Delft University of Technology

Literature study

Tuning the surface tension in SPH independently of the fluid rheology

Author:
Kjeld Broekema (4378644)

October 22, 2020

Table of contents

1 Introduction

Spray drying is a widely used technology in industries such as the food, paint and pharmaceutical industry. Spray drying converts droplets into powders, through hot air. The droplets will enter at the top of the spray chamber, where the droplets are exposed to a heated airflow, introduced by air inlets. The resulting powder leaves the spray chamber at the bottom, see Figure 1. During the drying, droplets will start colliding, coalesce, and agglomerate. Furthermore, droplets in different drying stadiums are involved, which increases the complexity. This will significantly influence the end-product in terms of consistency and quality; e.g. particle-size and morphology. The morphology is of great importance since this determines the properties of the end-product, such as density and solubility, which influences the end-product usability.

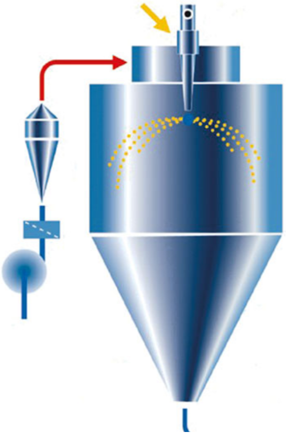

Figure 1: A schematic overview of a spray dryer, where the yellow arrow is the droplet inlet and the red arrow represents the hot air inlet [19].

The current way to get the desired powder properties is by trial-and-error. A new product involves time-consuming and costly production-trials, where in case the end-product is not satisfactory it will be discarded. A way to avoid such costly production-trials is by introducing computer models that can simulate a spray drying process. However, current models lack the complexity, that is needed to properly simulate a drying process. To be able to provide more realistic simulations, a deeper understanding of the complex internal processes at droplet scale level is needed. We will focus in simulating more realistic droplet collisions. An obstacle for simulating realistic droplet collisions is the implementation of the correct "rheological" fluid properties, where rheology is the science of deformation and flow. With traditional numerical methods (e.g. finite elements and computational finite difference) the fluid rheology is added through a constitutive equation[1]. However, finding the right constitutive equation and solving it, leads to a complex and troublesome task. To avoid usage of such complex constitutive equations, a different method will be investigated, namely Particle Interaction Forces (PIF). Important to note is that PIF can only be implemented in particle based numerical methods. The considered PIF generates an attraction force for distant particles and repulsion for close particles. PIF is implemented in the numerical method Smoothed Particle Hydrodynamics (SPH). Aside from the ability to implement PIF, other advantages of SPH are: the ability to deal with large deformations, complex (free) surfaces and multi-phases. A disadvantage, however, is the difficulty in obtaining second-order global convergence

[1] A constitutive equation approximates the response of a specific material to an external stimulus, i.e. relates stresses in the material to strains and strain rates

[27][2]. In Linde [21] it has been shown that PIF can be used to simulate the fluid rheology and although in [30, 3, 15] it has been shown that PIF also influences the surface tension, it is yet unknown to what extend PIF will affect the surface tension. At the droplet scale level, the surface tension is an important factor in determining if droplets will coalesce or bounce [34, 14]. Hence, to achieve realistic simulations of colliding droplets, the fluid rheology and surface tension has to be correctly modelled.

The aim of this thesis is to investigate the influence of PIF on the surface tension in order to simulate realistic fluids; i.e., how can PIF be used to tune the fluid rheology and surface tension independently?

To be able to answer the main question, we will first briefly explain the concept of rheology (Chapter 2). After which we will give a detailed explanation of SPH and discuss the convergence issue (chapter 3). Next, we will investigate the possible tuning methods for the surface tension (chapter 4). In chapter 5, we will first explain the (analytical) methods that will be used to validate the numerical implementation and to measure the surface tension. Furthermore, we will explain the small amplitude oscillatory shear (SAOS) method that will be used to measure the fluid rheology. Lastly, we will conclude the literature study by stating the main research question and the sub questions.

[2] Order of convergence represents how fast the numerical method approaches its limit, i.e. how fast the approximated function equals the function. For a numerical method, to keep the computational time reasonable, a minimum of second order converge is needed.

2 Rheology

In this section, we will first introduce the Navier-Stokes equations for incompressible Newtonian fluids. Furthermore, the concept of non-Newtonian fluids is briefly explained. Lastly, viscoelastic fluids are explained.

2.1 Navier-Stokes equations

We will omit the derivation of the momentum and continuum equations and just state them. However, if desired the derivation can be found in [25]. The equations are given by

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p - \nabla \cdot \underline{\underline{\tau}} + \mathbf{F}_{body} \tag{1}$$

$$\frac{Dp}{Dt} = -\rho \nabla \cdot \mathbf{u}, \tag{2}$$

where $\frac{D}{Dt}$ is the material derivative, ρ is the density, p is the pressure, $\mathbf{F}_{body}$ is the collection of all contributing body forces (e.g. gravity, surface tension and boundary forces), $\mathbf{u}$ is the velocity vector and $\underline{\underline{\tau}}$ is the stress tensor. The stress tensor relates the fluid stress to deformation. An expression that specifies $\underline{\underline{\tau}}$ is called a constitutive equation. For an incompressible Newtonian fluid ($\nabla \cdot \mathbf{u} = 0$) the constitutive equation is given by [25][3]

$$\underline{\underline{\tau}} = -\mu \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right], \tag{4}$$

where μ is the viscosity. Now substituting expression (4) into expression (1) and assuming that the only body force is due to surface tension yields[4]

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{F}_{surf}, \tag{5}$$

Furthermore, the continuum equation for an incompressible fluid reduces to

$$\nabla \cdot \mathbf{u} = 0. \tag{6}$$

Equations (5) and (6) are known as the Navier-Stokes equations for an incompressible Newtonian fluid. A fluid is called non-Newtonian if it exhibits behavior that is not predicted by the Newtonian constitutive equation given by expression (4) [25].

2.2 Viscoelastic fluids

Rheology is the science of deformation and flow of all kinds of material [25]. An important step to accurately predict the fluid rheology is the formulation of a constitutive equation. Determining a constitutive equation for non-Newtonian fluid is a difficult challenge mainly because non-Newtonian fluids can behave differently over time when a certain stress or strain rate is applied. For example, when applying large stresses, one fluid can exhibit shear-thinning, while the other fluid exhibits shear-thickening[5]. Furthermore, rheological properties can be time and temperature dependent. Apart from the (viscous)

[3] Expression (4) is also known as the (viscous) Newtonian constitutive equation and is often written in the form of

$$\underline{\underline{\tau}} = -\mu \underline{\underline{\dot{\gamma}}} \tag{3}$$

, where $\underline{\underline{\dot{\gamma}}}$ is the rate of strain tensor.

[4] Rewriting into Einsteins notation is used to get expression (5) , see appendix A for an introduction to tensor calculus.

[5] Shear-thinning is characterized by a decreasing viscosity due to increased shear rates, while shear-thickening means an increasing viscosity when the shear rates are increased [25]. Viscosity is a measure of a fluid's resistance to flow.

Newtonian constitutive equation, a constitutive equation for a purely linear elastic material is given by Hooke's law of elasticity [25]:

$$\underline{\underline{\tau}} = -G\underline{\underline{\gamma}}, \tag{7}$$

where G is the elastic modulus. In this thesis, we will consider non-Newtonian fluids that can display both viscous and elastic behaviour[6], i.e. viscoelastic fluids. Viscoelastic behaviour for example can be measured by applying an oscillatory strain on the liquid and measuring the resulting stress or vice versa. For a purely elastic material, the stress and strain responses are in phase and for a purely viscous material, the strain will have a 90° phase shift. A viscoelastic material will have a delayed strain response in between that of 0° phase (purely elastic material) or a 90° phase shift (purely viscous material), see Figure 2. The viscous and elastic phase shifts can easily be deduced by using expressions (3) and (7). That is, considering an oscillating stress in the form of a sine function. We then observe from expression (7) that the strain is proportional to the stress up to a constant, which explains the zero phase shift between the responses. However, for a viscous material, the stress and strain are related by the derivative of the strain. The derivative of a sine function results in a cosine, hence a 90° phase shift.

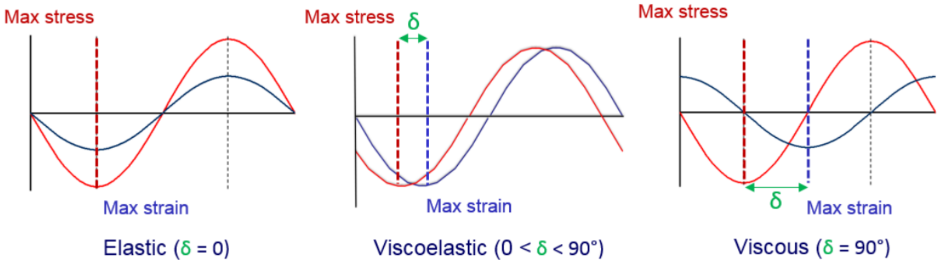

Figure 2: Stress and strain response for a purely elastic, purely viscous and a viscoelastic material, where δ is the phase shift [16].

The phase shift that determines the viscoelastic behaviour can be expressed in terms of the storage modulus G' and the loss modulus G'' by

$$\tan\delta = \frac{G''}{G'}, \tag{8}$$

where the storage modulus is the in-phase (elastic) component and the loss modulus is the out-of-phase (viscous) component. Furthermore, for a sine changing shear strain[7], the stress response can be represented in terms of the storage and loss modulus:

$$\tau_{xy}(t) = \gamma_0 \left[G'(\omega)\sin\omega t + G''(\omega)\cos\omega t\right], \tag{9}$$

where ω is the frequency and γ_0 is the strain amplitude.

[6] Note, that we consider the fluids only in its linear regime, i.e in the regime where the stress and strain have a linear relation.

[7] i.e. $\gamma(t) = \gamma_0 sin(\omega t)$.

3 SPH: Smoothed Particle Hydrodynamics

In this section, based on the articles of Liu [22] and Hirschler et al. [14] we will explain the basics behind SPH and discus the method that is used to simulate an incompressible flow, namely Incompressible SPH (ISPH). Furthermore, we will derive the SPH discretization of the incompressible Newtonian Navier-Stokes equations. After, which the implementation of a free surface boundary condition is discussed. Then, the consistency of the numerical method is discussed and finally the choice of the smoothing functions is explained.

3.1 Basics SPH

In SPH the computational domain is represented by a finite set of particles, where each particle carries the material properties. The interaction between the particles is controlled by a smoothing function; the smoothing function defines the mutual influence between particles based on their distance. However, we will first consider the smoothing function for a continuous field. The smoothing function is based on the following identity [22]:

$$f(\mathbf{r}) = \int_{\Omega} f(\mathbf{r}')\delta(\mathbf{r}-\mathbf{r}')d\mathbf{r}', \tag{10}$$

where $\mathbf{r}$ is the position vector and $\delta(\mathbf{r}-\mathbf{r}')$ is the Dirac delta function[8]. However, the delta function has only "one point" support and hence cannot be used to establish a discrete numerical model [22]. Now, replacing the Delta function with a smoothing function $W(r,h)$, we get the smoothed function

$$f(\mathbf{r}) \approx \langle f(\mathbf{r})\rangle := \int_{\Omega} f(\mathbf{r}')W(\mathbf{r}-\mathbf{r}',h)d\mathbf{r}', \tag{11}$$

where h is the smoothing length, which is proportional to the radius of the kernel support domain Ω, see Figure 3 for an illustration of the smoothed function.

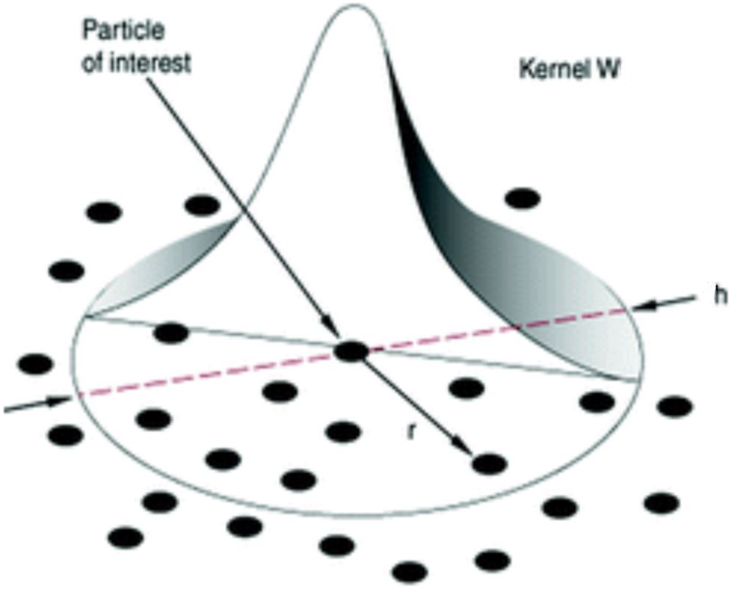

Figure 3: Illustration of a smoothed function/kernel in 2D [22].

[8] The Dirac delta function is non zero whenever $\mathbf{r} = \mathbf{r}'$ and everywhere else zero, furthermore $\int_{\Omega} \delta(\mathbf{r}-\mathbf{r}')d\mathbf{r}^{'} = 1$

For a function to be employed as an SPH smoothing function, the function needs to have the following properties [22]:

- Unity: $\int_\Omega W(\mathbf{r}-\mathbf{r}',h)d\mathbf{r}' = 1$
- Compact support: outside of the supported domain the smoothing function should be equal to zero, that is
$$W(\mathbf{r}-\mathbf{r}',h) = 0, \text{ whenever } |\mathbf{r}-\mathbf{r}'| > \kappa h, \tag{12}$$
where κ is a scaling factor. Note: κh is called the cut-off radius.
- Positivity: $W(\mathbf{r}-\mathbf{r}',h) \geq 0$ for any point in the domain.
- Decay: $W(\mathbf{r}-\mathbf{r}',h)$ decreases monotonously as $|\mathbf{r}-\mathbf{r}'|$ increases.
- Delta function approximation: $\lim_{h\to 0} W(\mathbf{r}-\mathbf{r}',h) = \delta(\mathbf{r}-\mathbf{r}')$, such that $< f(\mathbf{r}) >\to f(\mathbf{r})$.
- Symmetric property: smoothing function values for an equal distance size should be of equal value.

Properties and states such as density and velocity for the (discretized field) particles are calculated by an approximation of expression (11), that is

$$\langle f(\mathbf{r}_i)\rangle \approx \sum_{j=1}^{\mathcal{N}} f(\mathbf{r}_j)W(\mathbf{r}_i-\mathbf{r}_j,h)\Delta V_j = \sum_{j=1}^{\mathcal{N}} \frac{m_j}{\rho_j} f(\mathbf{r}_j)W(\mathbf{r}_i-\mathbf{r}_j,h), \tag{13}$$

where $\mathcal{N}$ is the total number of particles within the influence area of particle i, m_j is the corresponding mass of particle j, ρ_j is the density of particle j and ΔV_j is the volume occupied by particle j. In the last step of expression (13) we used that a volume without a fixed shape can be written as a mass to density ratio, that is $\Delta V_j = \frac{m_j}{\rho_j}$. Note that the gradient of expression (13) can be calculated exact by

$$\langle \nabla \cdot f(\mathbf{r}_i)\rangle \approx \sum_{j=1}^{\mathcal{N}} \frac{m_j}{\rho_j} f(\mathbf{r}_j)\cdot \nabla_i W(\mathbf{r}_i-\mathbf{r}_j,h). \tag{14}$$

For convenience, we will later on write $W(\mathbf{r}_i-\mathbf{r}_j,h)$ as W_{ij}. In the literature two different expressions for a particle derivative are used, which can be confusing and leads to wrong application of the expressions. The first expression is given in equation (14) and the second expression is given by

$$\langle \nabla \cdot f(\mathbf{r}_i)\rangle \approx -\sum_{j=1}^{\mathcal{N}} \frac{m_j}{\rho_j} f(\mathbf{r}_j)\cdot \nabla_j W_{ij}. \tag{15}$$

Observe that the first expression has a derivative taken with respect to particle i and the second expression has a derivative with respect to particle j. However, often in literature the notification with which respect the derivative is taken is left out. The two expression are, however equivalent. To show this, first note that the derivatives can be expressed as

$$\begin{aligned} \nabla_i W(\mathbf{r}_i-\mathbf{r}_j,h) &= W'(\mathbf{r}_i-\mathbf{r}_j,h)\frac{\partial(\mathbf{r}_i-\mathbf{r}_j)}{\partial \mathbf{r}_i} = W'(\mathbf{r}_i-\mathbf{r}_j,h)\frac{\mathbf{r}_i-\mathbf{r}_j}{|\mathbf{r}_i-\mathbf{r}_j|} \\ \nabla_j W(\mathbf{r}_i-\mathbf{r}_j,h) &= W'(\mathbf{r}_i-\mathbf{r}_j,h)\frac{\partial(\mathbf{r}_i-\mathbf{r}_j)}{\partial \mathbf{r}_j} = -W'(\mathbf{r}_i-\mathbf{r}_j,h)\frac{\mathbf{r}_i-\mathbf{r}_j}{|\mathbf{r}_i-\mathbf{r}_j|}, \end{aligned} \tag{16}$$

resulting in $\nabla_i W(\mathbf{r}_i-\mathbf{r}_j,h) = -\nabla_j W(\mathbf{r}_i-\mathbf{r}_j,h)$. Both can be used, however a specific mention, with respect to which derivative is taken is needed.

3.2 Incompressible flow in SPH

When considering an incompressible flow, the solution methods can be divided into two classes, namely Weakly Compressible SPH (WCSPH) and Incompressible SPH (ISPH). In this thesis, we will use ISPH. The three reasons for choosing ISPH over WCSPH are [4]:

- WCSPH requires a very short time step in order to enforce low compressibility, leading to large computational times.
- WCSPH can have negative pressures, leading to instability.
- ISPH leads to a more accurate pressure representation.

The ISPH method uses a predictor-corrector scheme in combination with a first-order Euler method in order to solve the incompressible Navier-Stokes equations. In the predictor step, the intermediate velocity (u^*) is calculated by solving the momentum equation excluding the pressure force, for a Newtonian fluid this yields: [30]

$$\mathbf{u}_i^* = \mathbf{u}_i^n + \left(\mu \Delta \mathbf{u}_i^n + \mathbf{F}_{surf}\right) \frac{\Delta t}{\rho}, \tag{17}$$

where $\mathbf{u}_i^n$ is the particle velocity at the previous timestep. Note, that the particle position is not changed in the predictor step. In the corrector step the Pressure Poisson Equation (PPE) is solved to enforce incompressiblilty:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p^{n+1}\right)_i = \frac{\nabla \cdot \mathbf{u}_i^*}{\Delta t}. \tag{18}$$

After the above linear system is (iteratively) solved, the particle velocity and position ($\mathbf{r}$) at the next time step are calculated by [18, 7]

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \left(\frac{1}{\rho} \nabla p^{n+1}\right)_i \Delta t \tag{19}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \left(\frac{\mathbf{u}_i^{n+1} + \mathbf{u}_i^n}{2}\right) \Delta t. \tag{20}$$

In order to increase the stability, the particle shifting method introduced by Xu, Stansby, and Laurence [32] is used. The ISPH method is implemented in the SPH software called "SiPER"[9]. In this software, the linear system given in expression (18) is solved using the algebraic multigrid preconditioner boomerAMG and a Bi-CGSTAB solver from the PETSc library. Lastly, since semi-implicit timestepping is used the time step is limited. The limited time step is given by

$$\Delta t = min\left(\Delta t_{CFL}, \Delta t_{visc}\right), \tag{21}$$

where Δt_{CFL} and Δt_{visc} are given by

$$\begin{aligned} \Delta t_{CFL} &= \alpha_{CFL} \frac{L_0}{u_{max}} \\ \Delta t_{visc} &= \alpha_{diff} \frac{L_0^2}{\nu_{max}}, \end{aligned} \tag{22}$$

with $\alpha_{CFL} = 0.05$, L_0 the initial particle distance, u_{max} is the magnitude of the maximum velocity, $\alpha_{diff} = 0.125$ and ν_{max} the maximum kinematic viscosity [14].

[9] SiPER is a program package from the University of Stuttgart. For more details see their website: https://www.icvt.uni-stuttgart.de/en/research/siper/.

3.3 SPH discretization

The discretization of the pressure term can be rewritten in many different forms, each with it own advantages and disadvantages. However, a detailed comparison about their performance is rarely done [6]. Because of this, we will give the two most used expressions and compare their performance. However, we will first show why direct discretization is not often done in practice. Direct discretization of the pressure term results in

$$\left(\frac{1}{\rho}\nabla p\right)_i \approx \frac{1}{\rho_i}\sum_{j=1}^{\mathcal{N}}\frac{m_j}{\rho_j}p_j\nabla_i W_{ij}. \tag{23}$$

Now, by noting that $\rho\frac{D\mathbf{u}}{Dt} = -\nabla p$ in combination with conservation of linear momentum ($F_i = -F_j$) and only considering the force of particle i on j and vice versa, we have [10]

$$F_i = -\frac{m_i m_j}{\rho_i\rho_j}p_j\nabla_i W_{ij} \neq -\frac{m_j m_i}{\rho_j\rho_i}p_i\nabla_j W_{ij} = \frac{m_j m_i}{\rho_j\rho_i}p_i\nabla_i W_{ij} = -F_j. \tag{25}$$

Note that the inequality follows from $p_i \neq p_j$. Now from expression (25) we can conclude that direct discretization does not conserve linear momentum. To satisfy linear momentum the pressure term is first rewritten, after which the pressure term is discretized. Resulting in a symmetric interaction between the particles.

For the first discrete pressure expression, we note that the pressure gradient can be rewritten as

$$\frac{\nabla p}{\rho} = \nabla\left(\frac{p}{\rho}\right) + p\frac{\nabla\rho}{\rho^2}. \tag{26}$$

This follows from $\nabla\left(\frac{p}{\rho}\right) = \frac{1}{\rho}\nabla(p) + p\nabla\left(\frac{1}{\rho}\right)$ and applying the quotient rule to the last term on the right-hand side. The first discrete pressure term can now be given by

$$\begin{aligned}\left(\frac{\nabla p}{\rho}\right) &\approx \sum_{j=1}^{N}\frac{m_j}{\rho_j}\frac{p_j}{\rho_j}\nabla_i W_{ij} + \frac{p_i}{\rho_i^2}\sum_{j=1}^{N}\frac{m_j}{p_j}p_j\nabla_i W_{ij}\\ &= \sum_{j=1}^{N}m_j\left(\frac{p_j}{\rho_j^2}+\frac{p_i}{\rho_i^2}\right)\nabla_i W_{ij}.\end{aligned} \tag{27}$$

The second expression is given by[11]:

$$\frac{\nabla p}{\rho} \approx \sum_{j=1}^{N}\frac{m_j}{\rho_i\rho_j}\left(p_i + p_j\right)\nabla_i W_{ij}. \tag{30}$$

[10] This results from expression (5), in the case of that the pressure force is the only acting force. Furthermore, we can express F_i as

$$F_i = m_i a_i. \tag{24}$$

[11] This is derived by rewriting ∇p as

$$\nabla p = \nabla p + p\nabla 1, \tag{28}$$

and by noting that the SPH discretization of $\nabla 1$ is given by

$$\nabla 1 \approx \sum_{j=1}^{N}\frac{m_j}{\rho_j}\nabla_i W_{ij}. \tag{29}$$

The SPH discretizations for the PPE are given by:

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p\right)_i \approx \sum_{j=1}^{N} \frac{m_j}{\rho_j}\frac{4}{\rho_j+\rho_i}\frac{p_{ij}\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^2}\cdot \nabla_i W_{ij} \tag{31}$$

$$\left(\nabla \cdot \mathbf{u}^*\right)_i \approx \sum_{j=1}^{\mathcal{N}} \frac{m_j}{\rho_j}\left(\mathbf{u}^*{}_j - \mathbf{u}^*{}_i\right)\cdot \nabla_i W_{ij}, \tag{32}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ for the distance vector between the particles. The derivation of the discrete pressure force is based on the following integral approximant

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p\right) \approx \int \left[\frac{1}{\rho(\mathbf{r}')} + \frac{1}{\rho(\mathbf{r})}\right]\left[p(\mathbf{r}) - p(\mathbf{r}')\right]\frac{(\mathbf{r}-\mathbf{r}')\cdot \nabla_r W_{rr'}}{|\mathbf{r}-\mathbf{r}'|^2} d\mathbf{r}'. \tag{33}$$

Using Taylor series about $\mathbf{r}$, we can verify that the integral approximant is of second order accuracy [23][12]. The integral approximant in SPH form is

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p\right)_i \approx \sum_{j=1}^{\mathcal{N}} \frac{m_j}{\rho_j}\left[\frac{1}{\rho_j} + \frac{1}{\rho_i}\right]\frac{p_{ij}\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^2}\cdot \nabla_i W_{ij}, \tag{34}$$

with $p_{ij} = p_i - p_j$. However, Cleary and Monaghan [8] showed that a correction term is needed, whenever the flux is discontinuous. The correction term is given by

$$\frac{4\frac{1}{\rho_i}\frac{1}{\rho_j}}{\frac{1}{\rho_i}+\frac{1}{\rho_j}}, \tag{35}$$

resulting in the final SPH discretization given in equation (31). Note that the SPH discretization of the intermediate velocity gradient is a result of rewriting with means of the product rule and applying basic SPH discretization. Now using the same correction term given by (35) leads to the following discrete equation for the viscous term in the momentum equation:

$$\left(\frac{\mu}{\rho}\Delta \mathbf{u}\right)_i \approx \sum_{j=1}^{N} \frac{4m_j\left(\mu_i + \mu_j\right)}{\rho_j}\left(\frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^2}\cdot \nabla W_{ij}\right)\left(\mathbf{u}_i - \mathbf{u}_j\right). \tag{36}$$

3.4 Free surface

At the free surface, a boundary condition in the form of a prescribed pressure is used. However, the particles belonging to the free surface first need to be identified. This is done through the usage of a Shepard-kernel [14]:

$$S_i = \sum_{j=1}^{\mathcal{N}} \frac{m_j}{\rho_j} W_{ij}. \tag{37}$$

Due to a truncated kernel, the particles near the free surface will have a lower value. Particles with a value lower than $S_i < 0.78$ are labelled as free surface particles [14]. In our case, we will impose a zero pressure boundary condition. Resulting, that for free surface particles the PPE is replaced by [4]:

$$\left(\nabla \cdot \frac{1}{\rho}\nabla p\right)_i \approx \sum_{j=1}^{\mathcal{N}} m_j \frac{8}{\left(\rho_i + \rho_j\right)^2}\frac{\left(2p_i - p_j\right)\mathbf{r}_{ij}\cdot \nabla W_{ij}}{\|\mathbf{r}_{ij}\|^2}. \tag{38}$$

[12] For a more in depth derivation of the integral in multiple dimensions, see [23]

3.5 Kernel and particle consistency

The consistency[13] of the SPH method depends on the kernel approximation and the particle approximation. Using Taylor expansion the requirements for zero and first order consistency for the kernel approximation are given by [22][14]

$$\int_{\Omega} W(\mathbf{r}-\mathbf{r}', h)d\mathbf{r}' = 1 \tag{39}$$

$$\int_{\Omega} (\mathbf{r}-\mathbf{r}')\, W(\mathbf{r}-\mathbf{r}', h)d\mathbf{r}' = 0. \tag{40}$$

Observe that expressions (39) and (40) are the unity and the symmetric properties of a smoothing function. Since, these properties are already satisfied, it can be concluded that the kernel approximation has up to C^1 consistency in regions without boundaries. For regions truncated with boundaries, the requirements (39) and (40) are not satisfied, leading to lacking even C^0 consistency. In Figure 4, we observe that, due to boundary truncation, $W(\mathbf{r}-\mathbf{r}', h)$ is no longer an even function. As a result $(\mathbf{r}-\mathbf{r}')\, W(\mathbf{r}-\mathbf{r}', h)$ is no longer an odd function and when taking the integral the positive and negative parts no longer cancel each other out. Hence, equation (40) is not equal to zero resulting in the consistency requirements given by the Taylor expansion being violated.

By introducing a normalization factor, C^0 consistency can be restored. Furthermore, with the usage of corrective kernels C^1 consistency for interior regions can be achieved. A 1D example of a corrective kernel approximation for a function $f(x)$ at particle i is given by [22][15].

$$f_i \approx \frac{\int f(x) W_i(x) dx}{\int W_i(x) dx}, \tag{42}$$

from where we can observe that for an interior region the corrective kernel has C^1 consistency and for a boundary region it has C^0 consistency, due to the normalization factor, which makes sure that equation (39) is satisfied.

[13] The order of consistency refers to which n'th order the numerical method can reproduce a polynomial exactly [22].

[14] Note achieving higher order consistency by solemnly relying on a smoothing function is impossible without violating the positivity requirement [22]. A smoothing function, which can be negative is undesirable, since it can result in unphysical solutions, such as negative density and negative energy.

[15] By Taylor expansion of $f(x)$ at a nearby point x_i and multiplying both sides with W and integrating over the entire computation domain, we get

$$\begin{aligned} \int f(x) W_i(x) dx = f_i \int W_i(x) dx + f_{i,x} \int (x - x_i) W_i(x) dx + \\ \frac{f_{i,xx}}{2} \int (x-x_i)^2\, W_i(x) dx + O(\Delta x)^3). \end{aligned} \tag{41}$$

Combining expressions (41) and (42) the consistency statements about the correction kernel can be derived [22].

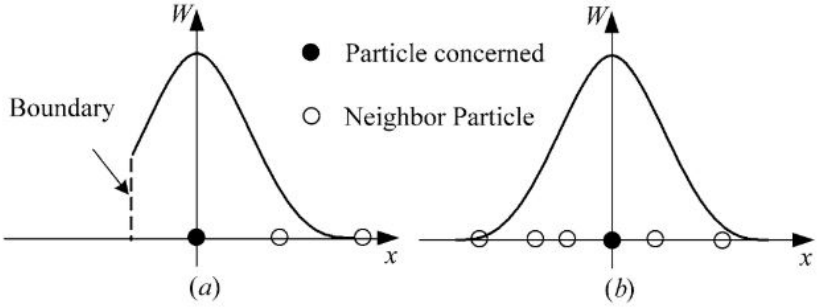

Figure 4: One-dimensional visual representation of the influence of a boundary (a) and particle disorder (b) on the consistency [22].

For the consistency of the particle approximation, we consider the discrete versions of (39) and (40)

$$\sum_{j=1}^{N} W(\mathbf{r}_i - \mathbf{r}_j, h)\Delta V_j = 1 \tag{43}$$

$$\sum_{j=1}^{N} (\mathbf{r}_i - \mathbf{r}_j)\, W(\mathbf{r}_i - \mathbf{r}_j, h)\Delta V_j = 0. \tag{44}$$

In general, C^0 consistency is not achieved, since requirement (43) is violated due to particle disorder or truncation of the support domain by boundaries. A way to restore C^0 consistency is by renormalization [22].

In Di G. Sigalotti et al. [10] the requirement for C^2 consistency is derived[16], that is

$$\langle x^2 \rangle - \langle x \rangle^2 = 0, \tag{45}$$

where $\langle\,\rangle$ is the kernel approximation. Expression (45) is only achieved when $N \to \infty$, $h \to 0$, and $\mathcal{N} \to \infty$, with $\mathcal{N}$ the number of particles within the kernel support and N the total number of particles. However, the increase in kernel particles, whenever the total number of particles increases, demands the use of a Wendland-type kernel [11]. The reason behind this is that when $\mathcal{N}$ attains large values, standard SPH smoothing kernels become unstable due to particle clumping. A Wendland-type kernel avoids particle clumping for all values of $\mathcal{N}$ [9][17]. Furthermore, an error bound for the SPH estimate of a function is given in Di et al. [11]. Due to the complexity and high chance for wrong interpretation, we refer to Di et al. [11] for the expression and derivation of the error bound. Furthermore Di et al. [11] concluded that the particle approximation converges to the kernel approximation independently of h, when the following scaling is satisfied[18]

$$\mathcal{N} \propto h^{n-\beta}, \tag{46}$$

where n is the spatial dimension, $\mathcal{N}$ is number of kernel particles and $\beta > n$.

[16] Provided that C^0 and C^1 consistency is already satisfied. Furthermore the requirement is derived in the 1D case, but can be extended to higher dimensions.

[17] This follows from a linear stability analysis performed by Dehnen and Aly [9]. Dehnen and Aly [9] conclude that smoothing kernels whose Fourier transform is negative will inevitably trigger pairing instability for large enough $\mathcal{N}$. Wendland kernels have been constructed with the condition that they need to posses a non-negative Fourier transform.

[18] When $\mathcal{N} \to \infty$, we need to have that the particle mass scales with h as $m \propto h^{\beta}$, with $\beta < n$. This results in the requirement that $m \to 0$ as $h \to 0$, which leads to the scaling $\mathcal{N} \propto h^{n-\beta}$[11].

3.6 Kernel type

Throughout this thesis, we use a Wendland C4 kernel [17]:

$$W\left(\mathbf{r}-\mathbf{r}', h\right)=\frac{\alpha_d}{h^d} K\left(\frac{\mathbf{r}-\mathbf{r}'}{h}\right), \tag{47}$$

with $\alpha_d = \frac{9}{21}, \frac{3\pi}{4}$ and $\frac{165\pi}{256}$ and where d stands for the different spatial dimensions, namely $d = 1, 2$ and 3. Furthermore, K is defined by

$$K(\xi)=\left\{\begin{array}{ll} \left(1-\left|\frac{\xi}{2}\right|\right)^6\left(\frac{35}{4}\xi^2+9|\xi|+3\right) & 0 \leq |\xi| \leq 2 \\ 0 & |\xi|>2. \end{array}\right. \tag{48}$$

The main reason for choosing a Wendland-type kernel is the earlier mentioned immunity to particle clumping. Furthermore, from a comparison with different kernels (Cubic spline, Gauss and Quintic kernels) it followed that the Wendland-type kernel has the best ”ability” to obtain the highest accuracy [28].

4 Rheology and surface tension in SPH

4.1 Tuning of rheological fluid properties

The main objective of this thesis is to be able to tune the surface tension independently of the viscoelastic fluid properties. In previous studies [21] it has been shown that particle interaction forces (PIF) can be used to tune rheological properties of the simulated fluid. Particle interactions introduce an additional force, that generates an attraction force for distant particles and repulsion for close particles. There are multiple ways to express and implement a PIF. An example of the total force acting on a particle is given by [21]

$$\boldsymbol{F}_i = \sum_j^{\mathcal{N}} F_{ij} \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|}, \tag{49}$$

where F_{ij} is

$$F_{ij} = \begin{cases} \frac{S_{ij}}{p}\left(-\left(r_{ij}-q\right)^2 + p\right),\ q = \frac{x_1^2 - x_2^2}{2x_1 - 2x_2},\ p = (x_2 - q)^2, & r_{ij} \le x_2 \\ 0, & r_{ij} > x_2. \end{cases} \tag{50}$$

The tunable parameters are the interaction strength (S_{ij}), the range of the particle interaction (x_2) and the ratio of repulsive/attractive force (x_1). The visual representation of equation 50 is given in Figure 5.

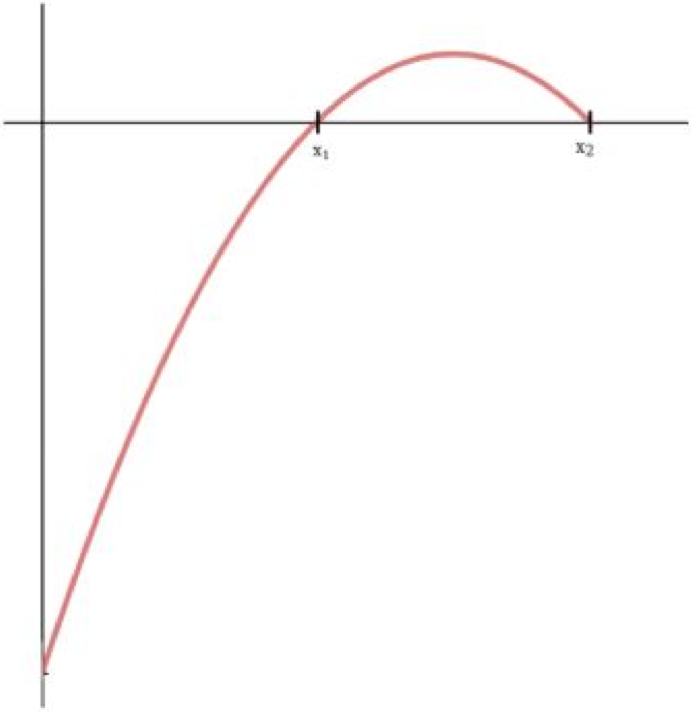

Figure 5: Visual representation of the particle interaction force given in equation 50, where on the horizontal axis we have $\mathbf{r}_{ij}$ and on the vertical axis we have F_{ij} [21].

We will now state the side effects due to the particle interaction force. However, since this is dependent on the final used particle interaction force, it will be written at the end of the thesis (the particle force *can change). However, the for now known side effect is an additional pressure due to a difference in the amount of repulsive and attractive forces.*

It has already been show in [3, 30] that the particle interaction force has an influence on the surface tension. Two methods are considered for further tuning of the surface tension, namely the Continuum Surface Force (CSF) method and a particle interaction surface minimization force (PISMF) method. Both methods are based on a force that minimises the surface area. We will first discus the CSF method after which the PISMF method will be explained.

4.2 Continuum surface force method

The CSF method translates the surface tension into a volumetric force. The force acts as a smoothing for high curvature regions in order to minimize the surface area. The volumetric force is given by [24]

$$\mathbf{F}_s = \sigma\kappa\hat{\mathbf{n}}\delta_s, \tag{51}$$

where σ is the surface tension coefficient, $\hat{\mathbf{n}}$ is the unit surface normal, κ is the curvature and δ_s is the surface delta function.

The drawbacks of the CSF method are that the accuracy depends on the normal vector calculation and curvature estimation, where the later is prone to errors. Most common methods use a colour function to calculate the normal and the curvature, where the colour function identifies each fluid [5][19]. To resolve the error prone calculations of the curvature, many corrections were introduced, each with its own pros and cons. In Adami and Account [2] a comparison of the most popular correction methods is presented. From the comparison, it followed that the classical method developed by Sirotkin and Yoh [29] is the most stable one. They used a correction matrix to solve the problem of the decreased number of interpolation points near the free surface for one-fluid phases, resulting in more accurate results. Furthermore, a transition band (multiple layers of surface particles) for the calculation of the curvature is introduced. Other methods, such as presented by [1] and [24] tend to underestimate the curvature near the surface. However, stability still remains to be an issue, since it will depend on many other factors such as the pressure force, order of viscosity force and the density calculation near the free surface.

Despite all the corrections, the methods are still depending on the amount of particles; the methods become less reliable when handling thin liquid layers with few particles. He et al. [13] developed another method that is able to handle thin film features. The difference between the methods is that He et al. [13] uses $|\nabla c|^2$ instead of ∇c, with the advantage that the normal direction is no longer needed, which causes high errors in thin film sheets. This turn out to improve robustness against particle sparsity.

In a promising new study, Duan, Sun, and Jiang [12] formulated a method based on improving the following two drawbacks of the CSF method: (i) curvature calculations may be subjected to large and nonphysical errors, (ii) instability with the increase of the computational resolution. The main improvement is the different way of calculating the curvature[20]. The curvature is calculated by the surface divergence instead of the divergence of the unit normal[21], that is

$$\kappa = -\nabla_S \cdot \mathbf{n}. \tag{54}$$

[19] The color function in single-phase is equal to one for all fluid particles. Often the following smoothing of the color function is used:

$$c_i = \sum_j c_j V_j W_{ij}, \tag{52}$$

where c_j is the color function [12].

[20] Also a different method for tracking the surface particles is implemented, namely the Marrone algorithm. For more details see Duan, Sun, and Jiang [12]

[21] Normally the curvature is calculated by

$$\kappa = -\nabla \cdot \mathbf{n} \tag{53}$$

According to [5] the gradient tangent to the interface can be expressed as

$$\nabla_S = \nabla - \nabla_N. \tag{55}$$

In a theoretical case $\nabla_N \cdot \mathbf{n}$ is equal to zero, since all the interior particles are aligned with normal $\mathbf{n}$ on the surface (see Figure 6), hence

$$\nabla \cdot \mathbf{n} = \nabla_S \cdot \mathbf{n}. \tag{56}$$

However, due to numerical discretization this is no longer the case. The particles are no longer perfectly aligned with normal $\mathbf{n}$, resulting in that $\nabla \cdot \mathbf{n}$ is no longer zero and hence

$$\nabla \cdot \mathbf{n} \neq \nabla_S \cdot \mathbf{n}. \tag{57}$$

When in the numerical case $\nabla \cdot \mathbf{n}$ is used, it will results in computational errors.

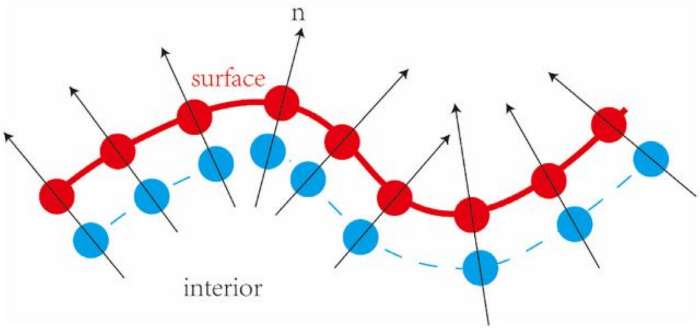

Figure 6: The theoretical situation in which the derivatives are equal. However, due to numerical discretizations the particles will no longer arrange in the same way [12].

4.3 Particle interaction surface minimization

The PISMF method introduces an additional force, that counteracts the surface curvature. The first step of the PISMF method is to compute the normal vector based on the gradient of the density field [3, 15]

$$\mathbf{n}_i = h \sum_{j=1}^{\mathcal{N}} \frac{m_j}{\rho_j} \nabla_i W(|\mathbf{r}_{ij}|), \tag{58}$$

where h is the smoothing length. To give an expression for the minimization force, we use the assumption that the magnitude of $\mathbf{n}_i$ is proportional to the curvature. The minimization force is then given by

$$\mathbf{F}_i^{curv} = -\gamma m_i \sum_{j=1}^{\mathcal{N}} (\mathbf{n}_i - \mathbf{n}_j), \tag{59}$$

where γ is a coefficient to control the strength of the surface tension. It can be easily verified that $\mathbf{F}_i^{curv}$ increases, whenever the curvature increases. Furthermore, for flat regions and inside the fluid, the force is equal to zero. This is exactly what we wanted. The force should only influence the surface area and must act as a minimization. The particle surface force is then given by

$$\mathbf{F}_i^{surf} = K_i (\mathbf{F}_i + \mathbf{F}_i^{curv}), \tag{60}$$

where

$$K_i = \sum_{j=1}^{\mathcal{N}} \frac{2\rho_0}{\rho_i + \rho_j} \tag{61}$$

is a symmetrized correction factor, with ρ_0 the rest density of the fluid[22]. The advantages of the PISMF method are:

- Avoids explicit computation of the surface curvature. Hence, improving sensitivity to particle disorder.
- Avoids normalization of $\mathbf{n}_i$.
- Symmetric force.

Yang et al. [33] improved the accuracy by introducing an anisotropic filtering term [23]

$$\mathbf{F}_i := \mathbf{T}_i \mathbf{F}_i, \tag{62}$$

where $\mathbf{T}_i$ is the anisotropic filtering term[24]. The anisotropic filtering term scales the particle interaction force. In an isotropic case $\mathbf{T}_i$ is equal to the identity matrix, however in the anisotropic case $\mathbf{T}_i$ scales the interaction forces of each particle differently [33], see Figure 7.

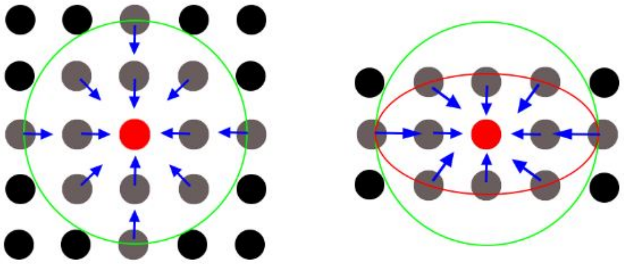

Figure 7: Illustration of how the anisotropic filtering works. Left: isotropic case, resulting in equal scaling factors (blue arrows). Right: anisotropic case, resulting in different scaling factors. Green boundary represents the influence domain of the (considered) red particle [33]

.

[22] Question: since we assume with PPE that the density is not allowed to change is ρ_i not equal to ρ_j

[23] This method omitted $\mathbf{F}_i^{curv}$ and only used $\mathbf{F}_i$.

[24] For a detailed mathematical description, explicit expression of the filtering term and a derivation see [33].

5 Numerical experiments

In this section, we will first describe the SPH simulation in general. Secondly, we will state two ways that will be used to measure the surface tension, where the first one is valid for both Newtonian and non-Newtonian fluids and the second one is only valid for Newtonian fluids. Lastly, we will explain the SAOS method.

5.1 Simulations details

The desired fluid will be simulated in a box (partially) filled with fluid particles. The particles will be initialized on a grid, where the particle placement is defined in terms of the (initial) particle diameter L_0. The particle diameter is defined by

$$L_0 = \frac{L_x}{N_{particles}}, \tag{63}$$

where L_x is the length of the box in the x-direction (meters) and $N_{particles}$ is the maximum possible number of particles in the x-direction[25], see Figure 8. The particles can now be placed on the grid by expressing their coordinates x, y and z in terms of L_0. Important to note is that when $L_x = L_y$ and the maximum number of particles in the y-direction is larger than in the x-direction, the particles will cram/merge together.

The boundaries of the box are periodic, i.e. the box is extended in every direction through attaching a copy of the whole box at every border, see Figure 9. Due to the period box, two restrictions are needed to prevent computational errors from occurring, these restrictions are

- The cut-off distance should be smaller than the dimensions of the box divided by two.
- When a particle leaves the box at a certain boundary, the particle should enter the box at the opposite boundary.

The first restriction follows from that due to the periodic box, particles can interact not only with the central box, but also with the ”copied” particles. To avoid errors, a single particle should only interact with the original neighboring particle or only with a single copy. Resulting in the restriction on the cut-off distance, see Figure 9. The second restriction is more a definition of periodic boundaries, then it is a restriction.

[25] L_0 is solemnly based on the x-direction.

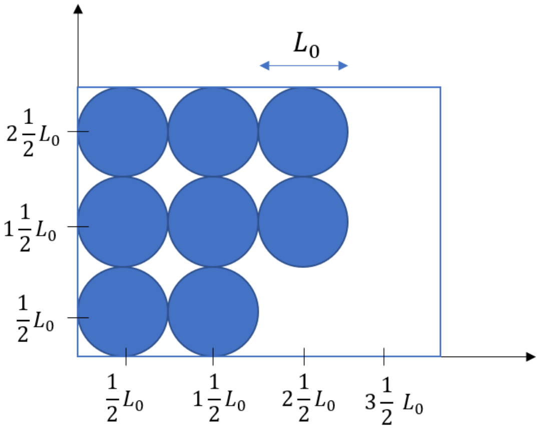

Figure 8: Visualisation of initial particle placement in 2D. When we take $L_x = 2\ m$, we have that $L_0 = \frac{1}{2}\ m$, since the maximum number of particles is equal to 4.

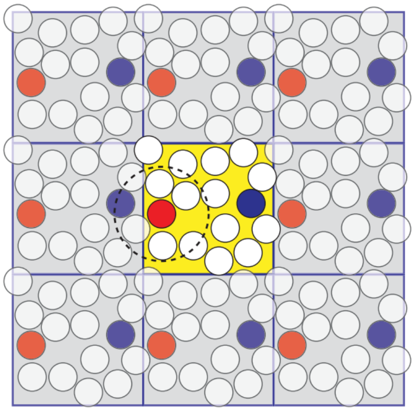

Figure 9: Example of a 2D periodic box, where the dashed line indicates the cut-off radius of the red particle and the yellow box represents the central box. Observe, that if the cut-off distance is increased, the red particle will interact with multiple blue particles.[26]

5.2 Surface tension

The first method involves simulating a stationary droplet. The surface tension force acts as a minimization of the droplet surface, which increases the droplets pressure. When the droplet is in a stationary phase, the surface tension is in equilibrium with the pressure gradient at the interface of the droplet. Using the Young-Laplace equation, this balance is expressed as [20]

$$P_{in} - P_{out} = \frac{\sigma}{R_e}, \tag{64}$$

where R_e is the equilibrium radius of the droplet and P_{in} and P_{out} are the pressure inside and outside the droplet. The numerical method can now be verified by computing the difference between the calculated and the expected surface tension. This method is also used to determine the PIF influence on the surface tension.

The second method simulates an oscillating Newtonian droplet with small-amplitudes and low viscosity. In the case that the surface tension is the driving force, the theoretical period is given by [20]

$$T = 2\pi R_e \sqrt{\frac{R_e\left(\rho_d + \rho_l\right)}{6\sigma}}, \tag{65}$$

where ρ_d is the droplets density and ρ_l is the surrounding density. Since the simulations involve periodic boundaries, it is important that the particles do not interact with the boundaries of the box. To guarantee this, we make the dimensions of the central box many times larger than the dimensions of the fluid, see Figure 10. This will have no affect on the computational time, since except from the fluid particles the box will be empty.

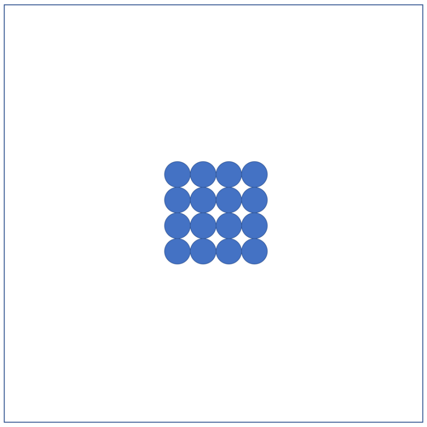

Figure 10: 2D visualization of the initialization particle placement for the droplet simulations.

5.3 SAOS: Small Amplitude Oscillatory Shear

Making use of the in section 2.2 explained viscous and elastic stress-strain responses, we can determine the viscoelastic fluid behaviour through SAOS. The goal of SAOS is to obtain the characterizing viscoelastic properties, which are the storage and loss modulus. To obtain the loss and storage modulus, a body force $\mathbf{F}_{sin}$ is applied to a periodic box. The body force results in a delayed velocity field, which is used to determine the loss and storage modules. The imposed body force only acts in the x-direction, given by [21]

$$F_{sin} = Asin(ly)sin(\omega t), \tag{66}$$

where l is the wavelength, A is the amplitude (determines the strength) and ω is the frequency. The visualisation of the body force is given by Figure 11. The body force has the following properties

- F_{sin} is a sinusoidal force.
- The wave period equals the box length (in the y-direction), that is $l = \frac{2\pi}{l_{boxwidth}}$.
- $\mathbf{F}_{sin}$ only acts in the x-direction.
- $sin(\omega t)$ changes the amplitude of the standing wave $Asin(ly)$ in time.

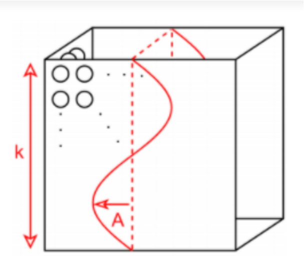

Figure 11: A schematic overview of the body force. The force is a standing wave, where the oscillations are due to the sinusoidal time force. Furthermore, the force at time t is constant in the x-direction and hence varies only in the y-direction[21].

From Linde [21], it follows that the storage and loss modulus can be expressed in terms of the force amplitude A, force frequency ω, wave length l and the velocity amplitudes v_0 and w_0[26]. That is, G' and G'' can be expressed as [21]

$$G' = \frac{\rho\omega^2}{l^2} + \frac{A\omega}{l^2 v_0 \left(\frac{\omega_0}{v_0} + \frac{v_0}{\omega_0}\right)} \tag{68}$$

$$G'' = \frac{A\omega}{l^2 \omega_0 \left(\frac{\omega_0}{v_0} + \frac{v_0}{\omega_0}\right)}. \tag{69}$$

The only unknowns are v_0 and ω_0, however these can be determined by [21]

$$\frac{1}{4} v_0 \approx \frac{1}{M} \sum_{j=1}^{M} \frac{1}{N} \sum_{i=1}^{N} \left(v_{x,i} \sin\left(l y_i\right)\right) \cdot \sin\left(\omega t_j\right) \tag{70}$$

$$\frac{1}{4} w_0 \approx \frac{1}{M} \sum_{j=1}^{M} \frac{1}{N} \sum_{i=1}^{N} \left(v_{x,i} \sin\left(l y_i\right)\right) \cdot \cos\left(\omega t_j\right), \tag{71}$$

where M is the number of time steps and N is the number of particles. In short, expressions (70) and (71) are derived by applying a Fourier transform to the measured velocity field, in order to filter out the sine and cosine responses [21]. Linde [21] found that to avoid inaccuracies from occurring, due to startup effects, the first few oscillations are to be truncated. Another important observation is that for Newtonian fluids, the loss modulus should be equal to the viscosity multiplied with the frequency, i.e. [21]

$$G'' = \mu\omega. \tag{72}$$

Expression (72) will be used to determine if PIF has the ability to simulate Newtonian fluids. Lastly, SAOS can only by used for small values of A, since the above expressions only hold in the linear viscoelastic regime.

[26] The body force results in a delayed velocity profile, given by

$$v_x = sin(ly)\left[v_0 sin(\omega t) + \omega_0 cos(\omega t)\right]. \tag{67}$$

6 Research question

In this final chapter, we conclude the literature study with stating the main research question and the corresponding sub questions.
Main research question

- How can pairwise interaction forces be used to tune the fluid rheology and surface tension independently?

Sub-question

- What are advantage and disadvantage of SPH and what is SPH?
- What is the influence of using different discrete expressions for the pressure force?
- Which methods can be used to tune the surface tension?
- What is the effect of particle interaction forces on the surface tension?
- How can the surface tension be measured?
- Which (analytical) methods can be used to verify the numerical implementations?
  - For the Newtonian case.
  - For the viscoelastic case.
- What numerical and/or experimental results can be used to test/compare the implementations?

References

[1] S. Adami, X. Y. Hu, and N. A. Adams. "A new surface-tension formulation for multi-phase SPH using a reproducing divergence approximation". In: *Journal of Computational Physics* 229.13 (2010), pp. 5011–5021. ISSN: 10902716. DOI: 10.1016/j.jcp.2010.03.022. URL: http://dx.doi.org/10.1016/j.jcp.2010.03.022.

[2] S Adami and J Account. "Comparison of interface models to account for surface tension in SPH method". In: (2020), pp. 714–725.

[3] Nadir Akinci, Gizem Akinci, and Matthias Teschner. "Versatile surface tension and adhesion for SPH fluids". In: *ACM Transactions on Graphics* 32.6 (2013). ISSN: 07300301. DOI: 10.1145/2508363.2508395.

[4] Arne Bøckmann, Olga Shipilova, and Geir Skeie. "Incompressible SPH for free surface flows". In: *Computers and Fluids* 67 (2012), pp. 138–151. ISSN: 00457930. DOI: 10.1016/j.compfluid.2012.07.007.

[5] J. U. BRACKBILL, D. B. KOTHE, and C. ZEMACH. "A Continuum for Modeling Surface Tension *". In: *Journal of Computational Physics* 335354 (1992).

[6] Shuai Chen and Wei Niu. "On different calculation formulas of the pressure term in bi-phase SPH simulations". In: *AIP Advances* 8.10 (2018). ISSN: 21583226. DOI: 10.1063/1.5052504. URL: http://dx.doi.org/10.1063/1.5052504.

[7] Alex D Chow et al. "Incompressible SPH (ISPH) with fast Poisson solver on a GPU". In: *Computer Physics Communications* 226 (2018), pp. 81–103. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2018.01.005. URL: https://doi.org/10.1016/j.cpc.2018.01.005.

[8] Paul W. Cleary and Joseph J. Monaghan. "Conduction Modelling Using Smoothed Particle Hydrodynamics". In: *Journal of Computational Physics* 148.1 (1999), pp. 227–264. ISSN: 00219991. DOI: 10.1006/jcph.1998.6118.

[9] Walter Dehnen and Hossam Aly. "Improving convergence in smoothed particle hydrodynamics simulations without pairing instability". In: *Monthly Notices of the Royal Astronomical Society* 425.2 (2012), pp. 1068–1082. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2012.21439.x. arXiv: 1204.2471.

[10] Leonardo Di G. Sigalotti et al. "On the kernel and particle consistency in smoothed particle hydrodynamics". In: *Applied Numerical Mathematics* 108 (2016), pp. 242–255. ISSN: 01689274. DOI: 10.1016/j.apnum.2016.05.007. arXiv: 1605.05245.

[11] Leonardo Di et al. "A new insight into the consistency of the SPH interpolation formula". In: *Applied Mathematics and Computation* 356 (2019), pp. 50–73. ISSN: 0096-3003. DOI: 10.1016/j.amc.2019.03.018. URL: https://doi.org/10.1016/j.amc.2019.03.018.

[12] Riqiang Duan, Chen Sun, and Shengyao Jiang. "A new surface tension formulation for particle methods". In: *International Journal of Multiphase Flow* 124 (2020), p. 103187. ISSN: 03019322. DOI: 10.1016/j.ijmultiphaseflow.2019.103187. URL: https://doi.org/10.1016/j.ijmultiphaseflow.2019.103187.

[13] Xiaowei He et al. "Robust Simulation of Small-Scale Thin Features in SPH-based Free Surface Flows". In: *Life.Kunzhou.Net* 1.212 (2014), pp. 1–8. ISSN: 07300301. DOI: 10.1145/XXXXXXX.YYYYYYY. arXiv: 1006.4903. URL: http://life.kunzhou.net/2013/SPHsurfacetension.pdf.

[14] Manuel Hirschler et al. "Modeling of droplet collisions using Smoothed Particle Hydrodynamics". In: *International Journal of Multiphase Flow* 95 (2017), pp. 175–187. ISSN: 03019322. DOI: 10.1016/j.ijmultiphaseflow.2017.06.002. URL: http://dx.doi.org/10.1016/j.ijmultiphaseflow.2017.06.002.

[15] Markus Huber et al. "Evaluation of Surface Tension Models for SPH-Based Fluid Animations Using a Benchmark Test". In: *VRIPHYS* (2015).

[16] Malvern Instruments. "A basic introduction to rheology". In: *Malvern Instruments Limited* (2016).

[17] Yongsong Jiang et al. "A second-order numerical method for elliptic equations with singular sources using local filter". In: *Chinese Journal of Aeronautics* 26.6 (2013), pp. 1398–1408. ISSN: 10009361. DOI: 10.1016/j.cja.2013.07.004. URL: http://dx.doi.org/10.1016/j.cja.2013.07.004.

[18] P Kunz et al. "Study of Multi-phase Flow in Porous Media : Comparison of SPH Simulations with Micro-model Experiments". In: *Transport in Porous Media* 114.2 (2016), pp. 581–600. ISSN: 1573-1634. DOI: 10.1007/s11242-015-0599-1.

[19] Labrotovap. *How does spray dryer work*. 2020. URL: https://www.labrotovap.com/how-does-spray-dryer-work/ (visited on 10/09/2020).

[20] Yixin Lin, G R Liu, and Guangyu Wang. "A particle-based free surface detection method and its application to the surface tension effects simulation in smoothed particle hydrodynamics (SPH)". In: *Journal of Computational Physics* 383 (2019), pp. 196–206. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.12.036. URL: https://doi.org/10.1016/j.jcp.2018.12.036.

[21] Mark van der Linde. "Viscoelasticity in SPH through particle interactions". PhD thesis. TU Delft, 2020.

[22] M B Liu G R Liu. *Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments*. 2010, pp. 25–76. ISBN: 1183101090. DOI: 10.1007/s11831-010-9040-7.

[23] J. J. Monaghan. "Smoothed particle hydrodynamics". In: *Reports on Progress in Physics* 68.8 (2005), pp. 1703–1759. ISSN: 00344885. DOI: 10.1088/0034-4885/68/8/R01. arXiv: 0507472 [astro-ph].

[24] Joseph P Morris. "Simulating surface tension with smoothed particle hydrodynamics". In: *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS Int. J. Numer. Meth. Fluids 2000; 33: 333–353 Simulating* 6 (2000), pp. 333–353.

[25] Faith A. Morrison. *Understanding Rheology*. Oxford University Press, 2001. ISBN: 978-0-19-514166-5. URL: https://app.knovel.com/hotlink/toc/id:kpUR00000I/understanding-rheology/understanding-rheology.

[26] J.T. Padding. "Particle-Based Simulations Lecture Notes". In: (2017), pp. 109–132, 141–148.

[27] M S Shadloo, G Oger, and D Le Touzé. "Smoothed particle hydrodynamics method for fluid flows , towards industrial applications : Motivations , current state , and challenges". In: 136 (2016), pp. 11–34. DOI: 10.1016/j.compfluid.2016.05.029.

[28] Meng Shuangshuang et al. "The study on performances of kernel types in solid dynamic problems by smoothed particle hydrodynamics". In: *Computational Particle Mechanics* (2020). ISSN: 2196-4378. DOI: 10.1007/s40571-020-00339-0. URL: https://doi.org/10.1007/s40571-020-00339-0.

[29] Fedir V. Sirotkin and Jack J. Yoh. "A new particle method for simulating breakup of liquid jets". In: *Journal of Computational Physics* 231.4 (2012), pp. 1650–1674. ISSN: 10902716. DOI: 10.1016/j.jcp.2011.10.020. URL: http://dx.doi.org/10.1016/j.jcp.2011.10.020.

[30] A. M. Tartakovsky et al. "Smoothed particle hydrodynamics and its applications for multiphase flow and reactive transport in porous media". In: *Computational Geosciences* 20.4 (2016), pp. 807–834. ISSN: 15731499. DOI: 10.1007/s10596-015-9468-9.

[31] Jean Luc Thiffeault. "Covariant time derivatives for dynamical systems". In: *Journal of Physics A: Mathematical and General* 34.29 (2001), pp. 5875–5885. ISSN: 03054470. DOI: 10.1088/0305-4470/34/29/309. arXiv: 0102038 [nlin].

[32] Rui Xu, Peter Stansby, and Dominique Laurence. "Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach". In: *Journal of Computational Physics* 228.18 (2009), pp. 6703–6725. ISSN: 10902716. DOI: `10.1016/j.jcp.2009.05.032`. URL: `http://dx.doi.org/10.1016/j.jcp.2009.05.032`.

[33] Tao Yang et al. "Pairwise Force SPH Model for Real-Time Multi-Interaction Applications". In: X.X (2017), pp. 1–13.

[34] Wei Yu et al. "Theory for drop deformation in viscoelastic systems". In: *Journal of Rheology* 48.2 (2004), pp. 417–438. ISSN: 0148-6055. DOI: `10.1122/1.1647559`.

Appendix A: Introduction Tensor Calculus

In this section we will explain two different ways for constructing a basis in an Euclidean space, where it is no longer desired for the bases to be orthonormal (orthogonal and unit vectors). Furthermore, the transformation rules between the coordinate systems are given. Since the bases are no longer orthonormal, a standard partial derivative is no longer sufficient[27]. The covariant derivative is therefore introduced, after which a covariant time derivative is presented.

We will consider a three-dimensional Euclidean space, with a standard Cartesian coordinate system (x,y,z) associated with a set of unit vectors $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$. The purpose is to construct a basis that describes a vector at a point P. This will be done in two ways:

- Normals to coordinate surface (dual basis)
- Tangents to coordinate curves (natural basis).

Consider a non-Cartesian coordinates system (u,v,w), where we can express the Cartesian coordinates in terms of u,v,w and vice versa. The position vector **r**(u,v,w) expressed in non-Cartesian coordinates is given by

$$\mathbf{r}(u,v,w) = x(u,v,w)\mathbf{i} + y(u,v,w)\mathbf{j} + z(u,v,w)\mathbf{k}. \tag{73}$$

The natural basis is defined as the three tangent vectors to the three coordinate curves. These are the partial derivatives of $\mathbf{r}$ with respect to u, v, w evaluated at point P[BRON]:

$$\mathbf{e_u} := \frac{\partial \mathbf{r}}{\partial u}, \mathbf{e_v} := \frac{\partial \mathbf{r}}{\partial v}, \mathbf{e_w} := \frac{\partial \mathbf{r}}{\partial w}. \tag{74}$$

In contrast to the Cartesian basis, the natural basis does not need to be orthonormal. For the dual basis, the non-Cartesian coordinates needs to be expressed in terms of x,y,z. Each coordinate can now be viewed as a scalar field and such, the dual basis is given by taking the gradients of the coordinates:

$$\mathbf{e}^u := \nabla u,\ \mathbf{e}^v := \nabla v,\ \mathbf{e}^w := \nabla w, \tag{75}$$

where $\nabla u = \frac{\partial u}{\partial x}\mathbf{i} + \frac{\partial u}{\partial y}\mathbf{j} + \frac{\partial u}{\partial z}\mathbf{k}$. Observe that the dual basis and the natural basis are equivalent when the coordinate system (u,v,w) is orthonormal. Using the natural and dual basis we can express at point P the vector $\boldsymbol{\lambda}$ in two ways:

$$\begin{aligned} \boldsymbol{\lambda} &= \lambda^u \mathbf{e}_u + \lambda^v \mathbf{e}_v + \lambda^w \mathbf{e}_w \\ \boldsymbol{\lambda} &= \lambda_u \mathbf{e}^u + \lambda_v \mathbf{e}^v + \lambda_w \mathbf{e}^w \end{aligned} \tag{76}$$

To shorten such notation we introduce Einstein notation. In Einstein notation, we use $u^i (i = 1, 2, 3)$ as coordinates instead of (u, v, w), $\{\mathbf{e_i}\}$ and $\{\mathbf{e^i}\}$ in place of $\{\mathbf{e_u}, \mathbf{e_v}, \mathbf{e_w}\}$ and $\{\mathbf{e^u}, \mathbf{e^v}, \mathbf{e^w}\}$ and the same way for the vector components. Furthermore, when an index variable is repeated, it implies a summation over the values of the index. Hence, we can rewrite expression 76 as

$$\begin{aligned} \boldsymbol{\lambda} &= \sum_{i=1}^{3} \lambda^i \mathbf{e}_i = \lambda^i \mathbf{e}_i \\ \boldsymbol{\lambda} &= \sum_{i=1}^{3} \lambda_i \mathbf{e}^i = \lambda_i \mathbf{e}^i. \end{aligned} \tag{77}$$

[27] In curvature coordinates the basis will change, resulting in that the partial derivative of a tensor is no longer a tensor, the partial derivative is depending on its coordinate system. For a detailed example see [BRON will follow]

Coordinate transformation

The contravariant and covariant vector is defined by the transformation of the vector components between two coordinate systems. Given two arbitrary coordinates system X and $X^{'}$, with coordinates x^m and $x^{m^{'}}$, the transformation between two contravariant vector components V^m and $V^{m^{'}}$ is given by[28]

$$V^{m^{'}} = \frac{\partial x^{m^{'}}}{\partial x^m} V^m, \tag{79}$$

and the transformation between covariant vector components is given by

$$V_{m'} = \frac{\partial x^m}{\partial x^{m'}} V_m. \tag{80}$$

We can easily extend it for higher order tensors. A second order tensor component is defined as $T^{mn} = V^m U^n$ [29]. The transformation of T^{mn} can be deduced from the transformation (see equation (79)) of the individual components:

$$T^{m^{'}n^{'}} = V^{m^{'}} U^{n^{'}} = \frac{\partial x^{m^{'}}}{\partial x^m} V^m \frac{\partial x^{n^{'}}}{\partial x^n} U^n = \frac{\partial x^{m^{'}}}{\partial x^m} \frac{\partial x^{n^{'}}}{\partial x^n} T^{mn}. \tag{81}$$

Using the transformation of the individual components the transformation for even higher order contravariant tensors, covariant tensors or mixed tensors can be deduced.

Metric tensor

An intuitive way to see what a metric tensor is and does, is by considering the length of a vector:

$$\boldsymbol{\lambda} \cdot \boldsymbol{\lambda} = \lambda^j \mathbf{e}_j \cdot \lambda^i \mathbf{e}_i = \lambda^j \lambda^i \left(\mathbf{e}_j \cdot \mathbf{e}_i \right), \tag{82}$$

where the term in the brackets is the metric tensor g_{ij}. The metric tensor specifies how to calculate the length between vectors. We can now also define the length of a differential displacement vector as

$$ds^2 = g_{ij} dx^i dx^j, \tag{83}$$

where dx^i are the differential displacements components. Furthermore, we have [BRON]

$$g_{ij} g^{jk} = \delta_i^k, \tag{84}$$

resulting in the identity matrix. This is a useful property to lower or raise indices on tensor components.

Covariant derivative

A fundamental property of a tensor is that it will not change under a coordinate transformation, i.e it is independant of the manner in which it is expressed in a coordinate system. However, the partial derivative of a tensor will not result in a tensor and hence losses its independence. To solve this, we introduce a new derivative operator, namely a covariant derivative [BRON].

[28] This can be deduced by considering

$$\mathbf{e}_j := \frac{\partial \mathbf{r}}{\partial u^j} = \frac{\partial \mathbf{r}}{\partial u^{i'}} \frac{\partial u^{i^{'}}}{\partial u^j} = \frac{\partial u^{i^{'}}}{\partial u^j} \mathbf{e}_{i'}, \tag{78}$$

in combination with equation (77).

[29] Note, that the correct notation is $T^{mn} = V^m \otimes V^n$

We can derive the covariant derivative operator by considering the following expression [BRON]:

$$\frac{\partial \boldsymbol{\lambda}}{\partial u^{j'}} = \frac{\partial(\lambda^i \mathbf{e}_i)}{\partial u_{j'}} = \frac{\partial \lambda^i}{\partial u_{j'}}\mathbf{e}_i + \lambda^i \frac{\partial \mathbf{e}_i}{\partial u_{j'}}, \tag{85}$$

where the second term in the derivative is the Christoffel symbol, hence we can rewrite it to

$$\frac{\partial \boldsymbol{\lambda}}{\partial u_{j'}} = \frac{\partial \lambda^i}{\partial u_{j'}}\mathbf{e}_i + \lambda^i \Gamma^k_{ij'}\mathbf{e}_k. \tag{86}$$

In order to factor out $\mathbf{e}_i$, we will rename the dummy variables in the Christoffel symbol, yielding

$$\frac{\partial \boldsymbol{\lambda}}{\partial u_{j'}} = \left(\frac{\partial \lambda^i}{\partial u_{j'}} + \lambda_k \Gamma^i_{kj'}\right)\mathbf{e}_i. \tag{87}$$

The term in the brackets are the components of a tensor with respect to a covariant basis, hence the term in the brackets is a tensor [BRON]. Here, we used that a derivative of an invariant tensor ($\boldsymbol{\lambda}$) is a tensor. The covariant derivative of V^i is given by

$$\nabla_{j'} V^i = \frac{\partial V^i}{\partial j'} + \Gamma^i_{kj'} V^k, \tag{88}$$

with the desired property, that a derivative of a tensor results in a tensor. Furthermore, the covariant derivative of a covariant vector is almost identical, except the sign in front of the Christoffel symbol is negative[30] [BRON]:

$$\nabla_{j'} V_i = \frac{\partial V_i}{\partial j'} - \Gamma^k_{j'i} V_k \tag{89}$$

From the covariant derivative we can see the influence of non-Cartesian coordinates (curvature coordinates). The second term relates to how the bases will change due to the curvature of space. When expressed in Cartesian coordinates the second term vanishes, resulting in that the covariant derivative equals the partial derivative. Furthermore, the covariant derivative follows the same rules as a partial derivative, such as the product rule and chain rule.

Covariant time derivative

The time derivative for a vector field X (with arbitrary coordinates z^b) is defined as

$$\frac{dX^a(t, z(t))}{dt} := \frac{X^a}{\partial t} + \frac{X^a}{\partial z^b}\frac{\partial z^b}{\partial t} = \frac{X^a}{\partial t} + \frac{X^a}{\partial z^b} v^b, \tag{90}$$

where $z^b(t)$ represent the general time dependent coordinates, which is not covariant[31]. The covariant time derivative $\mathcal{D}$ is defined by [31]

$$\mathcal{D} X^a := \frac{dX^a}{dt} + \alpha^a{}_b X^b \tag{92}$$

[30] This is derived by expanding $\nabla_k(A^i B_i)$ with the product rule and using the above derived covariant derivative. Then noting that $A^i B_i$ is a scalar and the value of a scalar does not depend on its basis vectors. Hence, the covariant derivative is equal to an ordinary partial derivative. Finally, the minus sign follows from comparing the two derived expressions.

[31] The transformation law for equation (90) is given by

$$v^a = \frac{\partial z^a}{\partial x^k} v^k + \frac{\partial z^a}{\partial t}, \tag{91}$$

from where we can observe that the last term prevents v from transforming as a tensor [31].

, where the α^a_b are time-dependent correction quantities to make $\mathcal{D}$ obey the tensorial transformation laws. For a first-order contravariant tensor component the tensor transformation is

$$\mathcal{D}X^a = \frac{\partial z^a}{\partial z^{a'}}\mathcal{D}X^{a^{'}}. \tag{93}$$

To deduce the rule for covariant vectors components we desire two additional properties:

- Covariant time derivative reduces to the ordinary derivative (equation (90)) when acting on scalars.
- Must obey Leibniz (product) rule.

Consider a scalar product of a co- and a contravariant vector: $S = X^aY_a$. Using the Leibniz rule, we can therefore write

$$\begin{aligned} \mathcal{D}(X^aY_a) &= \mathcal{D}(X^a)Y_a + X^a\mathcal{D}(Y_a) \\ &= \left(\frac{dX^a}{dt} + \alpha^a{}_bX^b\right)Y_a + X^a\left(\frac{dY_a}{dt} + \tilde{\alpha}^b{}_aY_b\right). \end{aligned} \tag{94}$$

Now using that X^aY_a is a scalar and hence reduces to an ordinary derivative, we have

$$\begin{aligned} \mathcal{D}(X^aY_a) &= \frac{d\left(X^aY_a\right)}{dt} \\ &= \frac{d\left(X^a\right)}{dt}Y_a + X^a\frac{d\left(Y_a\right)}{dt}. \end{aligned} \tag{95}$$

For both expressions 94 and 95 to be true[32], the following needs to hold

$$\tilde{\alpha}^b{}_a = -\alpha^b{}_a. \tag{96}$$

The covariant time derivative acting on a covariant component is hence defined by

$$\mathcal{D}Y_a = \frac{dY_a}{dt} - \alpha^b{}_aY_b. \tag{97}$$

The $\alpha^b{}_a$ is used to enforce tensor transformation of $\mathcal{D}$ between general coordinates. We can derive an general expression for $\alpha^a{}_b$ by considering how $\alpha^a{}_b$ transform between coordinates [31]:

$$\begin{aligned} \mathcal{D}X^a &= \mathcal{D}\left(\frac{\partial z^a}{\partial z^{a'}}\mathcal{D}X^{a^{'}}\right) \\ &= \frac{d}{dt}\left(\frac{\partial z^a}{\partial z^{a'}}\right)X^{a^{'}} + \frac{\partial z^a}{\partial z^{a'}}\frac{dX^{a^{'}}}{dt} + \frac{\partial z^b}{\partial z^{a'}}X^{a^{'}}\alpha^a{}_b. \qquad ([33]) \end{aligned} \tag{98}$$

Hence, by combining expressions (93) and (98) and thus enforcing tensorial transformation, α is required to transform as

$$\alpha^{a^{'}}{}_{b'} = \frac{\partial z^{a^{'}}}{\partial z^a}\frac{\partial z^b}{\partial z^{b'}}\alpha^a{}_b + \frac{\partial z^{a^{'}}}{\partial z^c}\frac{d}{dt}\left(\frac{\partial z^c}{\partial z^{b'}}\right), \tag{99}$$

[32] For the scalar expression to hold, we need to have $\alpha^a{}_bX^bY_a + X^a\tilde{\alpha}^b{}_aY_b = 0$. Now rearranging dummy indices and terms, we have $\alpha^b{}_aX^aY_b + \tilde{\alpha}^b{}_aX^aY_b = 0$.

[33] The last term follows from forcing the desired term $X^{a^{'}}\alpha^a{}_b$ to appear. However, this should in the end only depend on the variable "a", so the dummy variables b and $a^{'}$ needs to be summed out (contraction). This is done by following the contracting rules: an upper index will be summed out by a lower index and vice versa. Hence, the term $\frac{\partial z^b}{\partial z^{a'}}$, resulting that all dummy variables are summed out.

where the first term is the usual tensorial transformation law and the second term prevents α from being a tensor. In [31] an expression for time evolution of the Jacobian matrix between two arbitrary time-dependent transformation is derived:

$$\frac{d}{dt}\left(\frac{\partial z^a}{\partial z^{a'}}\right) = \frac{\partial v^a}{\partial z^b}\frac{\partial z^b}{\partial z^{a'}} - \frac{\partial z^a}{\partial z^{b'}}\frac{\partial v^{b'}}{\partial z^{a'}}. \tag{100}$$

Now substituting the time-dependent transformation expression into equation 99 yields

$$\alpha^{a'}{}_{b'} + \frac{\partial v^{a'}}{\partial z^{b'}} = \frac{\partial z^{a'}}{\partial z^a}\frac{\partial z^b}{\partial z^{b'}}\left(\alpha^a{}_b + \frac{\partial v^a}{\partial z^b}\right). \tag{101}$$

Observe that the term in brackets transforms like a tensor. Hence, we can define the following expression for α up to an arbitrary tensor ($\mathcal{H}^a{}_b$), that is

$$\alpha^a{}_b = -\frac{\partial v^a}{\partial z^b} + \mathcal{H}^a{}_b. \tag{102}$$

If we now take $\mathcal{H}$ to be zero, we end up with the upper convected time derivative

$$\mathcal{D}X^a = \frac{\partial X^a}{\partial t} + \frac{\partial X^a}{\partial z^b}v^b - \frac{\partial v^b}{\partial z^b}X^b. \tag{103}$$