of simulation, respectively, with the momentum predictor turned on). It was therefore decided to use the PBiCG solver from now on.

MULES Corrector Two simulations (PCG pressure solver, PBiCG velocity solver, momentum predictor turned off, pressureInletOutletVelocity boundary condition for the velocity), one with the MULES corrector turned on (as was the case for preceding simulations), and one with the MULES corrector turned off, did not result in a significant difference in droplet spreading. With the MULES corrector turned off, the initial residuals for p_rgh were slightly better than when it was turned on. Velocities seemed to be slightly higher with the MULES corrector turned off, but the pressures were found to be very similar. The simulation with the MULES corrector turned off ran slower than when it was turned on (approx. 7500 seconds versus approx. 5100 seconds, respectively, for $2 \cdot 10^{-6}$ seconds of simulation time). Since the spreading was so similar, and the simulation with the MULES corrector turned on was much faster, it was decided that the slight decrease in initial residuals with the MULES corrector turned off was probably not worth the extra required computational time.

nCorrectors Four different values for nCorrectors were compared: 1, 2, 3 & 4. The droplet spreading for the four simulations were very similar, however, the droplet occupied a certain pore in the simulations with nCorrectors 1 & 4, but not for the simulations with nCorrectors 2 & 3. The initial residuals for p_rgh generally improved as the value for nCorrectors increased, although the residuals for nCorrectors 3 & 4 were very similar. There were no large differences in velocities for the four simulations, but the pressures differed somewhat, with nCorrectors 1 & 4 resulting in similar pressures and nCorrectors 2 & 3 having similar pressures as well. This is likely related to the fact that the droplet distribution was slightly different for the simulations with nCorrectors 1 & 4 versus the simulations with nCorrectors 2 & 3. As was expected, the computational demand increased as the value for nCorrecotrs increased, with a simulation of $2 \cdot 10^{-6}$ taking approx. 3000 seconds for nCorrectors 1, approx. 4360 seconds for nCorrectors 2, approx. 5140 seconds for nCorrectors 3, and approx. 6500 seconds for nCorrectors 4. Based on this

information it was decided that it was likely safe to use a value of 2 to save some computational time, as the results for nCorrectors 2 & 3 were so similar.

nOuterCorrectors Four simulations were run, using nOuterCorrector values of 1, 2, 3 & 4. The spreading of the droplet was similar for all four of the simulations, and although there were some differences in initial residuals for p_rgh, none of the simulations had consistently better residuals than the others. Therefore it was decided to simply use an nOuterCorrectors value of 1 to save computational time.

Alpha Solvers (smoothSolver & PBiCG) Two simulations were done using different solvers for the alpha equation, namely the smoothSolver and the PBiCG solver. Note that for this run the nCorrectors parameter was still set to a value of 3. There was very little difference in spreading of the droplets between the two simulations. Initial residuals for alpha and p_rgh were slightly better when using the PBiCG solver. Both the velocity and the pressure distributions were almost the same for the two simulations. In terms of computational time, the smoothSolver was slightly faster than the PBiCG solver, with a simulation of $2 \cdot 10^{-6}$ seconds taking approximately 5140 and 5330 seconds, respectively. It was decided that the better residuals when using the PBiCG solver were probably worth the slight increase in computational time.

4.1.3 Sensitivity Analysis on Refined Mesh

The following simulations were performed on the refined mesh, shown on the right in Figures 10 & 11. The initial settings that were used for this mesh were largely based on the findings for the unrefined mesh, but some parameters were retested for this mesh to see if the quality of the mesh had an influence on the sensitivity of the model to these parameters.

Pressure Solvers (GAMG & PCG) The GAMG and PCG pressure solvers were retested with the new mesh. The spreading of the droplet did not differ much between the two simulations. The initial residual for p_rgh were found to be slightly better for the simulation using the GAMG solver. Both the elocity and pressure distributions were very much alike for the two simulations. Similar to what was found for the unrefined mesh, the PCG solver ran much faster than the GAMG solver. Therefore, the PCG solver was chosen as the pressure solver for the following simulations.

nCorrectors The nCorrectors parameter was also retested on the refined mesh, but this time only with values 2 & 3. Both droplet spreading and velocity and pressure distributions were found to be very similar for the two simulations. The initial residuals for p_rgh were significantly better when using nCorrectors 3 as opposed to 2, but this came at the price of a much longer computational time (a simulation of $2.5 \cdot 10^{-6}$ seconds taking approx. 17000 seconds for nCorrectors 3 and approx. 12000 seconds for nCorrectors 2). In the end, nCorrectors 2 was chosen in order to save as much time as possible.

cAlpha To see the effect of cAlpha, which controls the amount of interface compression, on simulation results, four simulations with different values for cAlpha were done. The valus that were chosen were 0, 0.1, 0.8 and 1. There was a significance difference in droplet spreading when comparing the lower values (0 & 0.1) to the higher values (0.8 & 1). For lower values of cAlpha, the initial residuals for p_rgh were generally better than for higher values. However, the interface was found to be much more diffuse when using a lower value for cAlpha. After $2 \cdot 10^{-6}$ seconds of simulation, the interface was only about 2-3 cells wide for the simulations using a cAlpha of 1 and 0.8, while for a cAlpha of 0.1 it was approx. 16 cells wide, and for a cAlpha of 0 it was approx. 20 cells wide. Computational time increased with increasing cAlpha values, with the simulation using a cAlpha value of 0 taking approx. 4300 seconds for a simulation time of $2.5 \cdot 10^{-6}$ seconds, while the simulation using a cAlpha value of 1 took approx. 11700 seconds for the same amount of simulation time. Based on this, a value for cAlpha of 0.8 was chosen as a balance between computational time, interface diffusiveness and initial residuals.

MULES Corrector The simulations with the MULES corrector turned on and off were repeated for the refined mesh. Some slight differences in droplet spreading were noted, although there was no significant difference in the velocity and pressure distributions for the two simulations. Although the initial residuals for p_rgh were of a similar magnitude for the two simulations, the residuals for the simulation with the MULES corrector turned off had a larger spread than that for the simulation with the corrector turned on. The simulation with the MULES corrector turned off took slightly longer to run (approx. 12800 seconds to simulate $2.5 \cdot 10^{-6}$ seconds versus approx. 11700 seconds with the MULES corrector turned off). Thus, similar to what was decided for the unrefined mesh, it was chosen to keep the MULES

corrector on in further simulations.

nAlphaSubCycles Two simulations were done with different values for the nAlphaSubCycles parameter: 1 & 5. Between the two simulations there was no difference in droplet spreading, and the initial residuals for p_rgh were also of a similar magnitude. Furthermore, the velocity and pressure distributions were also not found to change much for the two simulations, and the diffusiveness of the interface was not significantly reduced when using a value of 5 for nAlphaSubCycles. The computational time was much longer for nAlphaSubCycles = 5 as opposed to 1, taking approx. 24000 and 10150 seconds, respectively, to simulate $2.5 \cdot 10^{-6}$ seconds. Therefore, it was decided to keep the value of nAlphaSubCycles at 1.

nAlphaCorrectors Three different values for nAlphaCorrectors were tested: 1, 2 & 3. All three of these simulations had similar spreading of the droplet. As nAlphaCorrectors gets higher, the initial residuals for p_rgh seemed to improve as well. The pressure distributions for the three simulations were similar, as were the velocity distributions, although for an nAlphaCorrectors value of 1 the velocities on average seemed to be somewhat higher than for nAlphaCorrectors values of 2 & 3. For $2.5 \cdot 10^{-6}$ of simulation time, the computation times were approx. 11200 seconds for an nAlphaCorrectors value of 1, 10150 seconds for an nAlphaCorrectors value of 2, and 14800 seconds for a value of 3. It's unclear why the computation time is less for nAlphaCorrectors 2 than for nAlphaCorrectors 1, although it might have something to do with the slightly higher velocities found for nAlphaCorrectors 1. In the end, it was decided to keep the value for nAlphaCorrectors at 2.

nAlphaSubCycles & Courant Number A simulation was performed with nAlphaSubCycles set to 5 in combination with a maximum Courant number of 5. This simulation resulted in very high initial residuals for p_rgh, increasing both in variance and in absolute value towards the end of the simulation. The computational time for this run was slightly shorter than that for the simulation with nAlphaSubCycles set to 1 and a maximum Courant number of 1, taking approximately 7600 seconds to simulate $2.5 \cdot 10^{-6}$ seconds, as opposed to 10200 seconds. Due to the high initial residuals it was decided to keep nAlphaSubCycles and the maximum Courant number both at a value of 1.

nNonOrthogonalCorrectors Three simulations were done using different values for the nNonOrthogonalCorrectors parameter: 1, 2 & 3. The initial residuals for p_rgh were found to improve with higher values for nNonOrthogonalCorrectors. The difference in computational time was found to be minimal between nNonOrthogonalCorrectors 1 & 2, while for nNonOrthogonalCorrectors 3 the computational time was significantly longer.

4.1.4 Resulting *fvSolution* Parameters

The updated *fvSolution* settings that were decided based on this sensitivity analysis, together with the fvSchemes file, can be found in appendices C.4 and C.5, respectively.

4.2 Results for Different Contact Angles

For the simulations of droplet spreading and penetration under different equilibrium contact angles, contact angles of 1°, 45° and 90° were used. The fluid properties of the droplet were that of water, as described in section 3.1. After a simulation time of $1.5 \cdot 10^{-5}$ seconds, the droplet was approximately static in all three the simulations. The results of the simulations for a number of time steps are shown below in Figures 12 and 13. The extents of the quarter droplets with different contact angles through time are shown in Figure 14. Figure 15 shows the total volume penetrated into the paper coating layer through time for the three different contact angles. Note that since the values of the extents and infiltration are for the quarter droplets that were simulated, the values of the x- and z-axis extent and the infiltration for a full droplet would be approximately 4 times larger.

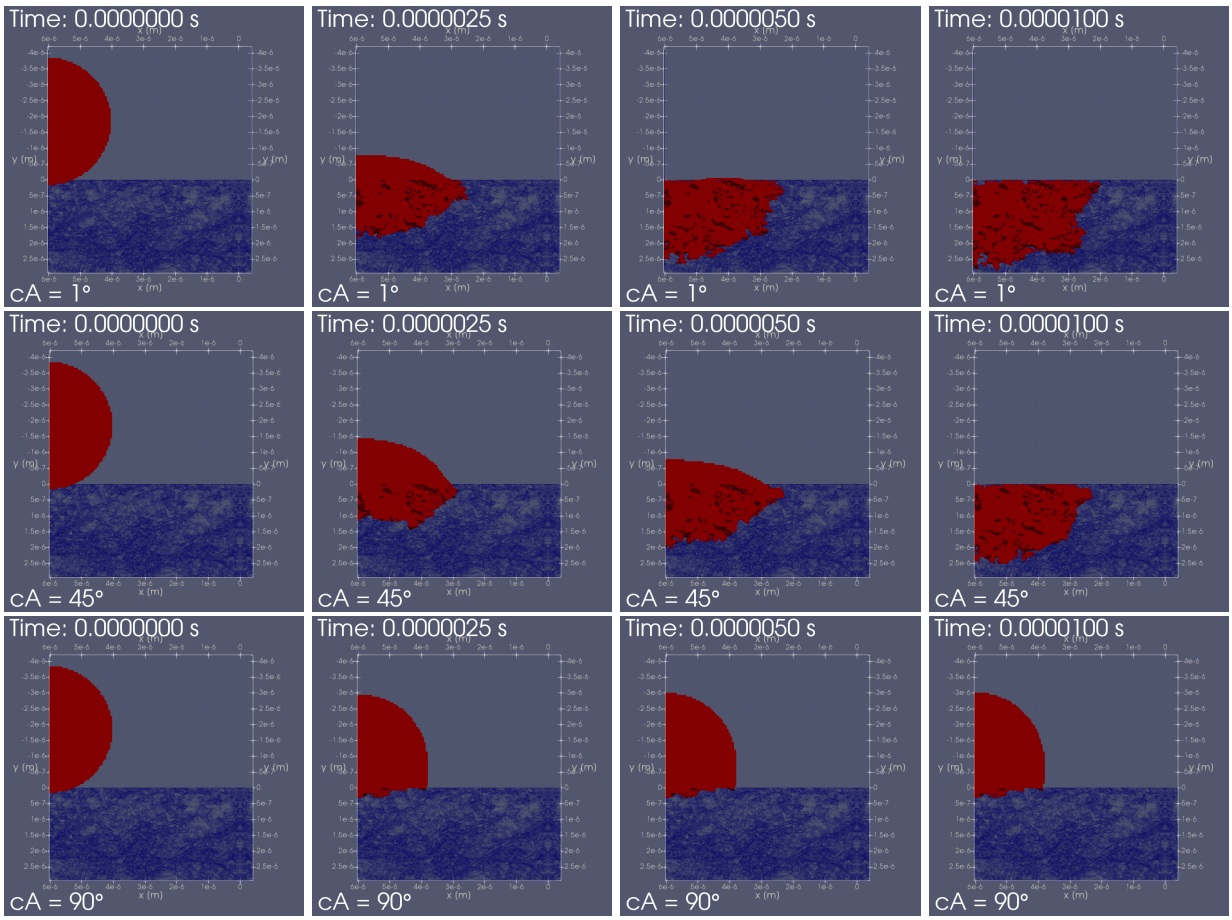


Figure 12: *Selected time steps from the results of the simulations with* $\theta_{eq} = 1°$, $\theta_{eq} = 45°$ *and* $\theta_{eq} = 90°$ *as seen from the side.*

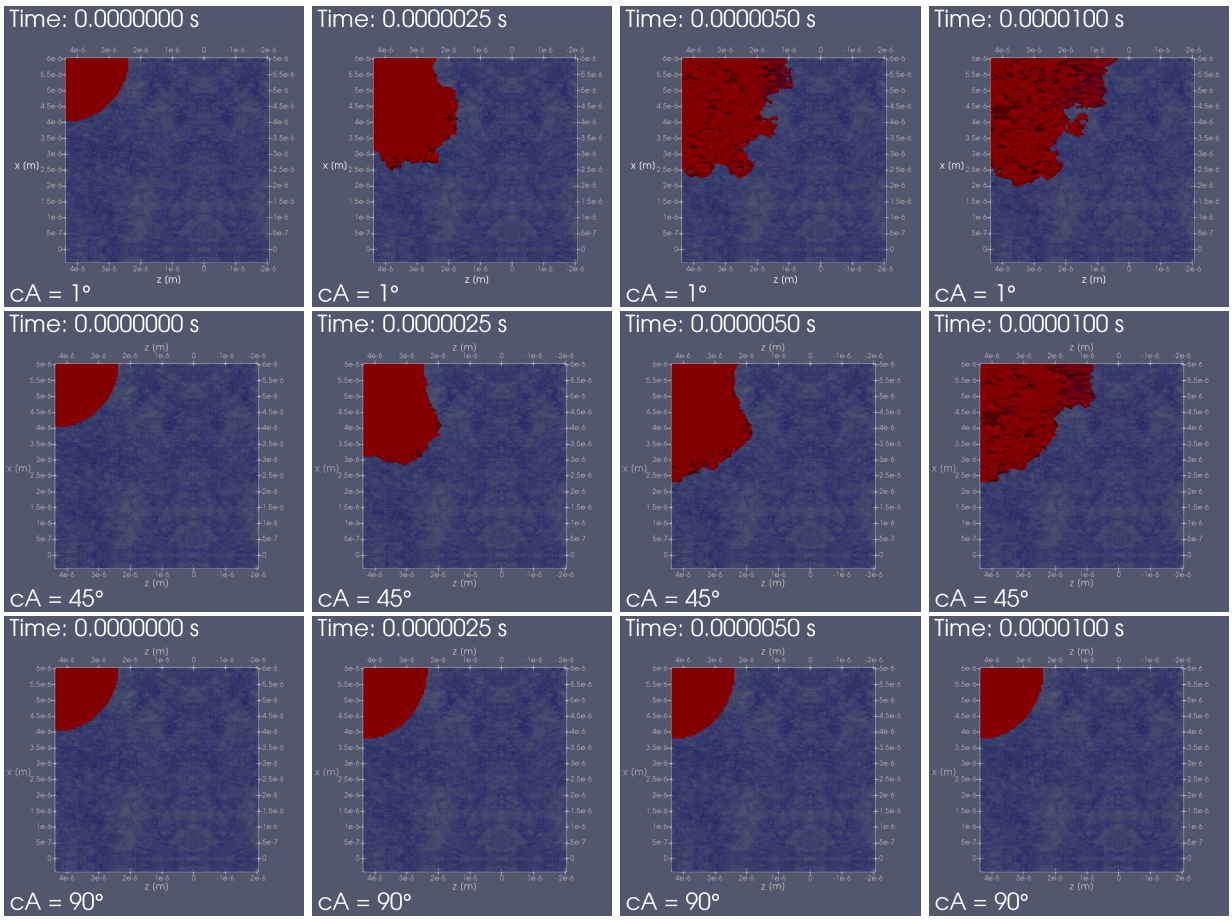


Figure 13: *Selected time steps from the results of the simulations with* $\theta_{eq} = 1°$, $\theta_{eq} = 45°$ *and* $\theta_{eq} = 90°$ *as seen from the top.*

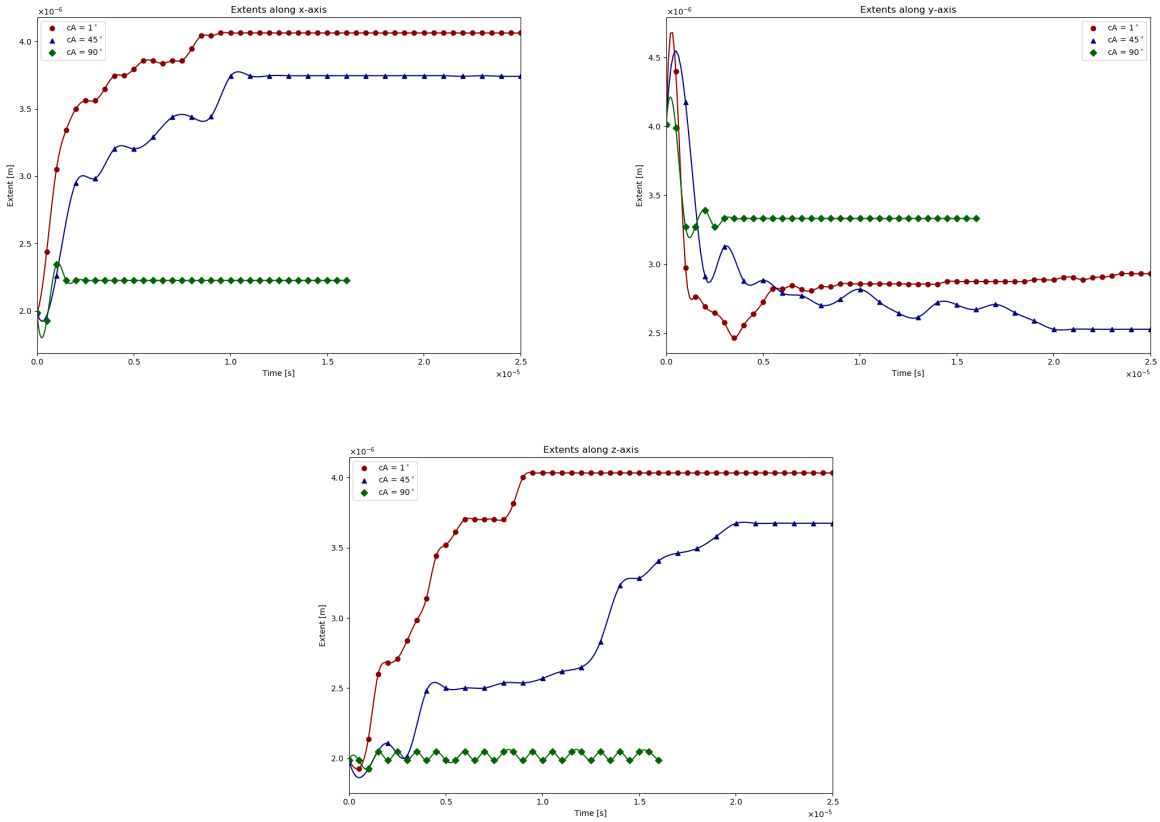


Figure 14: *Quarter droplet extents along the x-, y- and z-axis for the droplets with contact angles of 1°, 45° and 90°.*

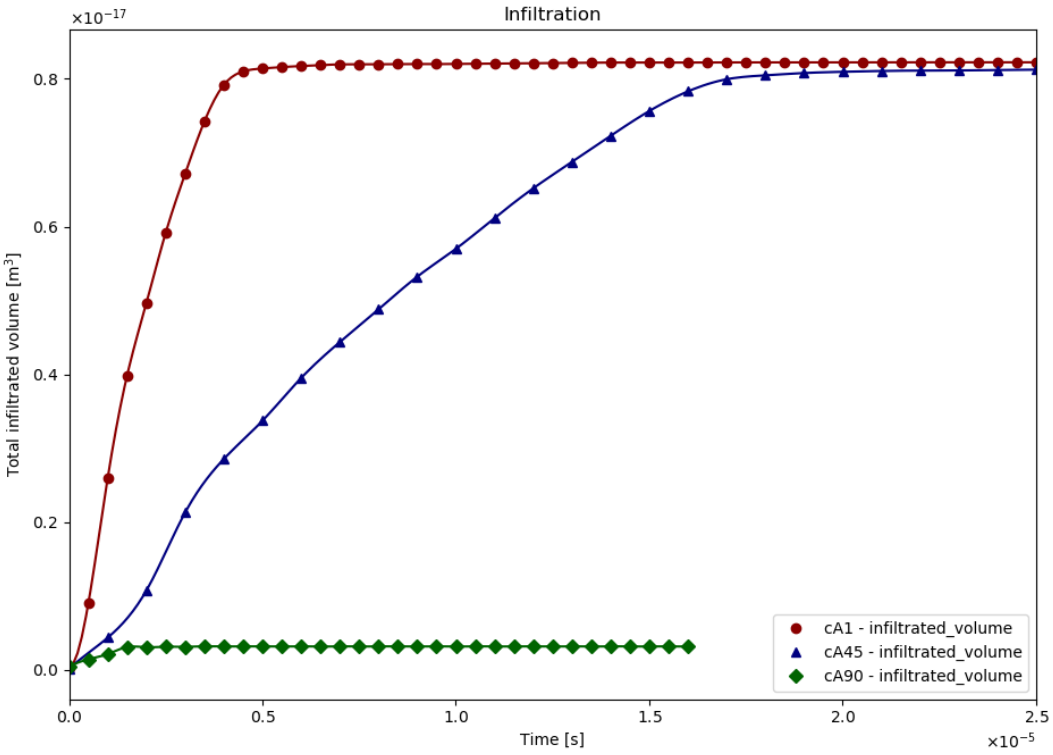


Figure 15: *Infiltrated volume through time for the quarter droplets with contact angles of 1°, 45° and 90°.*

4.3 Results for Droplet with a Radius of 3 micron

The results for the simulations with a droplet with a radius of 3 [μm] are shown in Figure 16. This droplet also had the fluid properties of water as described in section 3.1, and the droplet became static after approximately $5 \cdot 10^{-5}$ seconds. The droplet extents and the total infiltrated volume for the droplet with a radius of 3 [μm] are shown in Figure 17. As with the droplets with a radius of 2 [μm], these results are for a quarter of an actual droplet. To get the results for a full droplet, the extents along the x- and z-axis should be doubled, and the infiltrated volume should be multiplied by 4.

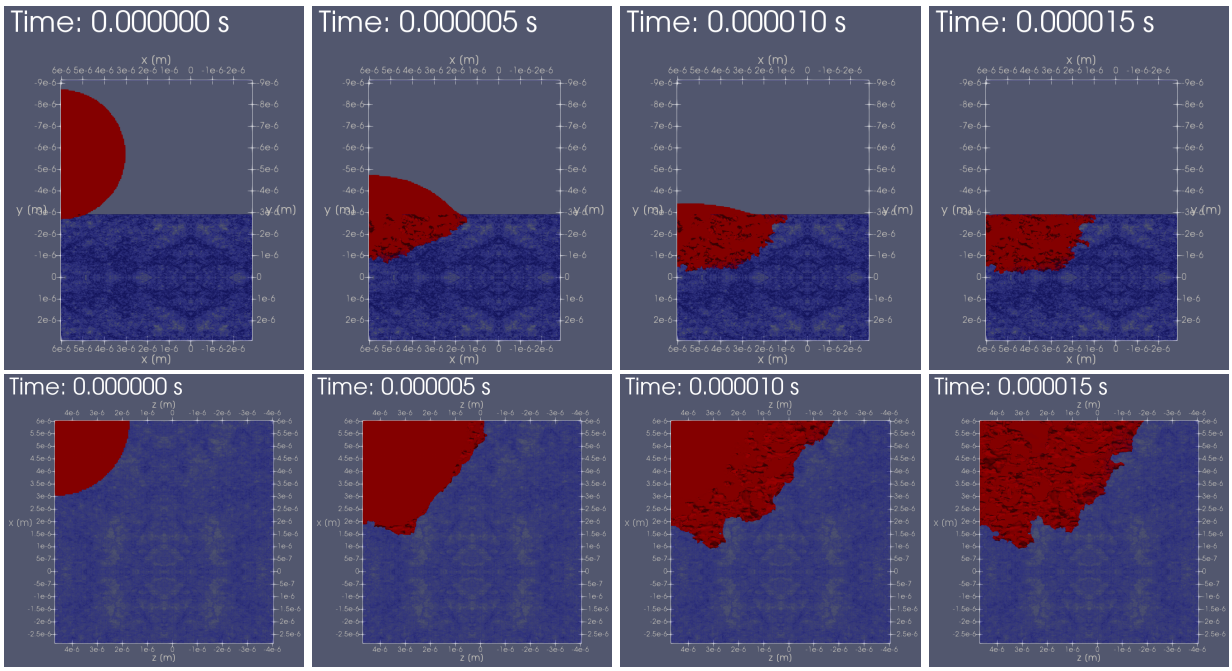
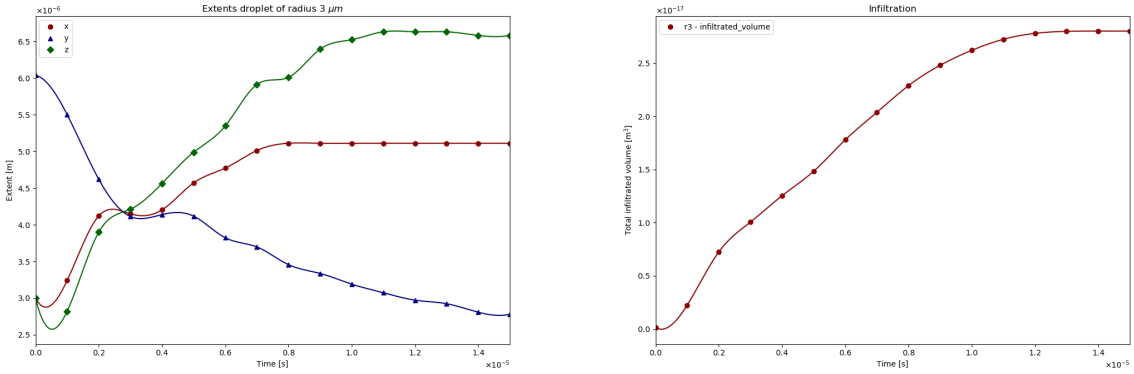


Figure 16: *Selected time steps from the simulation for a droplet with a radius of 3 [µm] as seen from the side and from the top.*



(a) (b)

Figure 17: *Extents along the x-, y- and z-axis (Figure 17a), and total infiltrated volume (Figure 17b) for a quarter of the droplet with radius 3 [µm].*

4.4 Results for Ink-like Droplet

Results for the simulation with an ink-like droplet (radius 2 [μm]) are shown in Figure 18. For these simulations the properties of the ink-like liquid described in section 3.1 were used. Droplet extents and total infiltrated volume for the ink-like droplet are shown in Figure 19.

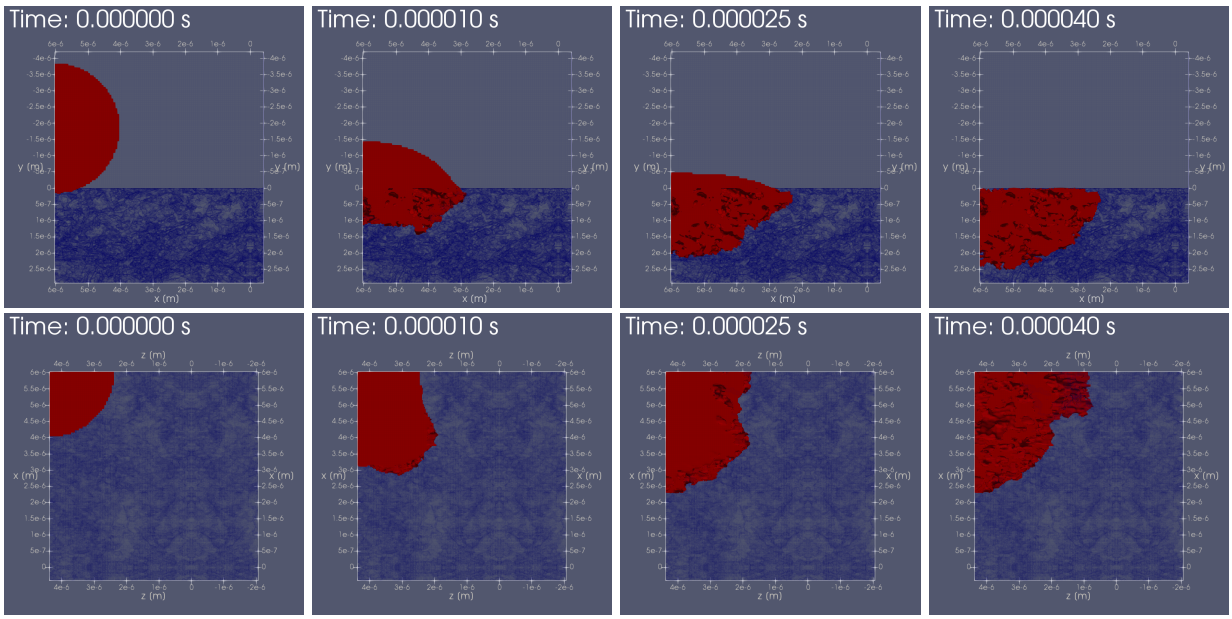
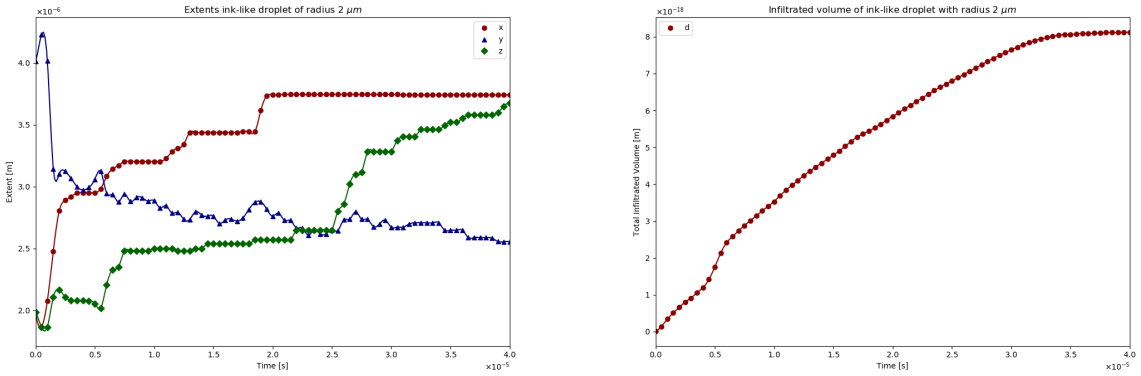


Figure 18: *Selected time steps from the simulation for an ink-like droplet with a radius of 2 [µm] as seen from the side and from the top.*

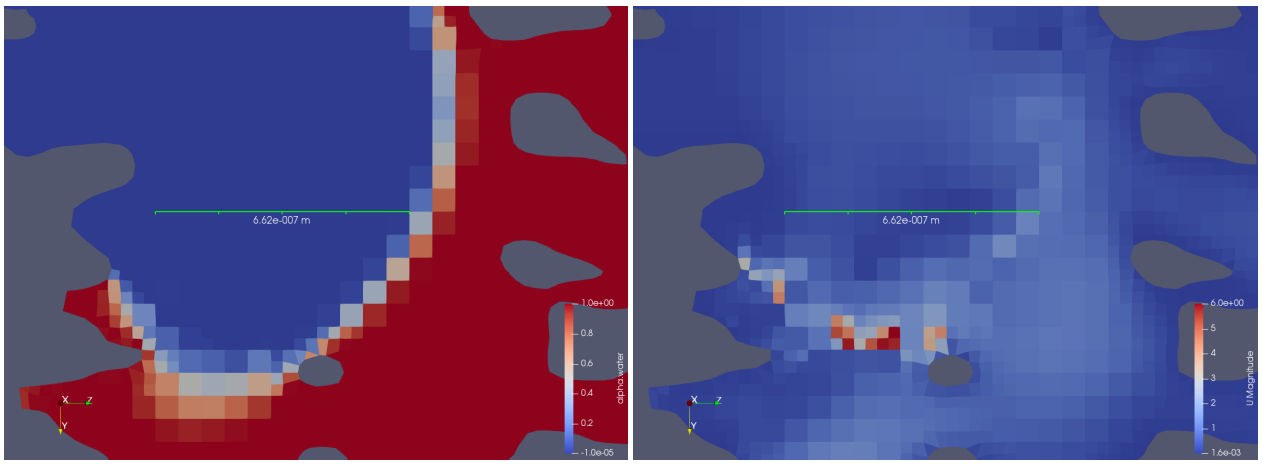


(a) (b)

Figure 19: *Extents along the x-, y- and z-axis (Figure 19a), and total infiltrated volume (Figure 19b) for a quarter of the ink-like droplet with radius 2 [µm].*

4.5 Occurrence of Parasitic Currents

One notable quirk of all the simulation results is the occurrence of so called parasitic currents at the interface, which are illustrated in Figure 20. Here, very high flow velocities are found (up to 6 [m s^{-1}]). The currents seem to be coming from the interface between the two fluids, from where they permeate through the rest of the model. These parasitic currents are the result of the implementation of the Continuum Surface Force Method, explained in section 2.5.2, in the Volume of Fluid method in cases where the flow is dominated by surface tension forces (Harvie et al., 2006).



(a) (b)

Figure 20: *Illustration of parasitic currents in a cross-sectional view of a randomly picked time step of one of the simulations. Figure 20a shows the volume fraction of the cross-section at this point in time, while Figure 20b shows the velocity magnitude at the same time.*

4.6 Influence of Contact Angle on Droplet Spreading & Infiltration

From Figures 12, 13, 14 and 15, it can be seen that the equilibrium contact angle assigned to the water droplet has a significant impact on the movement of the droplet through time, as well as on the distribution of the liquid after the system has become approximately static. Although the final distribution of water in the paper coating for equilibrium contact angles 1° and 45° are fairly similar, the figures clearly show that the liquid droplet with an equilibrium contact angle of 45° takes significantly longer to penetrate the paper coating than the liquid droplet with an equilibrium contact angle 1°. From Figure 14 it can also be seen that the droplet with a contact angle of 1° spreads slightly further along the paper surface than the droplet with a contact angle of contact angle of 45°. In the simulation with the droplet with equilibrium contact angle of 90° shows very different results from the other two simulations. The droplet remains fairly spherical, with very little infiltration occurring into the paper coating. In the application of inkjet printing, the latter result is most desirable, since the slight penetration into the coating would still allow for the dye to separate and adsorb onto the coating layer, while the spreading of the droplet is limited, thus making the coalescence of adjacent droplets much more unlikely. In reality, where evaporation also plays a role, the part of the droplet that stays on top of the coating layer would then have enough time to evaporate, leaving only the adsorbed dye attached to the paper coating. Between the equilibrium contact angles of 1° and 45°, where the droplet eventually fully penetrates, the case of a droplet with contact angle of 45° would seem to be preferable from these results. Firstly the droplet spreads less far along the x- and z-axis, and furthermore the droplet takes longer to penetrate into the coating (Figure 15), which would give evaporation more time to act on the droplet, leading to less volume entering the coating. This would be beneficial in the application of inkjet printing, since this either limits the radial wicking of the droplet, or the vertical penetration of the droplet, or both. Radial wicking should be minimised in order to reduce the likelihood of adjacent droplets touching, which would reduce print quality, while vertical penetration should be limited in order to reduce the likelihood that the liquid reaches the underlying fibrous layer and consequently causes swelling and deformation of the layer. Figure 13 also shows that the radial spreading of the liquid in the coating layer becomes more heterogeneous as the equilibrium contact angle gets lower. This is another indication that a higher equilibrium contact angle is preferable for inkjet printing applications on coated papers, as a more homogeneous spreading means less "mottle" in the printing results (Aslannejad et al., 2019a).

The simulation results are in line with the hypothesis that a lower equilibrium contact angle would result in both a faster penetration into the paper coating and more spreading of the droplet on the surface. Indeed, both a faster penetration into the paper coating, and more spreading of the droplet on the surface of the paper coating is seen for lower equilibrium contact angles.

4.7 Comparison Water & Ink-like Droplets

Comparing the simulations for a droplet with properties of water and with properties of ink (both with an equilibrium contact angle of 45°) shows that as the droplets becomes static, their extents (Figures 14 & 19a) and total infiltrated volume (Figures 15 & 19b) are very similar. However, the time over which the droplet is still in motion is much longer for the case of the ink-like droplet, which takes more than twice as long to fully penetrate the paper coating. This was expected, since the higher viscosity of the ink-like droplet results in a greater resistance to flow. The evolution through time is however very similar, with both droplets showing a jump in extent along the z-axis about halfway through the spreading process. Since the ink-like droplet takes longer to infiltrate, if evaporation would be taken into account, it would have more time to evaporate, which would limit the infiltration depth and spreading. Thus, in the process of inkjet printing, droplets with ink-like properties (a higher viscosity and density) would be more preferable than water-like droplets, since the likelihood of droplets coalescing would be reduced.

4.8 Comparison Droplet with 3 [μm] Radius & Experiments

The extent of the water droplet with a radius of 3 [μm] and an equilibrium contact angle of 45° after spreading was found to be around 6.5 [μm] along the z-axis and around 5 [μm] along the x-axis. These results are very similar to those found experimentally for small stains (droplets of a size less than 20 [μm]) observed by Aslannejad et al., 2019b. Here, the maximum spreading diameter was found to be 11 [μm]. Thus, the simulations for this droplet size agree with the experimental results.

4.9 Potential Shortcomings

One shortcoming of the model results is the fact that the liquid droplets introduced to the paper coating are of a somewhat smaller size than the representative elementary volume of the paper coating ($4 \times 4 \times 4$ [μm^3], (Aslannejad et al., 2017)). Due to the fact that simulations were done on the pore-scale, this should not influence the validity of the model results, However, great care should be taken in generalizing the findings of the simulations with a droplet radius much smaller than the REV of the coating layer, as the resulting spreading and infiltration of the those droplets could be heavily dependent on the droplet position. For a more rigorous study of the dependency of the spreading and infiltration of a droplet on the equilibrium contact angle, larger droplets should be used in order to ensure that the results are representative for the entire coating layer. However, in case of this work that was simply unpractical due to time concerns, as increasing the droplet size to above the representative elementary volume would also mean significantly increasing the number of cells in the mesh. The mirroring of the geometry in order to enable a larger mesh size could also have had an impact on the results. However, since the original unmirrored geometry already had a size similar to the representative elementary volume, and since in the simulation results, the droplets barely reached the mirrored parts of the mesh, the influence of the mirroring on the spreading and infiltration results are likely minimal.

Furthermore, currently the inertial effects of a droplet created by the inkjet printing process has been ignored in the simulations, as the droplet is introduced to the paper coating with no initial velocity. The Weber number, expressed in equation (5), for a water droplet would be about 0.7, if an impact velocity of 5 [m s^{-1}] and a characteristic length scale of 2 [μm] (the droplet radius) is assumed. Therefore, the exclusion of inertial effects in the simulations of droplets with a radius of 2 [μm] can be assumed to still lead to reasonable results.

The occurrence of parasitic currents in the application of the Volume of Fluid method combined with the Continuum Surface Force Model presented by Brackbill et al. (1992) in the case of surface tension dominated flows presents another major issue in the validity of the model results. It seems, however, that these parasitic currents are not trivial to avoid, as they are not decreased with increased mesh refinement or smaller computational time steps (Harvie et al., 2006)

4.10 Outlook

At this point, the model used to simulate the spreading and penetration of small droplets on a paper coating layer could still be improved. For example by increasing the droplet and domain size, and by the inclusion of additional processes, such as the evaporation of the droplets and surface charges. Furthermore, more insight into the dynamics of coated paper could be gained by modeling of the interface between the coating layer and the fibrous layer of coated paper. The hydraulic properties of this inter-

face have been characterised using image analysis by Aslannejad et al. (2019a). This work could be used as the basis for a model simulating the interaction between the two layers of coated paper.

In addition to the volume of fluid method, other types of modeling techniques, such as Level-Set or Lagrangian methods (Linder et al., 2015), could be used to model the spreading and penetration of droplets on the coated layer, in order to compare the results of these different techniques. Perhaps this could be used to analyse the significance of the parasitic currents found in the volume of fluid method. If parasitic currents are found to have a significant impact on the spreading and penetration results of the model, a method proposed by Shams et al. (2018), which effectively eliminates the parasitic currents in the Volume of Fluid method, could be employed in order to improve model results.

Furthermore, instead of assuming an equilibrium contact angle boundary condition, the advancing and receding contact angles could be measured for the paper coating material, in order to employ a dynamic contact angle boundary. The effect of the advancing and receding contact angles of the liquid droplets on the spreading and penetration on and into the paper coating layer could then be studied.

5 Conclusion

The Volume of Fluid method was employed to gain an insight in the influence of the equilibrium contact angle of a liquid droplet on that droplet's spreading and penetration on and into a paper coating layer. For this, the *interFoam* solver of the OpenFOAM software package was used to model the two-phase flow of air and liquid in a paper coating for three different equilibrium contact angles ($\theta_{\text{eq}} = 1°$, $\theta_{\text{eq}} = 45°$ and $\theta_{\text{eq}} = 90°$), for a droplet with a larger radius (3 [μm]) also observed in experiments, and for a droplet with ink-like properties. The simulation for a droplet with radius of 3 [μm] was found to have good agreement with experimental results. The hypothesis that a lower equilibrium contact angle would result in more penetration and more spreading of the liquid droplet on the paper coating surface was confirmed. The penetration of the liquid droplet was indeed found to increase with lowering contact angles, and there seemed to be an increase in droplet spreading on the surface of the coating. Thus, the results indicate that for the application of inkjet printing on coated paper, a higher contact angle is preferable, as it leads to a lower likelihood of two adjacent droplets coalescing, which would be detrimental to the resulting print quality. The results for the droplet with ink-like properties were very similar to that of water-like droplets, except that the spreading and penetration processes took place over a longer time scale in case of the ink-like droplet. Since this would allow for more evaporation to take place, the ink-like droplet would be preferable in the application of inkjet printing.

References

Arias, S. and A. Montlaur (2018). "Influence of Contact Angle Boundary Condition on CFD Simulation of T-Junction". In: *Microgravity Science and Technology* 30.4.

Ashish Saha, A. and S. K. Mitra (2009). "Effect of dynamic contact angle in a volume of fluid (VOF) model for a microfluidic capillary flow". In: *Journal of colloid and interface science* 339.2, pp. 461–480.

Aslannejad, H. and S. M. Hassanizadeh (2017). "Study of Hydraulic Properties of Uncoated Paper: Image Analysis and Pore-Scale Modeling". In: *Transport in Porous Media* 120.1, pp. 67–81.

Aslannejad, H., S. M. Hassanizadeh, A. Raoof, D. A. M. de Winter, N. Tomozeiu, and M. T. van Genuchten (2017). "Characterizing the hydraulic properties of paper coating layer using FIB-SEM tomography and 3D pore-scale modeling". In: *Chemical Engineering Science* 160, pp. 275–280.

Aslannejad, H., H. Fathi, S. M. Hassanizadeh, A. Raoof, and N. Tomozeiu (2018). "Movement of a liquid droplet within a fibrous layer: Direct pore-scale modeling and experimental observations". In: *Chemical Engineering Science* 191, pp. 78–86.

Aslannejad, H., S. M. Hassanizadeh, and M. A. Celia (2019a). "Characterization of the Interface Between Coating and Fibrous Layers of Paper". In: *Transport in Porous Media* 127.1, pp. 143–155.

Aslannejad, H., S. Loginov, B. van der Hoek, E.M. Schoonderwoerd, and S.M. Hassanizadeh (2019b). "Liquid droplet imbibition into a thin coating layer: direct pore-scale modeling and experimental observations". In: *Yet to be published.*

Bacri, L. and F. Brochard-Wyart (2000). "Droplet suction on porous media". In: *European Physical Journal E* 3.1, pp. 87–97.

Bosanquet, C. H. (1923). "On the flow of liquids into capillary tubes". In: *Philos.Mag.* 45.267, pp. 525–531.

Brackbill, J. U., D. B. Kothe, and C. Zemach (1992). "A continuum method for modeling surface tension". In: *Journal of Computational Physics* 100.2, pp. 335–354.

Calvert, P. (2001). "Inkjet printing for materials and devices". In: *Chemistry of Materials* 13.10, pp. 3299–3305.

Clarke, A., T. D. Blake, K. Carruthers, and A. Woodward (2002). "Spreading and imbibition of liquid droplets on porous surfaces". In: *Langmuir* 18.8, pp. 2980–2984.

Dam, D. B. van and C. Le Clerc (2004). "Experimental study of the impact of an ink-jet printed droplet on a solid substrate". In: *Physics of Fluids* 16.9, pp. 3403–3414.

Daniel, R. C. and J. C. Berg (2006). "Spreading on and penetration into thin, permeable print media: Application to ink-jet printing". In: *Advances in Colloid and Interface Science* 123-126.SPEC. ISS. Pp. 439–469.

De Gans, B. ., P. C. Duineveld, and U. S. Schubert (2004). "Inkjet printing of polymers: State of the art and future developments". In: *Advanced Materials* 16.3, pp. 203–213.

De Gennes, P. G. (1985). "Wetting: Statics and dynamics". In: *Reviews of Modern Physics* 57.3, pp. 827–863.

Deshpande, S. S., L. Anumolu, and M. F. Trujillo (2012). "Evaluating the performance of the two-phase flow solver interFoam". In: *Computational Science and Discovery* 5.1.

Ferrari, A., J. Jimenez-Martinez, T. L. Borgne, Y. Méheust, and I. Lunati (2015). "Challenges in modeling unstable two-phase flow experiments in porous micromodels". In: *Water Resources Research* 51.3, pp. 1381–1400.

Greenshields, C. J. (2018). *OpenFOAM User Guide, version 6*. https://cfd.direct/openfoam/user-guide/.

Harvie, D. J. E., M. R. Davidson, and M. Rudman (2006). "An analysis of parasitic current generation in Volume of Fluid simulations". In: *Applied Mathematical Modelling* 30.10, pp. 1056–1066.

Holzinger, G. (2018). *OpenFOAM: A little User-Manual*. https://github.com/ParticulateFlow/osccar-doc.

Huh, C. and L. E. Scriven (1971). "Hydrodynamic model of steady movement of a solid/liquid/fluid contact line". In: *Journal of colloid and interface science* 35.1, pp. 85–101.

Johnson Jr., R. E. and R. H. Dettre (1964). "Contact angle hysteresis. III. Study of an idealized heterogeneous surface". In: *Journal of Physical Chemistry* 68.7, pp. 1744–1750.

Järnström, J., M. Väisänen, R. Lehto, A. Jäsberg, J. Timonen, and J. Peltonen (2010). "Effect of latex on surface structure and wetting of pigment coatings". In: *Colloids and Surfaces A: Physicochemical and Engineering Aspects* 353.2-3, pp. 104–116.

Kettle, J., T. Lamminmäki, and P. Gane (2010). "A review of modified surfaces for high speed inkjet coating". In: *Surface and Coatings Technology* 204.12-13, pp. 2103–2109.

Le, H. P. (1998). "Progress and trends in ink-jet printing technology". In: *Journal of Imaging Science and Technology* 42.1, pp. 49–62.

Linder, N., A. Criscione, I. V. Roisman, H. Marschall, and C. Tropea (2015). "3D computation of an incipient motion of a sessile drop on a rigid surface with contact angle hysteresis". In: *Theoretical and Computational Fluid Dynamics* 29.5-6, pp. 373–390.

Lucas, R. (1918). "Ueber das Zeitgesetz des kapillaren Aufstiegs von Flüssigkeiten". In: *Kolloid-Zeitschrift* 23.1, pp. 15–22.

Malgarinos, I., N. Nikolopoulos, M. Marengo, C. Antonini, and M. Gavaises (2014). "VOF simulations of the contact angle dynamics during the drop spreading: Standard models and a new wetting force model". In: *Advances in Colloid and Interface Science* 212, pp. 1–20.

Nakamura, M., A. Kobayashi, F. Takagi, A. Watanabe, Y. Hiruma, K. Ohuchi, Y. Iwasaki, M. Horie, I. Morita, and S. Takatani (2005). "Biocompatible inkjet printing technique for designed seeding of individual living cells". In: *Tissue engineering* 11.11-12, pp. 1658–1666.

Ridgway, C. J., P. A. C. Gane, and J. Schoelkopf (2002). "Effect of capillary element aspect ratio on the dynamic imbibition within porous networks". In: *Journal of colloid and interface science* 252.2, pp. 373–382.

Schoelkopf, J., C. J. Ridgway, P. A. C. Gane, G. P. Matthews, and D. C. Spielmann (2000). "Measurement and network modeling of liquid permeation into compacted mineral blocks". In: *Journal of colloid and interface science* 227.1, pp. 119–131.

Scoutaris, N., S. Ross, and D. Douroumis (2016). "Current Trends on Medical and Pharmaceutical Applications of Inkjet Printing Technology". In: *Pharmaceutical Research* 33, pp. 1799–1816.

Shams, M., A. Q. Raeini, M. J. Blunt, and B. Bijeljic (2018). "A numerical model of two-phase flow at the micro-scale using the volume-of-fluid method". In: *Journal of Computational Physics* 357, pp. 159–182.

Sirringhaus, H., T. Kawase, R. H. Friend, T. Shimoda, M. Inbasekaran, W. Wu, and E. P. Woo (2000). "High-resolution inkjet printing of all-

polymer transistor circuits". In: *Science* 290.5499, pp. 2123–2126.

Sorbie, K. S., Y. Z. Wu, and S. R. McDougall (1995). "The extended washburn equation and its application to the oil/water pore doublet problem". In: *Journal of colloid and interface science* 174.2, pp. 289–301.

Starov, V. M., S. R. Kostvintsev, V. D. Sobolev, M. G. Velarde, and S. A. Zhdanov (2002a). "Spreading of liquid drops over dry porous layers: Complete wetting case". In: *Journal of colloid and interface science* 252.2, pp. 397–408.

Starov, V. M., S. R. Kosvintsev, V. D. Sobolev, M. G. Velarde, and S. A. Zhdanov (2002b). "Spreading of liquid drops over saturated porous layers". In: *Journal of colloid and interface science* 246.2, pp. 372–379.

Starov, V. M., S. A. Zhdanov, and M. G. Velarde (2002c). "Spreading of liquid drops over thick porous layers: Complete wetting case". In: *Langmuir* 18.25, pp. 9744–9750.

Szekely, J., A. W. Neumann, and Y. K. Chuang (1971). "The rate of capillary penetration and the applicability of the washburn equation". In: *Journal of colloid and interface science* 35.2, pp. 273–278.

Tadmor, R. (2004). "Line energy and the relation between advancing, receding, and Young contact angles". In: *Langmuir* 20.18, pp. 7659–7664.

Tan, H. (2017). "Absorption of picoliter droplets by thin porous substrates". In: *AIChE Journal* 63.5, pp. 1690–1703.

Tryggvason, G., A. Esmaeeli, J. Lu, and S. Biswas (2006). "Direct numerical simulations of gas/liquid multiphase flows". In: *Fluid Dynamics Research* 38.9, pp. 660–681.

Versteeg, H. K. and W. Malalasekera (2007). *An Introduction to Computational Fluid Dynamics. The Finite Volume Method*. 2nd ed. Pearson Education Limited.

Von Bahr, M., F. Tiberg, and B. V. Zhmud (1999). "Spreading dynamics of surfactant solutions". In: *Langmuir* 15.20, pp. 7069–7075.

Von Bahr, M., F. Tiberg, and B. Zhmud (2003). "Oscillations of Sessile Drops of Surfactant Solutions on Solid Substrates with Differing Hydrophobicity". In: *Langmuir* 19.24, pp. 10109–10115.

Washburn, E. W. (1921). "The dynamics of capillary flow". In: *Physical Review* 17.3, pp. 273–283.

Yin, X., H. Aslannejad, E. T. de Vries, A. Raoof, and S. M. Hassanizadeh (2018). "Droplet Imbibition into Paper Coating Layer: Pore-Network Modeling Simulation". In: *Transport in Porous Media* 125.2.

Appendices

A snappyHexMesh Settings

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website: https://openfoam.org
    \\  /    A nd           | Version: 6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version    2.0;
    format     ascii;
    class      dictionary;
    object     snappyHexMeshDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

castellatedMesh true;
snap          true;
addLayers     false;

geometry
{
    STL.stl
    {
        type   triSurfaceMesh;
        name   coating;
        scale  1e-9;
    }
};

castellatedMeshControls
{
    maxLocalCells 100000;
    maxGlobalCells 2000000;
    minRefinementCells 0;
    nCellsBetweenLevels 10;

    features
    (
    );

    refinementSurfaces
    {
        coating
        {
            level (1 1);
        }
    }

    resolveFeatureAngle 30;

    refinementRegions
    {
```

```
    }

    locationInMesh (5e-6 4e-6 3e-6);
    allowFreeStandingZoneFaces true;
}

snapControls
{
    nSmoothPatch  3;
    tolerance     4.0;
    nSolveIter    10;
    nRelaxIter    5;
}

addLayersControls
{
}

meshQualityControls
{
    maxNonOrtho       65;
    maxBoundarySkewness 4;
    maxIntervalSkewness 4;
    maxConcave        80;
    minVol            -1e30;
    minTetQuality     1e-30;
    minArea           -1;
    minTwist          0.05;
    minDeterminant    0.001;
    minFaceWeight     0.05;
    minVolRatio       0.01;
    minTriangleTwist  -1;
    nSmoothScale      1;
    errorReduction    0.75;
}

mergeTolerance 1e-6;

// ************************************************************************* //
```

B checkMesh Results

B.1 Mesh for simulations with droplet radius 2 [μm]

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website: https://openfoam.org
    \\  /    A nd           | Version: 6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
Build  : 6-d3fd147e6c65
Exec   : checkMesh -allGeometry -allTopology
Date   : Mar 18 2019
Time   : 01:46:06
Host   : "DESKTOP-GUUH61V"
PID    : 277
I/O    : uncollated
Case   : /mnt/d/OpenFOAM/dC_mirr_cA45_r2_run
```

```
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster (
    fileModificationSkew 10)
allowSystemOperations : Allowing user-supplied system call operations

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
Create time

Create polyMesh for time = 0

Enabling all (cell, face, edge, point) topology checks.

Enabling all geometry checks.

Time = 0

Mesh stats
    points:          3337927
    faces:           8742037
    internal faces:  7897167
    cells:           2747477
    faces per cell:  6.056176
    boundary patches: 7
    point zones:     0
    face zones:      0
    cell zones:      0

Overall number of cells of each type:
    hexahedra:    2394093
    prisms:       66273
    wedges:       0
    pyramids:     0
    tet wedges:   8905
    tetrahedra:   60
    polyhedra:    278146
    Breakdown of polyhedra by number of faces:
        faces  number of cells
            4  78109
            5  89983
            6  29043
            9  32196
           12  24445
           15  17042
           18  7270
           21  58

Checking topology...
    Boundary definition OK.
    Cell to face addressing OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Topological cell zip-up check OK.
    Number of identical duplicate faces (baffle faces): 10
  <<Number of duplicate (not baffle) faces found: 4. This might indicate a problem.
  <<Number of faces with non-consecutive shared points: 12. This might indicate a problem.
  <<Writing 44 faces with non-standard edge connectivity to set edgeFaces
  <<Writing 4 cells with zero or one non-boundary face to set oneInternalFaceCells
```

```
  <<Writing 1351 cells with two non-boundary faces to set twoInternalFacesCells
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
                  Patch   Faces  Points               Surface topology Bounding box
                     Up   15471   18489 ok (non-closed singly connected) (-4.1e-07 2.90981e-06
                         -2.06e-06) (6.03e-06 2.925e-06 4.38e-06)
                   Down   11449   11664 ok (non-closed singly connected) (-4.1e-07 -4.2e-06 -2.06e
                       -06) (6.03e-06 -4.2e-06 4.38e-06)
                  Right   16579   18661 ok (non-closed singly connected) (6.03e-06 -4.2e-06 -2.06e
                      -06) (6.03e-06 2.925e-06 4.38e-06)
                   Left   16144   18217 ok (non-closed singly connected) (-4.11735e-07 -4.2e-06
                       -2.06e-06) (-3.92535e-07 2.925e-06 4.38e-06)
                  Front   17339   19359 ok (non-closed singly connected) (-4.1e-07 -4.2e-06
                      -2.06179e-06) (6.03e-06 2.925e-06 -2.04474e-06)
                   Back   17623   19857 ok (non-closed singly connected) (-4.1e-07 -4.2e-06 4.38e
                       -06) (6.03e-06 2.925e-06 4.38e-06)
                  Fiber  750265  865557 multiply connected (shared edge) (-4.11735e-07 -1.14508e-09
                       -2.06179e-06) (6.03e-06 2.925e-06 4.38e-06)
  <<Writing 142 conflicting points to set nonManifoldPoints

Checking geometry...
    Overall domain bounding box (-4.11735e-07 -4.2e-06 -2.06179e-06) (6.03e-06 2.925e-06 4.38e-06)
    Mesh has 3 geometric (non-empty/wedge) directions (1 1 1)
    Mesh has 3 solution (non-empty) directions (1 1 1)
    Boundary openness (7.610137e-16 1.152707e-15 -6.461022e-16) OK.
    Max cell openness = 4.377549e-16 OK.
    Max aspect ratio = 8.219193 OK.
    Minimum face area = 3.230612e-18. Maximum face area = 4.92094e-15. Face area magnitudes OK.
    Min volume = 1.211508e-24. Max volume = 2.705508e-22. Total volume = 2.575798e-16. Cell volumes
         OK.
    Mesh non-orthogonality Max: 57.31835 average: 9.771364
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 3.927094 OK.
    Coupled point location match (average 0) OK.
 ***Error in face tets: 2 faces with low quality or negative volume decomposition tets.
  <<Writing 2 faces with low quality or negative volume decomposition tets to set
      lowQualityTetFaces
    Min/max edge length = 5e-11 8.601272e-08 OK.
   *There are 24407 faces with concave angles between consecutive edges. Max concave angle =
       79.95797 degrees.
  <<Writing 24407 faces with concave angles to set concaveFaces
    Face flatness (1 = flat, 0 = butterfly) : min = 0.2180438 average = 0.997325
   *There are 469 faces with ratio between projected and actual area < 0.8
    Minimum ratio (minimum flatness, maximum warpage) = 0.2180438
  <<Writing 469 warped faces to set warpedFaces
    Cell determinant (wellposedness) : minimum: 0 average: 12.6936
 ***Cells with small determinant (< 0.001) found, number of cells: 1992
  <<Writing 1992 under-determined cells to set underdeterminedCells
 ***Concave cells (using face planes) found, number of cells: 79886
  <<Writing 79886 concave cells to set concaveCells
    Face interpolation weight : minimum: 0.06312432 average: 0.4667363
    Face interpolation weight check OK.
    Face volume ratio : minimum: 0.03877051 average: 0.8331341
    Face volume ratio check OK.

Failed 3 mesh checks.
```

```
End
```

B.2 Mesh for simulations with droplet radius 3 [μm]

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website: https://openfoam.org
    \\  /    A nd           | Version: 6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
Build  : 6-f934eabb0d52
Exec   : checkMesh -allGeometry -allTopology
Date   : Mar 18 2019
Time   : 17:29:47
Host   : "LAPTOP-LFI351UQ"
PID    : 310
I/O    : uncollated
Case   : /mnt/c/Users/Bram/WIN_OpenFOAM/dC_r3
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster (
    fileModificationSkew 10)
allowSystemOperations : Allowing user-supplied system call operations

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
Create time

Create polyMesh for time = 0

Enabling all (cell, face, edge, point) topology checks.

Enabling all geometry checks.

Time = 0

Mesh stats
    points:          11629022
    faces:           30308873
    internal faces:  27280384
    cells:           9510158
    faces per cell:  6.055552
    boundary patches: 7
    point zones:     0
    face zones:      0
    cell zones:      0

Overall number of cells of each type:
    hexahedra:    8196768
    prisms:       250168
    wedges:       0
    pyramids:     0
    tet wedges:   33591
    tetrahedra:   240
    polyhedra:    1029391
    Breakdown of polyhedra by number of faces:
        faces  number of cells
            4  296714
            5  338853
```

```
           6   103809
           9   107551
          11   1
          12   90387
          15   64042
          18   27816
          21   218

Checking topology...
    Boundary definition OK.
    Cell to face addressing OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Topological cell zip-up check OK.
    Number of identical duplicate faces (baffle faces): 66
  <<Number of duplicate (not baffle) faces found: 4. This might indicate a problem.
  <<Number of faces with non-consecutive shared points: 93. This might indicate a problem.
  <<Writing 300 faces with non-standard edge connectivity to set edgeFaces
  <<Writing 19 cells with zero or one non-boundary face to set oneInternalFaceCells
  <<Writing 3795 cells with two non-boundary faces to set twoInternalFacesCells
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
                 Patch   Faces  Points               Surface topology Bounding box
                    Up   39732   46905 ok (non-closed singly connected) (-2.85e-06 2.90842e-06
                        -4.06e-06) (6.03e-06 2.92656e-06 4.7e-06)
                  Down   21608   21903 ok (non-closed singly connected) (-2.85e-06 -9.13e-06 -4.06
                      e-06) (6.03e-06 -9.13e-06 4.7e-06)
                 Right   38975   44130 ok (non-closed singly connected) (6.03e-06 -9.13e-06 -4.06e
                     -06) (6.03e-06 2.925e-06 4.7e-06)
                  Left   40178   45857 ok (non-closed singly connected) (-2.85e-06 -9.13e-06 -4.06
                      e-06) (-2.85e-06 2.925e-06 4.70504e-06)
                 Front   43206   48956 ok (non-closed singly connected) (-2.85e-06 -9.13e-06
                     -4.06058e-06) (6.03e-06 2.925e-06 -4.04559e-06)
                  Back   43145   48900 ok (non-closed singly connected) (-2.85e-06 -9.13e-06 4.7e
                      -06) (6.03e-06 2.925e-06 4.7e-06)
                 Fiber 2801645 3228148 multiply connected (shared edge) (-2.8515e-06 -2.92723e-06
                     -4.06058e-06) (6.03e-06 2.92656e-06 4.70504e-06)
  <<Writing 604 conflicting points to set nonManifoldPoints

Checking geometry...
    Overall domain bounding box (-2.8515e-06 -9.13e-06 -4.06058e-06) (6.03e-06 2.92656e-06 4.70504e
        -06)
    Mesh has 3 geometric (non-empty/wedge) directions (1 1 1)
    Mesh has 3 solution (non-empty) directions (1 1 1)
    Boundary openness (-6.905506e-17 5.627141e-15 -1.801078e-15) OK.
    Max cell openness = 5.282272e-16 OK.
    Max aspect ratio = 8.346021 OK.
    Minimum face area = 6.841758e-18. Maximum face area = 4.480703e-15. Face area magnitudes OK.
    Min volume = 1.095882e-24. Max volume = 3.039531e-22. Total volume = 7.951422e-16. Cell volumes
         OK.
    Mesh non-orthogonality Max: 58.38252 average: 10.12288
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 3.963375 OK.
    Coupled point location match (average 0) OK.
 ***Error in face tets: 2 faces with low quality or negative volume decomposition tets.
  <<Writing 2 faces with low quality or negative volume decomposition tets to set
```

```
      lowQualityTetFaces
    Min/max edge length = 1.226418e-10 8.762604e-08 OK.
   *There are 90057 faces with concave angles between consecutive edges. Max concave angle =
       79.93306 degrees.
  <<Writing 90057 faces with concave angles to set concaveFaces
    Face flatness (1 = flat, 0 = butterfly) : min = 0.2505891 average = 0.9970872
   *There are 1871 faces with ratio between projected and actual area < 0.8
    Minimum ratio (minimum flatness, maximum warpage) = 0.2505891
  <<Writing 1871 warped faces to set warpedFaces
    Cell determinant (wellposedness) : minimum: 0 average: 13.12376
 ***Cells with small determinant (< 0.001) found, number of cells: 5446
  <<Writing 5446 under-determined cells to set underdeterminedCells
 ***Concave cells (using face planes) found, number of cells: 291480
  <<Writing 291480 concave cells to set concaveCells
    Face interpolation weight : minimum: 0.06695206 average: 0.4642196
    Face interpolation weight check OK.
    Face volume ratio : minimum: 0.0315843 average: 0.8206256
    Face volume ratio check OK.

Failed 3 mesh checks.

End
```

C Simulation Settings

C.1 Initial p_rgh

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website: https://openfoam.org
    \\  /    A nd           | Version: 6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      binary;
    class       volScalarField;
    location    "0";
    object      p_rgh;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -1 -2 0 0 0 0];

internalField uniform 0;

boundaryField
{
    Up
    {
        type            fixedFluxPressure;
        gradient        uniform 0;
        value           uniform 0;
    }
    Down
    {
        type            totalPressure;
```

```
        rho           rho;
        psi           none;
        gamma         1;
        p0            uniform 0;
        value         uniform 0;
    }
    Right
    {
        type          symmetryPlane;
    }
    Left
    {
        type          totalPressure;
        rho           rho;
        psi           none;
        gamma         1;
        p0            uniform 0;
        value         uniform 0;
    }
    Front
    {
        type          totalPressure;
        rho           rho;
        psi           none;
        gamma         1;
        p0            uniform 0;
        value         uniform 0;
    }
    Back
    {
        type          symmetryPlane;
    }
    coating
    {
        type          fixedFluxPressure;
        gradient      uniform 0;
        value         uniform 0;
    }
}


// ************************************************************************* //
```

C.2 Initial U

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website: https://openfoam.org
    \\  /    A nd           | Version: 6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      binary;
    class       volVectorField;
    location    "0";
    object      U;
```

```
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];

internalField uniform (0 0 0);

boundaryField
{
    Up
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
    Down
    {
        type            pressureInletOutletVelocity;
        value           uniform (0 0 0);
    }
    Right
    {
        type            symmetryPlane;
    }
    Left
    {
        type            pressureInletOutletVelocity;
        value           uniform (0 0 0);
    }
    Front
    {
        type            pressureInletOutletVelocity;
        value           uniform (0 0 0);
    }
    Back
    {
        type            symmetryPlane;
    }
    coating
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
}


// ************************************************************************* //
```

C.3 Initial alpha.water

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website: https://openfoam.org
    \\  /    A nd           | Version: 6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version    2.0;
```

```
    format      ascii;
    class       volScalarField;
    location    "0";
    object      alpha.water;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 0 0 0 0];

internalField uniform 0;

boundaryField
{
    Up
    {
        type           zeroGradient;
    }
    Down
    {
        type           inletOutlet;
        inletValue     uniform 0;
        value          uniform 0;
    }
    Right
    {
        type           symmetryPlane;
    }
    Left
    {
        type           inletOutlet;
        inletValue     uniform 0;
        value          uniform 0;
    }
    Front
    {
        type           inletOutlet;
        inletValue     uniform 0;
        value          uniform 0;
    }
    Back
    {
        type           symmetryPlane;
    }
    coating
    {
        type           constantAlphaContactAngle;
        theta0         45;
        limit          gradient;
        value          uniform 0;
    }
}


// ************************************************************************* //
```

C.4 fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
```

```
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website: https://openfoam.org
    \\  /    A nd           | Version: 6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    "alpha.water.*"
    {
        nAlphaCorr      2;
        nAlphaSubCycles 1;
        cAlpha          0.8;

        MULESCorr       yes;
        nLimiterIter    3;

        solver          PBiCG;
        preconditioner DILU;
        tolerance       1e-8;
        relTol          0;
    }

    pcorr
    {
        solver                PCG;
        preconditioner        DIC;
        tolerance             1e-06;
        relTol                0;
    }

    pcorrFinal
    {
        $pcorr;
    }

    p_rgh
    {
        solver                PCG;
        preconditioner        DIC;
        tolerance             1e-06;
        relTol                0;
    }

    p_rghFinal
    {
        $p_rgh;
    }

    U
```

```
    {
        solver          PBiCG;
        preconditioner DILU;
        tolerance       1e-06;
        relTol          0;
    }

    UFinal
    {
        $U;
    }
}

PIMPLE
{
    momentumPredictor no;
    nOuterCorrectors  1;
    nCorrectors       2;
    nNonOrthogonalCorrectors 1;
}

relaxationFactors
{
    equations
    {
        ".*" 1;
    }
}


// ************************************************************************* //
```

C.5 fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website: https://openfoam.org
    \\  /    A nd           | Version: 6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version    2.0;
    format     ascii;
    class      dictionary;
    location   "system";
    object     fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default Euler;
}

gradSchemes
{
    default       Gauss linear;
```

```
}

divSchemes
{
    div(rhoPhi,U) Gauss upwind;
    div(phi,alpha) Gauss vanLeer;
    div(phirb,alpha) Gauss linear;
    div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default       Gauss linear corrected;
}

interpolationSchemes
{
    default       linear;
}

snGradSchemes
{
    default       corrected;
}


// ************************************************************************* //
```

C.6 controlDict

```
/*--------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website: https://openfoam.org
    \\  /    A nd           | Version: 6
     \\/     M anipulation  |
\*--------------------------------------------------------------------------*/
FoamFile
{
    version    2.0;
    format     ascii;
    class      dictionary;
    location   "system";
    object     controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application    interFoam;

startFrom      latestTime;

startTime      0;

stopAt         endTime;

endTime        5e-5;

deltaT         1e-11;
```

```
writeControl   adjustableRunTime;

writeInterval 5e-7;

purgeWrite     0;

writeFormat    binary;

writePrecision 7;

writeCompression uncompressed;

timeFormat     general;

timePrecision  6;

runTimeModifiable yes;

adjustTimeStep yes;

maxCo          0.7;
maxAlphaCo     0.7;

maxDeltaT      1;


// ************************************************************************* //
```