Unified Pressure, Surface Tension and Friction for SPH Fluids

TIMO PROBST, University of Freiburg, Germany
MATTHIAS TESCHNER, University of Freiburg, Germany

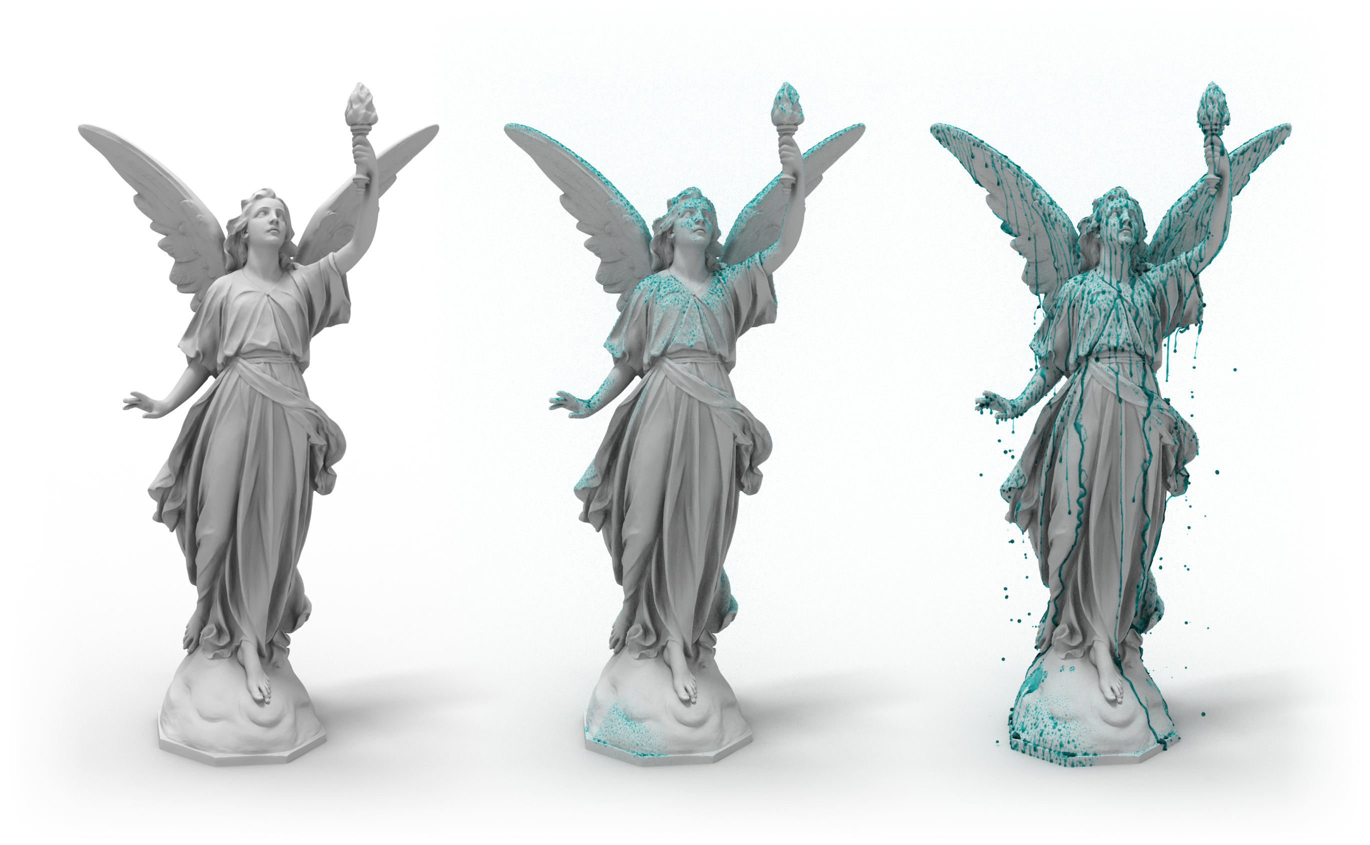


Fig. 1. Rain on the Lucy statue. At first, our friction and surface tension formulation together allow droplets to stick to the statue. Only after the droplets grow larger they start rolling downwards, leaving wet trails behind them. The trails mark preferred paths downwards for other droplets due to less frictional resistance, causing the formation of small fluid streams. Over ten million fluid particles of size 1 mm were used to simulate the rain water. The Lucy statue stems from the Stanford 3D Scanning Repository, we use a version modified by Mellla.

Fluid droplets behave significantly different from larger fluid bodies. At smaller scales, surface tension and friction between fluids and the boundary play an essential role and are even able to counteract gravitational forces. There are quite a few existing approaches that model surface tension forces within an SPH environment. However, as often as not, physical correctness and simulation stability are still major concerns with many surface tension formulations. We propose a new approach to compute surface tension that is both robust and produces the right amount of surface tension.

Conversely, less attention was given to friction forces at the fluid-boundary interface. Recent experimental research indicates that Coulomb friction can be used to describe the behavior of droplets resting on a slope. Motivated by this, we develop a novel friction force formulation at the fluid-boundary interface following the Coulomb model, which allows us to replicate a new range of well known fluid behavior such as the motion of rain droplets on a window pane.

Both forces are combined with an IISPH variant into one unified solver that is able to simultaneously compute strongly coupled surface tension, friction and pressure forces.

CCS Concepts: • **Computing methodologies → Physical simulation**.

Additional Key Words and Phrases: physically-based animation, smoothed particle hydrodynamics, droplets, surface tension, friction, pressure, implicit solver, strong coupling

ACM Reference Format:
Timo Probst and Matthias Teschner. 2024. Unified Pressure, Surface Tension and Friction for SPH Fluids. *ACM Trans. Graph.* 44, 1, Article 7 (December 2024), 28 pages. https://doi.org/10.1145/3708034

Authors' Contact Information: Timo Probst, probstt@cs.uni-freiburg.de, University of Freiburg, Freiburg, Baden-Württemberg, Germany; Matthias Teschner, University of Freiburg, Freiburg, Baden-Württemberg, Germany.

1 Introduction

Gravity, pressure, surface tension and frictional effects at the fluid-solid interface are the main driving forces acting on small fluid volumes such as droplets and water splashes. The overall objective of this work is to develop a simulation method that is able to replicate these small-scale fluid effects shown in fig. 1 within a *Smoothed Particle Hydrodynamics* (SPH) environment. While gravity is trivial to implement, in the past, the SPH community invested significant work in the development of pressure solvers such as PCISPH [Solenthaler and Pajarola 2009], IISPH [Ihmsen et al. 2014a], DFSPH [Bender and Koschier 2015] and others [Becker and Teschner 2007; Macklin and Müller 2013]. Also, a large selection of surface tension formulations have been proposed [Adami et al. 2010; Akinci et al. 2013; Becker and Teschner 2007; He et al. 2015b; Jeske et al. 2023; Morris 2000; Wang et al. 2017; Zorilla et al. 2020]. The implicit pressure solvers enable impressive water simulations with high fluid depth and scene complexity, while at the same time the underlying particle method remains easily extendible to other materials and forces [Bobzin et al. 2023; Gissler et al. 2020, 2019; Ihmsen et al. 2014b; Löschner et al. 2023; Mokrov et al. 2024; Peer et al. 2018]. Existing pairwise surface tension models typically focus on a robust and performant replication of expected surface tension effects, resulting in visually pleasing simulations of smaller scale fluid behavior. However, physical correctness of surface tension forces is often not discussed [Akinci et al. 2013; Becker and Teschner 2007; Duan et al. 2020; Jeske et al. 2023]. *Continuum surface force* (CSF) models [Brackbill et al. 1992] on the other hand offer a physically more consistent surface tension description, but there are serious concerns about conservation of momentum and simulation stability [Adami et al. 2010; Akinci et al. 2013; Becker and Teschner 2007; He et al. 2015b; Hyde et al. 2020]. With our new particle-based surface tension, we aim at developing a force formulation that is similarly robust and easy to implement as previous pairwise methods, while still being consistent with the underlying physical models.

Up to now, less attention was given to the frictional contact handling between SPH fluids and boundary. Here, friction at the solid-fluid interface is often implemented by simply incorporating boundary particles in the fluid viscosity computation [Bender et al. 2020; Müller et al. 2004; Peer et al. 2015; Peer and Teschner 2017; Weiler et al. 2018], with the downside that fluid droplets are not able to come to rest on an inclined surface. Recent research indicates that Coulomb friction can be used to model the tangential force component droplets experience when sticking to a solid surface [Gao et al. 2018; Hardt and McHale 2022; McHale et al. 2022]. Even though there exist works mentioning the use of a friction force for fluid particles that is inspired by Coulomb friction [Koschier and Bender 2017; Winchenbach et al. 2020; Zhang et al. 2012], we are not aware of an existing proper formulation of the Coulomb friction law for fluid particles at the fluid-solid interface.

Thus, in this paper, to model small scale fluid behavior, we focus on the following contributions:

- We develop a novel versatile SPH surface tension implementation that combines the robustness and intuitiveness of pairwise force models with the physical correctness of CSF models.
- We use a modified SPH formulation of a Coulomb friction force for the fluid-boundary interface that, in contrast to previous work, allows us to simulate fluid droplets truly sticking to inclined surfaces.
- We propose a single unified solving mechanism that combines our surface tension and friction computations with an incompressible SPH variant. The simultaneously solved pressure, surface tension and friction forces are all consistent with each other in the sense that during their computation process they already consider the effects of each other on the state of the fluid. We demonstrate that this unified solving procedure is superior compared to a separated approach, and further present a thorough evaluation of our solver's capabilities.

The rest of the paper is structured as follows: In section 2 we compare our solver and its individual force components to existing work in the literature. Section 3 gives an overview of the physical concepts behind pressure, surface tension and friction forces and their embedding into an SPH framework. Building on these concepts, in section 4 we translate the forces into an implicit formulation and describe how we can solve them in a unified and consistent manner. Section 5 discusses remaining implementation details and optimal parameter values. Finally, section 6 presents and discusses simulation experiments, which are summarized in sections 7 and 8.

2 Related Work

Since the early beginning of SPH [Gingold and Monaghan 1977; Lucy 1977] and its introduction to the computer graphics community [Desbrun and Gascuel 1996], many publications focused on expanding the range of effects that can be simulated with an SPH fluid [Gissler et al. 2020, 2019; Ihmsen et al. 2014b; Koschier et al. 2022; Macklin et al. 2014]. Improvements in the pressure computation enabled the simulation of incompressible fluids [Becker and Teschner 2007; Bender and Koschier 2015; Ihmsen et al. 2014a; Macklin and Müller 2013; Solenthaler and Pajarola 2009], highly viscous materials are modelled utilizing implicit solving mechanisms [Peer et al. 2015; Peer and Teschner 2017; Takahashi et al. 2015; Weiler et al. 2018] and even ferrofluids can be simulated within an SPH framework [Huang et al. 2019]. Multiphase approaches [Gissler et al. 2019; Ren et al. 2014; Solenthaler and Pajarola 2008] have shown that particle methods allow intuitive implementations of multiple interacting fluids and materials. Here, surface tension forces are often used to maintain separate fluid bodies with well-defined interfaces [Solenthaler and Pajarola 2008; Tartakovsky et al. 2016].

2.1 Surface Tension

Surface tension arises from an imbalance of attractive forces between molecules at the fluid surface [Cross and Plunkett 2014]. Although this simple description is not entirely uncontroversial as it fails to explain tension parallel to the surface [Berry 1971], there exist many surface tension models that imitate said microscopic effects with pairwise attractive forces between particles [Becker and Teschner 2007; Hong et al. 2008; Jeske et al. 2023; Nugent and Posch 2000; Tartakovsky and Meakin 2005; Yang et al. 2016, 2017]. Tartakovsky and Meakin [2005], Akinci et al. [2013] and Jeske et al. [2023] mitigate undesired particle clustering that is reported for some pairwise force models [Huber et al. 2015] by constructing specialized SPH kernel functions for the surface tension computation. Yang et al. [2016, 2017] argue that an insufficient number of particle neighbors at the surface causes inaccurate surface tension forces, and demonstrate that the surface smoothness can be improved by increasing kernel support size. Even though our force derivation is distinct from pairwise force models, in section 3.2 we can see that our surface tension formulation shares the same intuitive structure as pairwise models, while not relying on specialized kernel functions. Similar to Yang et al. [2016, 2017] we also use a slightly larger kernel support for the surface tension computation. Our motivation, however, is different from Yang et al. [2016, 2017] as discussed in section 5.1. In contrast to our surface tension formulation, pairwise force models are criticized for their inability to replicate the correct amount of surface tension depending on the curvature of the surface [Duan et al. 2020], which we also discuss in section 6.2. Nevertheless, they represent a class of robust, fast and easy to implement surface tension formulations that can replicate most of the expected visual effects of fluid surface tension.

2.1.1 *Continuum surface force models.*

Another approach to model surface tension for particle-based fluids, developed by Brackbill et al. [1992], is the CSF method. They propose to compute surface tension forces based on the interface curvature at surface particles. Instead of sharp boundaries between fluid phases, a mollifier is used to construct smoothed color interfaces. This allows the computation of color gradients that act as estimates of the interface normals required in the curvature calculation [Brackbill et al. 1992]. The CSF method was initially formulated as a *particle-in-cell* (PIC) approach and later adopted for SPH fluids by Morris [2000]. Due to its straightforward integration into an SPH framework, the CSF method was repeatedly used in subsequent works [Adami et al. 2010; Akinci et al. 2013; Hu and Adams 2006; Müller et al. 2003; Wang et al. 2017; Zorilla et al. 2020]. However, some CSF formulations do not conserve momentum [Adami et al. 2010; Morris 2000] and the normal estimation through normalized color gradients is noisy for particles inside the fluid body [Akinci et al. 2013; Becker and Teschner 2007; Hyde et al. 2020]. Our method, on the other hand, naturally conserves linear and angular momentum and does not rely on the estimation of surface normals. Hu and Adams [2006] avoid the normal computation by directly computing forces using surface stress tensors based on the color gradients. Müller et al. [2003] present a modification that enables the CSF method to be used in free-surface simulations. Since the curvature is proportional to the second derivative of the color field, curvature estimation with SPH is sensitive to particle disorder, especially at the free surface where particle neighborhoods are incomplete [Akinci et al. 2013; Becker and Teschner 2007; Jeske et al. 2023]. To alleviate the estimation sensitivity, He et al. [2015b] propose to compute surface tension in a two-step process: first the color gradients are estimated and then surface tension forces are computed as gradients of the squared norm of the color gradients. In section 6.2, we refer to their approach as *robust CSF*. Similar to He et al. [2015b], the method of Orthmann et al. [2013] uses the magnitude of the gradient of a signed-distance function between the center of the local iso-density distribution and particle position to estimate a particles' contribution to the surface. In contrast, our method avoids any stability problems associated with an approximation of second order derivatives at surface particles by detecting particle surface contributions using number densities.

2.1.2 *Hybrid models.*

Approaches such as Akinci et al. [2013] and Wang et al. [2017] combine pairwise attraction forces with curvature-based surface tension computation in an attempt to combine advantages of both methods. Detailed comparisons between pairwise and CSF surface tension models have been presented by Huber et al. [2015], Yang et al. [2019] and Arai et al. [2020].

Instead of a surface tension formulation purely based on SPH particles, there are approaches that first explicitly reconstruct surfaces based on the underlying fluid particle representation. The force computation is then performed based on this secondary surface description: Yu et al. [2012] and Xing et al. [2022] build a triangle mesh from particle positions that models the fluid surface. Similarly, Boyd and Bridson [2012] reconstruct the interface as a level-set from the positions of particles simulated with the *Fluid-Implicit-Particle* (FLIP) method. To represent the surface of their fluid simulated with a material point method, Hyde et al. [2020] sample fluid interfaces with particles, arguing that it is easier to implement compared to most front tracking and unstructured discretization approaches that require dynamic remeshing. Thürey et al. [2010] compute surface tension on a grid and employ a mesh-based surface representation for sub-grid surface flow. Zhang et al. [2012] simulate small droplets entirely based on a surface mesh description, arguing that the droplet motion is dominated by its boundary surface forces. Other publications focus on the simulation of thin sheets phenomena such as bubbles and fluid splashes [Batty et al. 2012; Da et al. 2015; Ishida et al. 2017; Zhu et al. 2015, 2014]. Wang et al. [2020] describe the construction and tracking of surface meshes as tedious. They propose a codimensional moving-least-square method that directly bases on the particle representation. Surface particles are explicitly classified by their feature type where codimensionality-0 means a particle is inside a fluid volume, particles with codimensionality-1 are part of a thin fluid sheet, particles forming filaments are labeled with codimensionality-2, and fluid points have codimensionality-3. In our approach, surface tension forces are also computed directly based on SPH particles. No secondary surface representation is required. Similar to Wang et al. [2020] and Zorilla et al. [2020], we also perform a surface particle detection by estimating to what part particles contribute to interfaces, but unlike Wang et al. [2020] we do not require an explicit classification of their codimensionality in order to simulate the thin fluid features shown in sections 6.1.1 and 6.6.3.

2.2 Fluid Friction

To some degree, depending on the material, fluid droplets stick to an inclined surface. On a smooth underlying surface, surface tension alone does not oppose the sliding motion of droplets and as such is not suited to model this phenomenon [Akinci et al. 2013]. Earlier approaches try to model fluid-boundary friction by integrating the boundary in the fluid viscosity computation [Akinci et al. 2012; Becker and Teschner 2007; Bender et al. 2020; Gissler et al. 2020; Macklin and Müller 2013; Morris et al. 1997; Müller et al. 2004; Peer et al. 2015; Peer and Teschner 2017; Schechter and Bridson 2012;

Stomakhin et al. 2019; Weiler et al. 2018]. Similar to viscous forces, Ren et al. [2018] employ a friction force that scales with the relative velocity between fluid and boundary mesh to simulate surface flows. Considering viscous forces at the fluid-boundary interface is important to replicate the no-slip condition that is assumed in many theoretical flow solutions. However, since the magnitude of these forces approaches zero for small relative velocities to the boundary, droplets can not truly stick through viscous forces alone when positioned on a slope [Pennestri et al. 2016]. In addition, viscous forces do not necessarily act in a direction tangential to the boundary surface [Koschier and Bender 2017]. In their more recent work, Koschier and Bender [2017] and Winchenbach et al. [2020] mention the use of an explicit friction model for fluid particles at the boundary inspired by the Coulomb friction law, though in their final force formulation, friction is still proportional to the relative velocity between fluid and boundary. As such, neither of them shows the desired effect of fluid coming to rest on an inclined surface. Zhang et al. [2012] employ a tangential friction force that is also quite similar to Coulomb friction in their droplet simulation, but do not consider the influence of the normal force magnitude on friction. Motivated by multiple current publications, including experimental research indicating that Coulomb friction can indeed be used to accurately describe the behavior of droplets on sloped planes [Gao et al. 2018; Hardt and McHale 2022; McHale et al. 2022], we propose an implicit particle-based proper Coulomb friction force that can be used to model physically meaningful tangential forces at the fluid-solid interface.

2.3 Unified Force Computation

There is a long-lasting trend in the computer graphics SPH community to prefer implicit force computation methods over their explicit counterparts [Ihmsen et al. 2014b; Jeske et al. 2023; Koschier et al. 2022], but traditional approaches typically utilize separate and independent solvers for each individual simulated force [Bender and Koschier 2015; Ihmsen et al. 2014a; Koschier et al. 2022; Peer et al. 2018, 2015; Weiler et al. 2018]. In contrast to this, more recent publications such as Macklin et al. [2014], Wang et al. [2020], Takahashi and Batty [2020], Liu et al. [2022], Probst and Teschner [2023] and Jeske et al. [2023] propose and evaluate unified solving mechanisms for a combined computation of forces with promising results. Encouraged by their previous work, in this paper we develop a novel implicit unified solver that is able to simultaneously handle incompressibility, surface tension and fluid friction.

3 Forces

In this section, we formulate pressure, surface tension and friction forces for fluid particles in an SPH environment. A summary of abbreviations and notation used throughout the paper is given in tables 1 and 2. As a pressure solver, we employ a variant of the well established IISPH method [Ihmsen et al. 2014a]. New formulations are developed for surface tension and fluid-boundary friction.

3.1 Pressure

The purpose of pressure forces is to preserve the fluid volume. For an incompressible particle fluid F, this means that all particles $f \in F$ should maintain a volume V_f (m^3) bigger or equal to their respective rest volume V_f^0 (m^3). In a three-dimensional setting, V_f^0 is typically set equal to h^3 with particle spacing h (m). We define a volume error

Table 1. Notation used throughout the paper.

symbol	description
f	fluid particle
b	boundary particle
f_f	fluid neighbors of f
f_b	boundary neighbors of f
b_f	fluid neighbors of b
b_b	boundary neighbors of b
F	fluid body, set of particles f
B	boundary, set of particles b
FV	fluid-vapor interface
FF	fluid-fluid interface
FB	fluid-boundary interface
BV	boundary-vapor interface
BF	boundary-fluid interface

$$V_f^{\mathrm{err}} \equiv 1 - \frac{V_f^0}{V_f} \tag{1}$$

which becomes negative if and only if particle f is compressed. Since we consider incompressible fluids, forces due to pressure p (N m^{-2}) at fluid particles f should maintain $V_f^{\mathrm{err}} \geq 0$ for all f. In addition, p_f is constrained to always be non-negative and can only become non-zero if $V_f^{\mathrm{err}} \leq 0$, meaning that we do not allow positive pressure in parts where current fluid volume V is greater than its rest volume V^0. Together, these constraints form a linear complementarity problem [Andersen et al. 2017] that can be written down in short as

$$0 \leq p_f \qquad \perp \qquad V_f^{\mathrm{err}} \geq 0. \tag{2}$$

A basic SPH interpolation can be employed to approximate V_f^{err}:

$$V_f^{\mathrm{err}} = 1 - V_f^0 \sum_{f_f} W_{f,f_f}^{\mathrm{P}} - V_f^0 \sum_{f_b} W_{f,f_b}^{\mathrm{P}}. \tag{3}$$

Here, fluid neighbors of particle f are denoted by f_f while boundary neighbors are written as f_b. $W_{i,j}^{\mathrm{P}}$ is a shorthand notation for the SPH smoothing kernel function $W(\mathbf{x}_i - \mathbf{x}_j, \hbar^{\mathrm{P}})$ with pressure kernel support $\hbar^{\mathrm{P}}$ (m) and positions $\mathbf{x}$. After calculating pressure p that satisfies eq. (2), pressure forces $\mathbf{F}^{\mathrm{P}}$ (N) are computed with

$$\begin{aligned} \mathbf{F}_f^{\mathrm{P}} &= -V_f^0 \nabla p_f \\ &= -V_f^0 \sum_{f_f} V_{f_f}^0 \left(p_f + p_{f_f}\right) \nabla W_{f,f_f}^{\mathrm{P}} \\ &\quad - V_f^0 \sum_{f_b} V_{f_b}^0 \left(p_f + p_{f_b}\right) \nabla W_{f,f_b}^{\mathrm{P}}. \end{aligned} \tag{4}$$

This approximation for pressure forces conserves linear and angular momentum exactly. Note that there are many ways to define pressure at boundary particles p_{f_b} [Akinci et al. 2012; Band et al. 2018a,b; Bender et al. 2019, 2020, 2023; Koschier and Bender 2017], all of which can be used here. We follow the recent idea of Bender et al. [2023] and set $p_{f_b} = 0$. In section 4, we describe the pressure

Table 2. Summary of used symbols. Units are given for a three-dimensional setting.

symbol	description	unit
A	interface area	m^2
A^0	rest area	m^2
C	contribution to an interface	
E	energy	$\mathrm{N\,m}$
$\mathbf{F}$	force	N
$\mathbf{F}^{\mathrm{F}}$	friction force	N
$\mathbf{F}^{*}$	target friction force	N
$\mathbf{F}^{N}$	normal force	N
$\mathbf{F}^{\mathrm{P}}$	pressure force	N
$\mathbf{F}^{\mathrm{ST}}$	surface tension force	N
$\mathcal{F}$	set of allowed friction values	
h	particle size	m
$\hbar^{\mathrm{P}}$	pressure kernel support	m
$\hbar^{\mathrm{ST}}$	surface tension kernel support	m
$\hbar^{\mathrm{F}}$	friction kernel support	m
$\mathbf{I}$	identity matrix	
m	mass	kg
$\max_\epsilon(C, 0)$	smoothed clamping of C	
p	pressure	$\mathrm{N\,m}^{-2}$
$\mathcal{P}$	set of all positive pressure values	
prox ( )	proximal operator	
S	derivative of $\max_\epsilon$ w.r.t. C	
t	time	s
$\mathbf{v}$	velocity	$\mathrm{m\,s}^{-1}$
$\mathbf{v}^{\mathrm{rel}}$	relative velocity	$\mathrm{m\,s}^{-1}$
$\mathbf{v}^{\mathrm{tang}}$	tangential relative velocity	$\mathrm{m\,s}^{-1}$
V	volume	m^3
V^0	rest volume	m^3
V^{err}	volume error	
W	SPH smoothing kernel	m^{-3}
W^{P}	kernel with support radius $\hbar^{\mathrm{P}}$	m^{-3}
W^{ST}	kernel with support radius $\hbar^{\mathrm{ST}}$	m^{-3}
W^{F}	kernel with support radius $\hbar^{\mathrm{F}}$	m^{-3}
$\mathbf{x}$	position	m
α	fixed-point convergence coefficient	
γ	surface energy density	$\mathrm{N\,m}^{-1}$
Δt	timestep	s
ϵ	smoothing strength	
μ	coefficient of friction	
ρ^0	rest density	$\mathrm{kg\,m}^{-3}$
ϖ	Jacobi relaxation coefficient	

solver used to calculate p and its connection to the surface tension and friction computation procedure in greater detail.

3.2 Surface Tension

Surface tension forces arise due to potential energy that is required to form an interface between different phases. In the literature, the potential surface energy E (J) is often defined by the interface area A (m^2) and the respective surface tension coefficient γ ($\mathrm{J\,m}^{-2}$) [Lautrup 2011]:

$$E = A\gamma. \tag{5}$$

Similar to He et al. [2015b] and Hyde et al. [2020], our surface force computation is derived from an estimation of E. For this, we first consider the total surface energy in our particle simulation by summing up contributions of all fluid particles f and boundary particles b:

$$E = \sum_f E_f + \sum_b E_b. \tag{6}$$

Note that also boundary particles b contribute to the total surface energy in our simulation as the boundary-vapor interface can store potential surface energy too. In section 3.2.1 we will see that contributions from boundary particles greatly determine fluid behavior at the fluid-boundary-vapor contact line. For reasons of simplicity, instead of requiring a user to provide surface tension coefficients γ for all possible pairs of fluid phases and boundaries, in our simulation all particles f belonging to the same fluid phase F are given one surface tension coefficient for the fluid-vapor interface γ^{FV}, one coefficient γ^{FF} that describes the interface energy density to other fluid phases and one coefficient γ^{FB} for the fluid-boundary interface. Fluid phases F are sets of fluid particles, where each fluid particle f belongs to exactly one fluid phase F. Likewise, per boundary B we allow users to define the boundary-vapor interface energy density γ^{BV} and the boundary-fluid interface energy density γ^{BF}. To give a better intuition, in fig. 2 we sketch the interface forces arising due to γ and in section 6.1.4 we showcase the resulting fluid behavior for a selection of different configurations of γ. The surface energy of a particle including all possible interfaces can now be written as

$$E_f = \gamma_f^{\mathrm{FV}} A_f^{\mathrm{FV}} + \gamma_f^{\mathrm{FF}} A_f^{\mathrm{FF}} + \gamma_f^{\mathrm{FB}} A_f^{\mathrm{FB}} \tag{7a}$$

$$E_b = \gamma_b^{\mathrm{BV}} A_b^{\mathrm{BV}} + \gamma_b^{\mathrm{BF}} A_b^{\mathrm{BF}}. \tag{7b}$$

Here, A_f^{FV}, A_f^{FF} and A_f^{FB} represent the estimated interface area fluid particle f has to the vapor phase, other fluid phases and to the boundary respectively. A_b^{BV} and A_b^{BF} are the estimated interface areas of boundary particle b to the vapor phase and to adjacent fluids. The way interface areas A are approximated is crucial for the effectiveness and robustness of the force computation [He et al. 2015b]. Approaches based on the CSF method [Brackbill et al. 1992; He et al. 2015b; Hu and Adams 2006; Morris 2000; Müller et al. 2003] utilize a color gradient as a tool to measure the contribution of a particle to the interface. The normalized color gradients can then be used to approximate the local curvature of the interface from which surface tension forces are computed. This approach however entails multiple problematic aspects. First, the color gradients should become zero inside the fluid, so one has to be careful when computing surface normals at those particles [Becker and Teschner 2007; Hu and Adams 2006; Müller et al. 2003]. Second, since the curvature is proportional to the second derivative of the color field, a direct estimation is sensitive to particle disorder, which is especially critical close to the free surface where particles have fewer neighbors [Becker and Teschner 2007; He et al. 2015b]. While we note that there exist approaches that try to mitigate said problems [He et al. 2015b; Hu and Adams 2006], to avoid any difficulties associated with the approximation of second order derivatives, we propose a new surface detection method that does not rely on the gradient of the color field, but instead utilizes number density estimations to detect surface particles. For this, at each fluid particle f and boundary particle b we compute an SPH number density summation to approximate contributions C towards interfaces to other phases:

$$C_f^{\mathrm{FV}} = 1 - \sum_{f_f} V_{f_f}^0 W_{f,f_f}^{\mathrm{ST}} - \sum_{f_b} V_{f_b}^0 W_{f,f_b}^{\mathrm{ST}} \tag{8a}$$

$$C_f^{\mathrm{FF}} = \sum_{f_f \notin F} V_{f_f}^0 W_{f,f_f}^{\mathrm{ST}} \tag{8b}$$

$$C_f^{\mathrm{FB}} = \sum_{f_b} V_{f_b}^0 W_{f,f_b}^{\mathrm{ST}} \tag{8c}$$

$$C_b^{\mathrm{BV}} = 1 - \sum_{b_f} V_{b_f}^0 W_{b,b_f}^{\mathrm{ST}} - \sum_{b_b} V_{b_b}^0 W_{b,b_b}^{\mathrm{ST}} \tag{8d}$$

$$C_b^{\mathrm{BF}} = \sum_{b_f} V_{b_f}^0 W_{b,b_f}^{\mathrm{ST}}. \tag{8e}$$

W^{ST} denotes the SPH smoothing kernel with a support range $\hbar^{\mathrm{ST}}$ used for surface tension related SPH interpolations. The sum over $f_f \notin F$ includes all fluid neighbors of f that are not part of the same fluid phase F as f. Fluid particles that neighbor a boundary particle b are denoted by b_f, boundary neighbors of b are written as b_b. Note that by using number densities, the approximation of interface contributions C naturally returns zero if no interface is detected and approaches one if a particle is completely surrounded by the other phase. This also holds true for the interface to the vapor phase. We now want to transform the interface contributions C into estimates of how much actual area A a particle contributes to an interface. For this, we need to clamp C at zero to prevent nonphysical negative area estimates when C^{FV} becomes smaller than zero inside a compressed fluid. Additionally, we want A to have a vanishing derivative for C close to zero in order to be robust against weak noise due to SPH approximation errors in the estimation of interface contributions C inside a fluid body. Thus, instead of simply clamping values of C, we perform a clamping using the smoothed maximum function $\max_\epsilon$ defined by

$$\max_{\epsilon}(C,0) \equiv -\epsilon + \sqrt{\max(C,0)^2 + \epsilon^2}. \tag{9}$$

Figure 3 shows a plot of $\max_\epsilon$ and its derivative. In section 5.1 we discuss the effects and an optimal value for ϵ in greater detail. To compute the interface areas A of a particle, we now scale the corresponding softly clamped C by some rest area A^0:

$$A = A^0 \max_{\epsilon}(C,0) \tag{10}$$

for all possible interface areas A^{FV}, A^{FF}, A^{FB}, A^{BV} and A^{BF}. Depending on which interface area A is estimated, the generic placeholder C is substituted by the respective interface contribution from eq. (8). In section 5.2 we further discuss the values given to A^0, for the derivation of forces it suffices to know that all A_f^0 and all A_b^0 are equal and constant over time.

3.2.1 Surface Force Derivation. Surface tension forces $\mathbf{F}^{\mathrm{ST}}$ (N) acting at a fluid particle f are equal to the negative derivative of the surface tension potential E described in eqs. (6) to (10) with respect

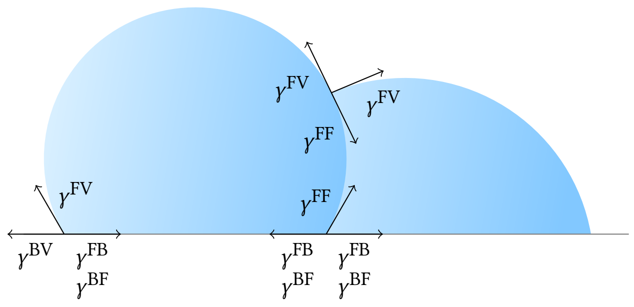


Fig. 2. Surface tension parametrization by defining surface energy densities γ. Surface tension acts towards minimal interface areas scaled by their respective γ. In our implementation we allow users to define γ per fluid for the fluid-vapor interface (FV), fluid-fluid interfaces (FF) and fluid-boundary interfaces (FB). Similarly, for each boundary the user can specify the energy density γ of boundary-vapor (BV) and boundary-fluid (BF) interfaces. We sketched the direction of the individual surface tension force components at three exemplary locations. Note that when using this parametrization, the forces minimizing fluid-fluid and fluid-boundary interfaces are the result of a combination of the respective surface tension parameters γ from both adjacent phases.

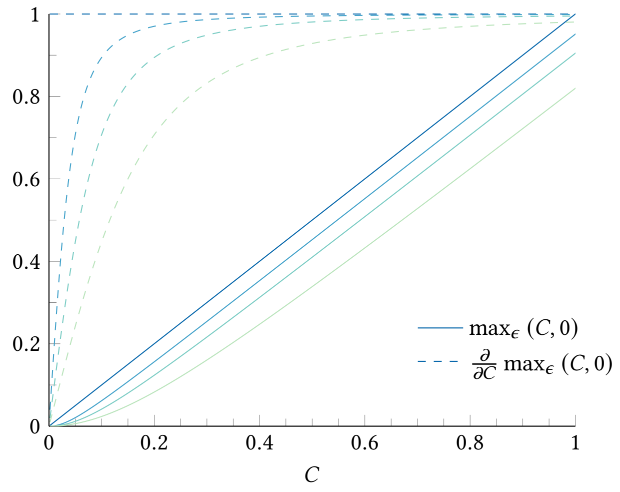


Fig. 3. The smoothed maximum function $\max_\epsilon(C,0)$ and its derivative with respect to C for a selection of different values given to ϵ. The dark blue curves are plotted using $\epsilon = 0$. In this edge case, $\max_\epsilon(C,0)$ equals $\max(C,0)$ for positive C. As such, it has a derivative of one with a discontinuity around zero. The curves displayed in lighter blue and greenish colors use ϵ equal to 0.05, 0.1 and 0.2 respectively. We can see that higher ϵ causes stronger smoothing of $\max_\epsilon(C,0)$ and its derivative. While the derivative always equals zero at $C = 0$ as long as $\epsilon > 0$, for larger C the smoothed maximum function $\max_\epsilon(C,0)$ approaches a linear shape with a derivative of one. Note that later in the text, we introduce the shorthand notation for the derivative $S \equiv \frac{\partial}{\partial C}\max_\epsilon(C,0)$.

to particle position $\mathbf{x}_f$:

$$\begin{aligned}\mathbf{F}_f^{\mathrm{ST}} &= -\frac{\partial}{\partial \mathbf{x}_f} E \\ &= \sum_{f_f} \left(A_f^0 V_{f_f}^0 \gamma_f^{\mathrm{FV}} S_f^{\mathrm{FV}} + A_{f_f}^0 V_f^0 \gamma_{f_f}^{\mathrm{FV}} S_{f_f}^{\mathrm{FV}} \right) \nabla W_{f,f_f}^{\mathrm{ST}} \\ &+ \sum_{f_f \notin F} \left(-A_f^0 V_{f_f}^0 \gamma_f^{\mathrm{FF}} S_f^{\mathrm{FF}} - A_{f_f}^0 V_f^0 \gamma_{f_f}^{\mathrm{FF}} S_{f_f}^{\mathrm{FF}} \right) \nabla W_{f,f_f}^{\mathrm{ST}} \\ &+ \sum_{f_b} \left(A_f^0 V_{f_b}^0 \gamma_f^{\mathrm{FV}} S_f^{\mathrm{FV}} + A_{f_b}^0 V_f^0 \gamma_{f_b}^{\mathrm{BV}} S_{f_b}^{\mathrm{BV}} \right) \nabla W_{f,f_b}^{\mathrm{ST}} \\ &+ \sum_{f_b} \left(-A_f^0 V_{f_b}^0 \gamma_f^{\mathrm{FB}} S_f^{\mathrm{FB}} - A_{f_b}^0 V_f^0 \gamma_{f_b}^{\mathrm{BF}} S_{f_b}^{\mathrm{BF}} \right) \nabla W_{f,f_b}^{\mathrm{ST}}.\end{aligned} \tag{11}$$

For better readability we introduced the shorthand notation

$$S \equiv \frac{\partial}{\partial C} \max_{\epsilon}(C, 0) = \frac{\max(C, 0)}{\sqrt{C^2 + \epsilon^2}}. \tag{12}$$

Constructed from our surface energy potential E, this purely particle-based surface tension force naturally conserves momentum exactly, as shown in appendix C, while only requiring a first order derivative approximation. Note that compared to pairwise surface tension formulations [Akinci et al. 2013; Becker and Teschner 2007; Jeske et al. 2023], eq. (11) shares the similarly intuitive attractive and repulsive force components between neighboring particles. However, as a crucial difference to pairwise approaches, the inclusion of S has the effect that surface tension forces only act between particles that are part of the interface, not inside the whole fluid body. We further evaluate the role of S in section 6.2 and demonstrate its relevance for computing physically correct surface tension forces.

3.2.2 Discussion. Existing surface tension models usually neglect the influence of the boundary-vapor interface tension γ^{BV} at the fluid-boundary-vapor contact line [Akinci et al. 2013; Becker and Teschner 2007; He et al. 2015b; Jeske et al. 2023; Morris 2000; Müller et al. 2003; Wang et al. 2017]. Instead, typically a simple parameter is used to scale surface tension forces at the fluid-boundary-vapor interface, without distinguishing between boundary-vapor and boundary-fluid tension [Akinci et al. 2013; Becker and Teschner 2007; Jeske et al. 2023; Wang et al. 2017]. As shown in section 6.1.4, the correct incorporation of γ^{BV} and γ^{BF} into the surface tension computation allows us to replicate effects such as complete wetting or hydrophobic droplet behavior with an intuitive parametrization.

Our surface force formulation in eq. (11) assumes that the boundary is sampled with particles. This way, we can easily distinguish between boundary parts that belong to the boundary-fluid interface where particles have S^{BF} close to one and parts of the boundary-vapor interface where S^{BV} is approximately one. Alternative boundary descriptions often promise a smoother boundary representation [Bender et al. 2019; Bodin et al. 2012; Fujisawa and Miura 2015; Koschier and Bender 2017]. However, to be viable in combination with our surface tension, these boundary descriptions must be able to separate parts of the boundary covered by fluid from parts in contact with the vapor phase. Only then it is possible to define an interface energy potential equivalent to eq. (7b) and derive surface tension forces from it.

3.3 Boundary Friction

The last force we want to consider here stems from friction between fluid and solid boundary. In this section we present the basic formulation of our particle-based Coulomb friction which will be used in section 4 to build an implicit force computation. The Coulomb law requires the magnitude of friction forces $\mathbf{F}^{\mathrm{F}}$ (N) to be smaller than or equal to the product between the friction coefficient μ and the magnitude of a normal force $\mathbf{F}^N$ (N) [Bender et al. 2014]:

$$\mathbf{F}_f^{\mathrm{F}} \in \mathcal{F}_f \tag{13a}$$

$$\mathcal{F}_f \equiv \left\{ \mathbf{F} \in \mathbb{R}^3 \;\middle|\; \mathbf{F} \cdot \mathbf{F}_f^N = 0 \text{ and } |\mathbf{F}| \leq \mu_f |\mathbf{F}_f^N| \right\}. \tag{13b}$$

Similar to the rigid body friction model presented by Probst and Teschner [2023], as an estimate for the normal force $\mathbf{F}_f^N$ we extract the part of the pressure force described in eq. (4) at fluid particle f that is due to the adjacent boundary:

$$\mathbf{F}_f^N = -V_f^0 \sum_{f_b} V_{f_b}^0 p_f \nabla W_{f,f_b}^{\mathrm{F}}. \tag{14}$$

Here, W^{F} denotes the SPH smoothing kernel with a support range $\hbar^{\mathrm{F}}$ used for friction related SPH interpolations. $\mathbf{F}_f^N$ not only gives us a good approximation of the boundary normal at fluid particle f, it also encodes the actual normal force employed in our simulation in the form of pressure forces at f. In addition to eq. (13), friction forces at fluid particles f should act in tangential direction to the contacting boundary. The principle of maximum dissipation constrains the direction in which friction forces act by [Erleben 2017]

$$\mathbf{F}_f^{\mathrm{F}} = \operatorname*{arg\,min}_{\mathbf{F} \in \mathcal{F}_f} \mathbf{F} \cdot \mathbf{v}_f^{\mathrm{tang}} \tag{15}$$

with $\mathbf{v}_f^{\mathrm{tang}}$ representing the relative velocity $\mathbf{v}_f^{\mathrm{rel}}$ between particle f and its adjacent boundary, projected onto the tangential contact plane. Note that in our case we model isotropic friction, so for $\mathbf{v}_f^{\mathrm{tang}} \neq \mathbf{0}$, eq. (15) simply states that friction forces should oppose the relative sliding motion of a particle f with the maximum strength allowed by the Coulomb constraint. Also inspired by Probst and Teschner [2023], the tangential component $\mathbf{v}_f^{\mathrm{tang}}$ of the relative velocity $\mathbf{v}_f^{\mathrm{rel}}$ is estimated using

$$\mathbf{v}_f^{\mathrm{tang}} = \left(\mathbf{I} - \frac{\mathbf{F}_f^N \mathbf{F}_f^{N\top}}{|\mathbf{F}_f^N|^2} \right) \mathbf{v}_f^{\mathrm{rel}} \tag{16a}$$

$$\mathbf{v}_f^{\mathrm{rel}} = \sum_{f_b} V_{f_b}^0 \left(\mathbf{v}_f - \mathbf{v}_{f_b} \right) W_{f,f_b}^{\mathrm{F}} \tag{16b}$$

with identity matrix $\mathbf{I}$. Here, we used the normal force $\mathbf{F}_f^N$ to determine the boundary normal required to project $\mathbf{v}_f^{\mathrm{rel}}$ onto the tangential contact plane. A simple SPH summation is used to approximate $\mathbf{v}_f^{\mathrm{rel}}$. We want to point out that while our friction approach shares similarities with the dry friction model by Probst and Teschner [2023] concerning basic particle-based approximations of surface normals and relative velocities, in contrast to Probst and Teschner [2023] we formulated our friction model in such a way that we directly solve for friction forces $\mathbf{F}^{\mathrm{F}}$ instead of abstract *friction multipliers*. This not only makes our friction optimization problem more intuitive, we are also no longer required to perform a costly friction processing step to transform friction multipliers into actual friction forces.

4 Solver Implementation

Previously in section 3, the main concepts and formulations for a particle-based pressure, surface tension and friction force have been presented. Building on this foundation, we can now evolve the concepts into an actual simulation method. Inspired by previous work that unifies the computation of multiple forces into one solving procedure [Jeske et al. 2023; Liu et al. 2022; Macklin et al. 2014; Probst and Teschner 2023; Takahashi and Batty 2020; Wang et al. 2020], in this section we develop an implicit unified solver that is able to simultaneously compute pressure, surface tension and fluid friction. We showcase the results of our combined solving mechanism and compare it to the separated approach in section 6.4.

4.1 Implicit Formulation

When computing forces implicitly, one does not consider the current particle-state at time t, but instead the predicted state at time $t+\Delta t$ with simulation timestep Δt. Since particle positions and velocities at time $t+\Delta t$ already depend on the forces computed at time t, a system needs to be solved in order to obtain the unknown forces. Revisiting the pressure, surface tension and friction force descriptions in eqs. (2), (11) and (15), an equivalent implicit problem formulation reads as

$$0 \le p_f \quad \perp \quad V_f^{\text{err}}(t+\Delta t) \ge 0. \tag{17a}$$

$$\mathbf{F}_f^{\text{ST}} = -\frac{\partial}{\partial \mathbf{x}_f} E(t+\Delta t) \tag{17b}$$

$$\mathbf{F}_f^{\text{F}} = \underset{\mathbf{F}\in\mathcal{F}_f}{\arg\min}\ \mathbf{F}\cdot\mathbf{v}_f^{\text{tang}}(t+\Delta t). \tag{17c}$$

Here, for the pressure computation we now consider the volume errors V^{err} at the next timestep $t+\Delta t$, for the surface tension force we consider the surface energy potentials E at time $t+\Delta t$ just as we consider the tangential parts of the relative velocities $\mathbf{v}^{\text{tang}}$ at time $t+\Delta t$ for the friction computation. As shown by Erleben [2017] and Probst and Teschner [2023], all three optimization problems can be converted into equivalent fixed-point problems [Courtecuisse and Allard 2009; Schindler et al. 2011]

$$p_f = \text{prox}_{\mathcal{P}_f}\left(p_f - \varpi\alpha_f^{\text{P}} V_f^{\text{err}}(t+\Delta t)\right) \tag{18a}$$

$$\mathbf{F}_f^{\text{ST}} = \mathbf{F}_f^{\text{ST}} - \varpi\alpha_f^{\text{ST}}\left(\mathbf{F}_f^{\text{ST}} + \frac{\partial}{\partial \mathbf{x}_f}E(t+\Delta t)\right) \tag{18b}$$

$$\begin{aligned} \mathbf{F}_f^{*} &= \text{prox}_{\mathcal{F}_f}\left(-\alpha_f^{\text{F}}\mathbf{v}_f^{\text{tang}}(t+\Delta t)\right) \\ \mathbf{F}_f^{\text{F}} &= \text{prox}_{\mathcal{F}_f}\left(\mathbf{F}_f^{\text{F}} + \varpi\mathbf{F}_f^{*}\right) \end{aligned} \tag{18c}$$

with $\mathcal{P}_f$ being the set of all valid pressure values and $\mathcal{F}_f$ the set of all friction forces allowed by the Coulomb constraint:

$$\mathcal{P}_f \equiv \left\{ p \in \mathbb{R} \;\middle|\; p \ge 0 \right\} \tag{19}$$

$$\mathcal{F}_f \equiv \left\{ \mathbf{F} \in \mathbb{R}^3 \;\middle|\; \mathbf{F}\cdot\mathbf{F}_f^N = 0 \text{ and } |\mathbf{F}| \le \mu_f|\mathbf{F}_f^N| \right\}. \tag{(13b) revisited}$$

The proximal operator $\text{prox}_{\mathcal{S}}(x) \equiv \arg\min_{s\in\mathcal{S}} |s-x|$ projects its input x onto the nearest point in some given set $\mathcal{S}$. Note that the pressure projection in eq. (18a) corresponds to the clamping of negative pressure values employed by IISPH [Ihmsen et al. 2014a]. The projection into $\mathcal{F}$ in eq. (18c) can be implemented with $\text{prox}_{\mathcal{F}_f}(\mathbf{F}) = \min(\mu_f|\mathbf{F}_f^N|/|\mathbf{F}|, 1)\mathbf{F}$, assuming that input $\mathbf{F}$ is already perpendicular to $\mathbf{F}^N$. To improve the stability of the fixed point iterations, we follow Probst and Teschner [2023] and split the friction fixed-point problem into a two-step iteration with intermediate target forces $\mathbf{F}^*$. A detailed discussion on why this version of the friction update should be preferred over a one-step formulation is given in Probst and Teschner [2023].

4.1.1 *Step size.*

Solving the fixed point problem in eq. (18) gives us the unknown pressure p, surface tension forces $\mathbf{F}^{\text{ST}}$ and friction $\mathbf{F}^{\text{F}}$. To do so, we choose strictly positive scalar values for α^{P}, α^{ST} and α^{F} according to Erleben [2017] and Probst and Teschner [2023], such that eq. (18) is equivalent to one iteration of a projected relaxed Jacobi solver. By repeatedly applying eq. (18) we obtain a robust Jacobi solving procedure that is able to implicitly compute p, $\mathbf{F}^{\text{ST}}$ and $\mathbf{F}^{\text{F}}$. Since the basic ideas concerning the update step sizes α of Jacobi iterations are already well explained by e.g. Ihmsen et al. [2014a], Bender and Koschier [2015], Gissler et al. [2019], Koschier et al. [2022] and Probst and Teschner [2023], we move the discussion of optimal values and efficient computation procedures for α^{P}, α^{ST} and α^{F} to appendix A. The relaxation coefficient ϖ can be used to balance solver convergence speed and robustness. Following previous work [Ihmsen et al. 2014a; Koschier et al. 2022], we always use $\varpi = 0.5$. Note that the provided values for α and ϖ do not guarantee solver convergence on an analytical level, but instead are based on experience from earlier publications that demonstrate reliable solver convergence for a large selection of simulation scenarios [Bender and Koschier 2015; Erleben 2017; Gissler et al. 2019; Ihmsen et al. 2014a; Koschier et al. 2022; Probst and Teschner 2023; Solenthaler and Pajarola 2009].

4.2 Unified Forces

Equation (18) gives us a description of the main solver update used to implicitly compute p_f, $\mathbf{F}_f^{\text{ST}}$ and $\mathbf{F}_f^{\text{F}}$ for all fluid particles f. This section discusses the required computations necessary to carry out one update step of unknown p, $\mathbf{F}^{\text{ST}}$ and $\mathbf{F}^{\text{F}}$ as described in eq. (18). In order to unify the computation of pressure, surface tension and friction, all forces must be able to already consider the effects of the other two forces that are computed simultaneously. To achieve this, after each solver update we estimate new predicted velocities $\mathbf{v}_f(t+\Delta t)$ including pressure forces $\mathbf{F}_f^{\text{P}}$, surface tension forces $\mathbf{F}_f^{\text{ST}}$ and friction forces $\mathbf{F}_f^{\text{F}}$:

$$\mathbf{v}_f(t+\Delta t) = \mathbf{v}_f(t) + \Delta t\frac{\mathbf{F}_f^{\text{P}} + \mathbf{F}_f^{\text{ST}} + \mathbf{F}_f^{\text{F}}}{m_f} \tag{20}$$

where $m_f = V_f^0\rho_f^0$ is the particle mass and ρ_f^0 the fluid rest density. We assume that all explicit forces such as gravity and viscous forces are already included in $\mathbf{v}_f(t)$. Pressure forces $\mathbf{F}_f^{\text{P}}$ are computed from the current pressure p using eq. (4). With the current predicted velocities $\mathbf{v}_f(t+\Delta t)$ at hand, we can then update the prediction for $V_f^{\text{err}}(t+\Delta t)$, $-\frac{\partial}{\partial \mathbf{x}_f}E(t+\Delta t)$ and $\mathbf{v}_f^{\text{tang}}(t+\Delta t)$, which are required in the next solver update. In the following, we describe the update step in greater detail.

4.2.1 Pressure. Starting with the volume errors required in the pressure update (eq. (18a)), we compute

$$\begin{aligned} &V_f^{\text{err}}(t+\Delta t) \\ &= 1 - V_f^0 \sum_{f_f} W^{\text{P}}_{f,f_f}(t) - V_f^0 \sum_{f_b} W^{\text{P}}_{f,f_b}(t) \\ &\quad - V_f^0 \sum_{f_f} \Delta t \left(\mathbf{v}_f(t+\Delta t) - \mathbf{v}_{f_f}(t+\Delta t)\right) \cdot \nabla W^{\text{P}}_{f,f_f}(t) \\ &\quad - V_f^0 \sum_{f_b} \Delta t \left(\mathbf{v}_f(t+\Delta t) - \mathbf{v}_{f_b}(t+\Delta t)\right) \cdot \nabla W^{\text{P}}_{f,f_b}(t). \end{aligned} \tag{21}$$

Here, we follow Ihmsen et al. [2014a] and employ a linear approximation of $V_f^{\text{err}}(t+\Delta t)$ (compare to eq. (3)). The kinematic boundary is not affected by forces, so we can set $\mathbf{v}_{f_b}(t+\Delta t) = \mathbf{v}_{f_b}(t)$. Since $\mathbf{v}_f(t+\Delta t)$ contains information about friction and surface tension forces as well as pressure forces acting at particle f, $V_f^{\text{err}}(t+\Delta t)$ not only considers the effects of fluid pressure p, but is also directly influenced by the current guess for $\mathbf{F}^{\text{F}}$ and $\mathbf{F}^{\text{ST}}$. This way, the pressure force computation is sensitized to be consistent with friction and surface tension.

4.2.2 Surface Tension. Similar to the pressure iteration, the surface tension solver update from eq. (18b) requires an updated estimate of $-\frac{\partial}{\partial \mathbf{x}_f} E(t+\Delta t)$ based on the current $\mathbf{v}_f(t+\Delta t)$. For this, we reconsider eqs. (8), (11) and (12):

$$\begin{aligned} &-\frac{\partial}{\partial \mathbf{x}_f} E(t+\Delta t) \\ &= \sum_{f_f} \left(A_f^0 V_{f_f}^0 \gamma_f^{\text{FV}} S_f^{\text{FV}}(t+\Delta t) + A_{f_f}^0 V_f^0 \gamma_{f_f}^{\text{FV}} S_{f_f}^{\text{FV}}(t+\Delta t)\right) \nabla W^{\text{ST}}_{f,f_f}(t) \\ &+ \sum_{f_f \notin F} \left(-A_f^0 V_{f_f}^0 \gamma_f^{\text{FF}} S_f^{\text{FF}}(t+\Delta t) - A_{f_f}^0 V_f^0 \gamma_{f_f}^{\text{FF}} S_{f_f}^{\text{FF}}(t+\Delta t)\right) \nabla W^{\text{ST}}_{f,f_f}(t) \\ &+ \sum_{f_b} \left(A_f^0 V_{f_b}^0 \gamma_f^{\text{FV}} S_f^{\text{FV}}(t+\Delta t) + A_{f_b}^0 V_f^0 \gamma_{f_b}^{\text{BV}} S_{f_b}^{\text{BV}}(t+\Delta t)\right) \nabla W^{\text{ST}}_{f,f_b}(t) \\ &+ \sum_{f_b} \left(-A_f^0 V_{f_b}^0 \gamma_f^{\text{FB}} S_f^{\text{FB}}(t+\Delta t) - A_{f_b}^0 V_f^0 \gamma_{f_b}^{\text{BF}} S_{f_b}^{\text{BF}}(t+\Delta t)\right) \nabla W^{\text{ST}}_{f,f_b}(t) \end{aligned} \tag{22}$$

with

$$S(t+\Delta t) = \frac{\max\left(C(t+\Delta t), 0\right)}{\sqrt{C(t+\Delta t)^2 + \epsilon^2}} \tag{23}$$

and

$$\begin{aligned} C_f^{\text{FV}}(t+\Delta t) = 1 &- \sum_{f_f} V_{f_f}^0 W^{\text{ST}}_{f,f_f}(t+\Delta t) \\ &- \sum_{f_b} V_{f_b}^0 W^{\text{ST}}_{f,f_b}(t+\Delta t) \end{aligned} \tag{24a}$$

$$C_f^{\text{FF}}(t+\Delta t) = \sum_{f_f \notin F} V_{f_f}^0 W^{\text{ST}}_{f,f_f}(t+\Delta t) \tag{24b}$$

$$C_f^{\text{FB}}(t+\Delta t) = \sum_{f_b} V_{f_b}^0 W^{\text{ST}}_{f,f_b}(t+\Delta t) \tag{24c}$$

$$\begin{aligned} C_b^{\text{BV}}(t+\Delta t) = 1 &- \sum_{b_f} V_{b_f}^0 W^{\text{ST}}_{b,b_f}(t+\Delta t) \\ &- \sum_{b_b} V_{b_b}^0 W^{\text{ST}}_{b,b_b}(t+\Delta t) \end{aligned} \tag{24d}$$

$$C_b^{\text{BF}}(t+\Delta t) = \sum_{b_f} V_{b_f}^0 W^{\text{ST}}_{b,b_f}(t+\Delta t). \tag{24e}$$

Since

$$\begin{aligned} W^{\text{ST}}_{i,j}(t+\Delta t) \equiv W\Big(&\mathbf{x}_i(t) + \Delta t \mathbf{v}_i(t+\Delta t) \\ &- \mathbf{x}_j(t) - \Delta t \mathbf{v}_j(t+\Delta t), \hbar^{\text{ST}}\Big), \end{aligned} \tag{25}$$

we can see that $-\frac{\partial}{\partial \mathbf{x}_f} E(t+\Delta t)$ depends on $\mathbf{v}(t+\Delta t)$ and as such the surface tension computation procedure directly takes into consideration what effects the simultaneously computed pressure and friction forces have on fluid particles. Note that in contrast to the linear expansion of W employed by IISPH and shown in eq. (21) [Ihmsen et al. 2014a], to improve the estimation accuracy of $C(t+\Delta t)$, in eq. (24) we directly plug in the newly predicted positions $\mathbf{x}(t+\Delta t)$ in the kernel function W^{ST}.

4.2.3 Friction. Friction at particles f is calculated dependent on the predicted tangential relative velocity to the boundary $\mathbf{v}_f^{\text{tang}}(t+\Delta t)$. We can easily incorporate current pressure and surface tension forces in the friction solving process by using $\mathbf{v}_f(t+\Delta t)$ in the estimation of $\mathbf{v}_f^{\text{tang}}(t+\Delta t)$ as shown in eq. (16):

$$\mathbf{v}_f^{\text{tang}}(t+\Delta t) = \left(\mathbf{I} - \frac{\mathbf{F}_f^N \mathbf{F}_f^{N\top}}{|\mathbf{F}_f^N|^2}\right) \mathbf{v}_f^{\text{rel}}(t+\Delta t) \tag{26a}$$

$$\mathbf{v}_f^{\text{rel}}(t+\Delta t) = \sum_{f_b} V_{f_b}^0 \left(\mathbf{v}_f(t+\Delta t) - \mathbf{v}_{f_b}(t+\Delta t)\right) W^{\text{F}}_{f,f_b}(t). \tag{26b}$$

Note that while $\mathbf{F}_f^N$ changes in magnitude during the solving procedure, and as such the set of allowed friction forces $\mathcal{F}_f$ defined in eq. (13b) also varies, looking at eq. (14) it is clear that the direction of $\mathbf{F}_f^N$, which is the only thing relevant for the velocity projection, stays constant. This way we end up with the desired property that only the magnitude of $\mathbf{F}_f^N$ depends on p_f, but the normal direction is purely defined by the geometry of neighboring boundary particles.

4.3 Jacobi Solver

Here, to present a better overview of the whole force computation procedure, we want to give a short summary of a basic Jacobi solver that can be used to find solutions to eq. (18). An illustration of the individual solver steps is shown in algorithm 1. After first initializing the unknown pressure p and forces $\mathbf{F}^{\text{ST}}$ and $\mathbf{F}^{\text{F}}$ with zero similar to other pressure solvers [Bridson 2008], we enter the main solver loop. Each solver iteration starts by computing pressure forces $\mathbf{F}^{\text{P}}$ from p since they are required to estimate $\mathbf{v}(t+\Delta t)$. Normal forces $\mathbf{F}^N$ are also recalculated using the current p, since we need them later on during the Jacobi update step of friction forces $\mathbf{F}^{\text{F}}$ (line 21). With the current predicted velocities $\mathbf{v}(t+\Delta t)$ at hand, we can then update $S_f^{\text{FV}}(t+\Delta t)$, $S_f^{\text{FF}}(t+\Delta t)$ and $S_f^{\text{FB}}(t+\Delta t)$ for all fluid particles f, as well as $S_b^{\text{BV}}(t+\Delta t)$ and $S_b^{\text{BF}}(t+\Delta t)$ for boundary particles b. For the last step, we compute current predicted volume errors $V^{\text{err}}(t+\Delta t)$, surface energy derivatives $-\frac{\partial}{\partial \mathbf{x}} E(t+\Delta t)$ and tangential relative velocities $\mathbf{v}^{\text{tang}}(t+\Delta t)$, which are then used in the actual Jacobi update step, as described in eq. (18), in order to obtain the updated guesses for pressure p, surface tension $\mathbf{F}^{\text{ST}}$ and friction $\mathbf{F}^{\text{F}}$.

```
1  $\mathbf{p} \leftarrow \mathbf{0}$
2  $\mathbf{F}^{\text{ST}} \leftarrow \mathbf{0}$
3  $\mathbf{F}^{\text{F}} \leftarrow \mathbf{0}$
4  while not converged do
5      $\mathbf{p}, \mathbf{F}^{\text{ST}}, \mathbf{F}^{\text{F}} \leftarrow$ Jacobi ($\mathbf{p}, \mathbf{F}^{\text{ST}}, \mathbf{F}^{\text{F}}$)

6  Function Jacobi ($\mathbf{p}, \mathbf{F}^{\text{ST}}, \mathbf{F}^{\text{F}}$):
7      foreach fluid particle $f$ do
8          Compute $\mathbf{F}_f^{\text{P}}$                          ▷ eq. (4)
9          Compute $\mathbf{F}_f^{N}$                          ▷ eq. (14)
10         Compute $\mathbf{v}_f(t+\Delta t)$                 ▷ eq. (20)
11     foreach fluid particle $f$ do
12         Compute all $S_f(t+\Delta t)$              ▷ eqs. (23) and (24)
13     foreach boundary particle $b$ do
14         Compute all $S_b(t+\Delta t)$              ▷ eqs. (23) and (24)
15     foreach fluid particle $f$ do
16         Compute $V_f^{\text{err}}(t+\Delta t)$              ▷ eq. (21)
17         Compute $-\frac{\partial}{\partial \mathbf{x}_f}E(t+\Delta t)$        ▷ eq. (22)
18         Compute $\mathbf{v}_f^{\text{tang}}(t+\Delta t)$            ▷ eq. (26)
19         Update $p_f$                              ▷ eqs. (18a) and (19)
20         Update $\mathbf{F}_f^{\text{ST}}$                            ▷ eq. (18b)
21         Update $\mathbf{F}_f^{\text{F}}$                             ▷ eqs. (13b) and (18c)
22     return $\mathbf{p}, \mathbf{F}^{\text{ST}}, \mathbf{F}^{\text{F}}$
```

Algorithm 1: The projected relaxed Jacobi solver loop to simultaneously solve for all pressures $\mathbf{p}$, surface tension $\mathbf{F}^{\text{ST}}$ and friction forces $\mathbf{F}^{\text{F}}$. The function Jacobi performs one Jacobi iteration. As such, it is the basic building block of the Jacobi solver presented here. Note that in line 12, $S_f(t+\Delta t)$ is computed for the fluid-vapor (FV), fluid-fluid (FF) and fluid-boundary (FB) interface. Similarly, in line 14, $S_b(t+\Delta t)$ is computed for both the boundary-vapor (BV) and the boundary-fluid (BF) interface.

4.3.1 Convergence criterion. The Jacobi solver described in algorithm 1 iterates until a user defined convergence criterion is reached. In order to ensure that values for pressure, surface tension and friction forces have converged to a satisfying degree, we propose to take into account the averaged residual errors of all forces:

$$\begin{aligned} 0.1\,\% \cdot \text{num. fluid particles} \geq & \sum_f \max\left(0, -V_f^{\text{err}}\right) \\ & + \sum_f |\mathbf{F}_f^{\text{ST}} + \frac{\partial}{\partial \mathbf{x}_f} E(t+\Delta t)| \\ & + \sum_f |\mathbf{F}_f^{\text{F}} - \text{prox}_{\mathcal{F}_f}\left(\mathbf{F}_f^{\text{F}} + \mathbf{F}_f^{*}\right)|. \end{aligned} \tag{27}$$

The clamped contributions of volume errors V^{err} and the threshold of 0.1 % are inspired by Ihmsen et al. [2014a].

4.4 NNCG

The solver described in this section is able to robustly compute unknown pressure, surface tension and friction. However, Jacobi solvers are well known for their slow convergence speed. In order to improve convergence behavior, Silcowitz-Hansen et al. [2010] proposed a *nonsmooth nonlinear conjugate gradient* (NNCG) method that acts on top of Jacobi iterates such as the one described in algorithm 1. Both, Erleben [2017] and Probst and Teschner [2023] suggest that significant speedups can be achieved this way compared to basic Gauss-Seidel and Jacobi solvers. Motivated by their recommendation, in appendix B we describe how the NNCG algorithm can be used to accelerate our solving procedure.

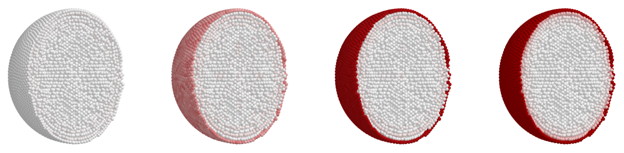
5 Calibration

We use this section to discuss remaining details concerning values given to parameters. Section 5.1 discusses the effects of surface tension kernel support $\hbar$ and the interface smoothing coefficient ϵ onto the surface detection. Next, in Section 5.2 we analyze the necessity to define rest areas A^0 that scale the estimated interface areas A and as such influence the strength of surface tension forces.

5.1 Interface Detection

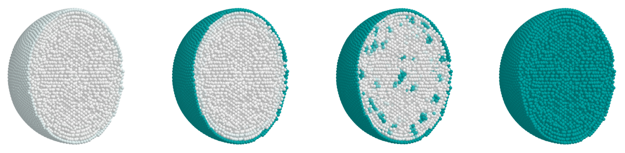
Our surface force computation as shown in eq. (11) heavily relies on the estimation of S as it governs how strong particles attract and repulse each other due to surface tension forces. In contrast to the pairwise model [Akinci et al. 2013; Becker and Teschner 2007; Jeske et al. 2023], this allows us to distinguish between particles at the interface to other phases and particles inside a fluid body, which in turn makes it possible for us to formulate physically meaningful surface tension forces that only act at the actual interface. The estimation of S is a two-step process:

5.1.1 Surface Tension Kernel Support. First, for all particles f and interfaces we estimate how much f contributes to the respective interface using eq. (8). Here, the relation between the employed kernel support radius in the pressure solver that ensures incompressibility and the kernel radius used in the approximation of the interface contributions C_f is important to consider. As fig. 4 shows, the surface tension kernel radius must be larger than the radius used in the pressure solver in order to reliably detect surface particles. Otherwise, we can see in fig. 4a that for equal kernel support radii, the number density estimation from eq. (8) would return one for particles at the surface, which is indistinguishable from particles inside the fluid body. Following Ihmsen et al. [2014a], in all our simulations we use kernel radius $\hbar = 2h$ with particle spacing h inside SPH approximations that are part of the pressure solver (eqs. (4) and (21)). Thus, for a robust detection of surface particles, we found that using a kernel radius of at least $3h$ is recommended. Any radius larger than $3h$ is still suitable to detect surface particles as shown in fig. 4d, but a larger number of particles is considered part of the surface, smoothing the otherwise sharply defined interface. Even more important, increasing the kernel support to values above $3h$ severely



(a) $\hbar^{\mathrm{ST}} = 2h$ (b) $\hbar^{\mathrm{ST}} = 2.5h$ (c) $\hbar^{\mathrm{ST}} = 3h$ (d) $\hbar^{\mathrm{ST}} = 4h$

Fig. 4. A cutaway view onto a fluid droplet. We visualize the magnitude of detected surface contributions C depending on the surface tension kernel support radius $\hbar^{\mathrm{ST}}$. White particles have small C, red particles have larger C. In this example, the volume and pressure computation use a kernel support radius of $2h$ with particle spacing h. We can see that in order for the surface detection to work, a kernel support larger than the one used for the volume estimation is required. Otherwise, surface particles are not able to detect being part of the surface since the estimated number density used in the surface detection (compare eq. (8)) is close or equal to one. The surface detection is already working quite well when using $\hbar^{\mathrm{ST}} = 2.5h$ for the surface tension kernel, except for light noise in the estimation of C at the surface and inside the fluid body. In our experiments we found that using both $\hbar^{\mathrm{ST}} = 3h$ or $\hbar^{\mathrm{ST}} = 4h$ results in a robust and reliable detection of surface particles. To keep the required computational effort to a minimum, in all our simulations we chose $\hbar^{\mathrm{ST}} = 3h$ as the surface tension kernel support radius.



(a) $\epsilon = 0.5$ (b) $\epsilon = 0.05$ (c) $\epsilon = 0.01$ (d) pairwise

Fig. 5. A cutaway view onto a fluid droplet to demonstrate the effect of the interface smoothing strength ϵ. We color the magnitude of S_f^{FV} which represents how strong the surface tension forces are pulling neighboring particles towards particle f. White particles f have $S_f^{\mathrm{FV}} = 0$, cyan particles have $S_f^{\mathrm{FV}} = 1$. We can see that for $\epsilon = 0.5$, the magnitude of S_f^{FV} of particles at the free surface is barely greater than zero. For $\epsilon = 0.05$ we get the desired result, surface particles are reliably detected while particles inside the fluid body have S_f^{FV} close to zero. For $\epsilon = 0.01$ the interface detection is too sensitive, small deviations from rest density inside the fluid are interpreted as an interface causing unwanted surface tension forces inside the fluid body. As a reference, fig. 5d illustrates that in the pairwise surface tension model all particles, including the ones inside the fluid body, are pulling onto each other. This is equivalent to setting $S_f^{\mathrm{FV}} = 1$ for all fluid particles f, and gives rise to nonphysical surface forces inside the fluid body.

impacts simulation performance as neighborhood list sizes grow cubically with kernel support $\hbar$. We want to point out that in contrast to Akinci et al. [2013] and Jeske et al. [2023], our surface tension computation does not rely on highly-specialized kernel functions that were specifically designed to prevent particle clustering. All our simulations were carried out using the default cubic spline kernel function [Monaghan 1992] for SPH interpolations and gradient approximations. Additionally, our surface tension formulation does not require any modifications to the pressure computation such as employed in He et al. [2015b] to counteract tensile instability.

5.1.2 Interface Smoothing. With the estimated surface contributions C for all fluid and boundary particles and for all respective interfaces at hand, we use eq. (12) to calculate S. As we can see in fig. 3, the parameter ϵ has great influence on how fast S approaches one with growing C. Since S is used in eq. (11) to compute surface tension forces, ϵ governs how easily a particle is considered to be part of the interface depending on its respective interface contribution C. Figure 5 visualizes this effect. For high smoothing strength ϵ such as in fig. 5a, particles require a large estimation of C in order to exert relevant surface tension forces. At the same time however, light noise in the estimation of interface contributions at particles inside the fluid body resulting in $C > 0$ is smoothed away and does not cause unwanted surface tension forces. In contrast, as displayed in fig. 5c, smaller values for ϵ allow interface particles to exert adequate surface tension, but small values of C due to noise in the SPH estimation could already be enough to consider the particle as part of an interface, giving rise to unwanted tension forces inside the fluid body. In our tests we found that using $\epsilon = 0.05$ reliably eliminates noise inside the fluid body while resulting in reasonable forces at the fluid interface. As a comparison, in fig. 5d S is set to one for all fluid particles, equivalent to the pairwise surface tension model where all neighboring particles are attracting each other, independent of whether they are part of an interface or not. Following eq. (11), this causes surface tension forces inside the fluid body which, as we will demonstrate in section 6.2, lead to nonphysical behavior of the droplet.

5.2 Rest Areas

Given the information from section 5.1, we know how to reliably detect particles belonging to an interface between different phases in the simulation. However, as illustrated in figs. 4 and 5, the sharpness and thickness of the interface can be defined with some degrees of freedom. Kernel support $\hbar^{\mathrm{ST}}$ and smoothing strength ϵ both influence the estimated values for S and as such have direct impact on the magnitude of computed surface tension forces. Fortunately, given a configuration of the parameters $\hbar^{\mathrm{ST}}$ and ϵ, it is easy to renormalize forces such that our simulated surface tension remains physically consistent. For this, rest areas A^0 introduced in eq. (10) are defined in such a way that spherical fluid droplets experience the right amount of surface tension. One can think of A^0 as the interface area a particle spans when it is completely surrounded by another phase. Since the interface area is directly proportional to the interface energy, with A^0 we can scale the maximum amount of interface energy a single particle can cause. We recommend using a kernel support radii $\hbar^{\mathrm{P}}$ and $\hbar^{\mathrm{F}}$ equal to $2h$ for the friction and pressure computation, $\hbar^{\mathrm{ST}} = 3h$ for surface tension and $\epsilon = 0.05$. With this configuration, we performed a calibration of A^0 through measurements of surface tension strength in a simulation and found that choosing $A^0 = \frac{\pi}{4}h^2$ results in the correct amount of surface tension for all curvatures in three dimensions. By adopting this value for A^0, we define that a single particle surrounded by another phase spans a surface area equivalent to a sphere with radius $\frac{h}{4}$. For a two-dimensional setting we recommend using $A^0 = \frac{1}{7}h$. Note that while A^0 indeed should be reevaluated when $\hbar$ or ϵ are altered, the process of doing so is rather simple: one measures the average pressure inside a spherical droplet and compares it to the analytical pressure as shown in section 6.2. Rest areas A^0 are then scaled according to the relation between measured and expected pressure. We also like to mention that our recommended parameter configuration works well for all scenarios displayed in section 6, which is why we speculate that there is actually rarely a need to reevaluate A^0.

6 Results

In this section, we want to demonstrate the validity, correctness and versatility of our solver and the forces calculated by it. In section 6.1, we start with some well-known scenarios that have been repeatedly shown in earlier work to visualize the effects of surface tension. Next, in section 6.2 we analyze the strength of surface tension forces resulting from different surface tension formulations and compare them with analytic results. In section 6.3, we present scenarios that showcase our new fluid friction formulation. In order to indicate that a unified force computation has advantages over separated approaches, in section 6.4 we compare different solver set-ups. Solver performance is evaluated in section 6.5. Last, section 6.6 displays a selection of simulation scenarios that outline the capabilities of our proposed simulation method.

6.1 Visual Evaluation

Previous works on particle-based surface tension formulations [Akinci et al. 2013; Becker and Teschner 2007; He et al. 2015b; Hyde et al. 2020; Jeske et al. 2023; Yang et al. 2016, 2017] often use similar scenarios to show that their methods produce plausible surface tension effects. Here, we display simulation results of some of those basic scenarios using our solver.

6.1.1 Crown, Chain, Bell and Sphere. Starting with a water crown, fig. 6 shows that our surface tension can handle thin sheets of fluid and produces the expected secondary upward splash due to water refilling the gap caused by the initial impact of the droplet on the fluid surface. Inspired by Jeske et al. [2023], we also present a water chain and a water bell in figs. 7 and 8. In both scenarios, the colliding water jets first broaden the surface of the fluid stream. Surface tension forces act towards a minimal surface area. As such, they pull the fluid stream back together. In fig. 7, the oscillating stream thickness loosely resembles the outline of a chain. Another well known test setting consists of a water jet colliding with a sphere as shown in fig. 9. With strong enough surface tension forces, the fluid is able to flow around the sphere and merges into a single stream below. Note that after the fluid stream stops, droplets are able to stick to the sphere even at inclined surfaces due to our fluid-solid friction force. In all scenarios, we use the viscosity formulation proposed by Weiler et al. [2018].

6.1.2 Two Dimensions. Our surface tension formulation can be directly employed in two-dimensional settings. To demonstrate this, we simulate a two-dimensional Rayleigh instability with low and high interface tension γ^{FF}. The results are visualized in fig. 10. As we can see in fig. 10b, compared to low γ^{FF}, the high interface tension forces are able to notably minimize the fluid-fluid contact areas, causing much smoother interfaces with less detail.

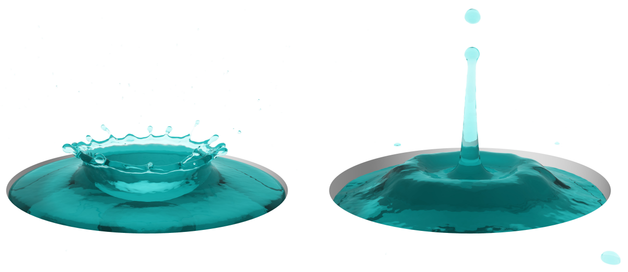


Fig. 6. A droplet falls into water, the splashes form a water crown. Our surface tension is able to replicate the thin water sheets at the side of the crown. The fluid body has a diameter of 0.08 m and $\gamma^{\mathrm{FV}} = 0.01\,\mathrm{N\,m^{-1}}$.

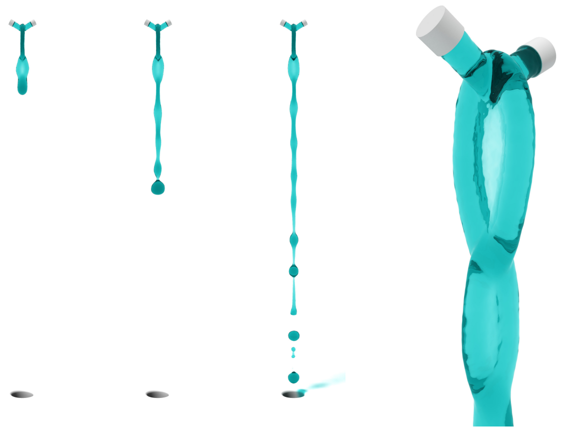


Fig. 7. A water chain is forming due to two water jets colliding midair. We chose $\gamma^{\mathrm{FV}} = 0.25\,\mathrm{N\,m^{-1}}$, the distance between source and sink is 5 m.

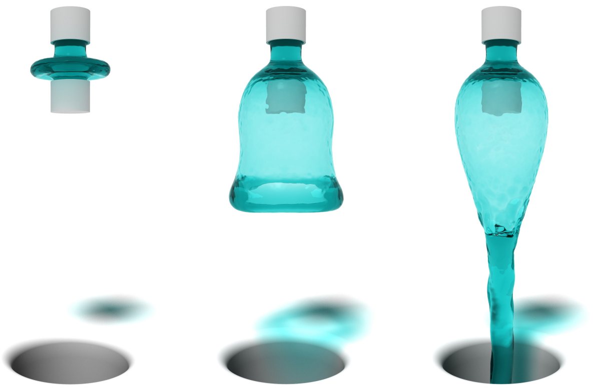


Fig. 8. Two vertical water jets are colliding. Due to surface tension, the water that was initially pushed outwards is pulled together until it merges into a single water stream. The fluid has $\gamma^{\mathrm{FV}} = 0.16\,\mathrm{N\,m^{-1}}$, the distance between fluid collision and sink is 2 m.

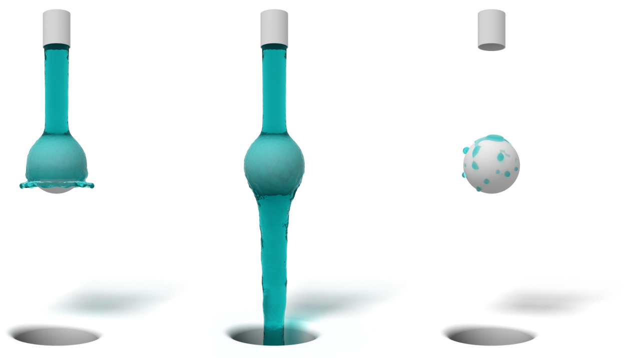


Fig. 9. A water stream hits a sphere. Due to surface tension forces, the water stream is able to flow around the sphere and merge below. Note that after we stop the stream, droplets are able to stick to the sphere at sloped surfaces due to our friction force. The sphere has a diameter of 0.2 m and we set $\gamma = 0.35\,\mathrm{N\,m^{-1}}$.

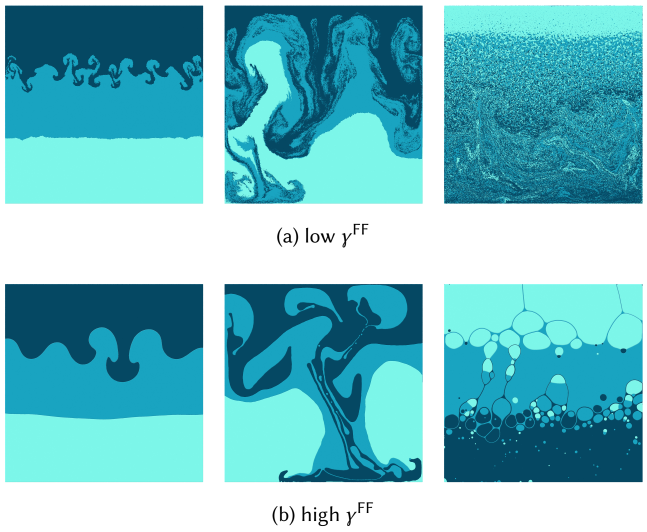


Fig. 10. Rayleigh-Taylor instability of three fluids as an exemplary two-dimensional fluid simulation including surface tension. With low surface tension the fluids easily mix as we can see in fig. 10a. Due to higher surface tension, the fluid interface shapes in fig. 10b are much smoother.

6.1.3 Capillary Action. Less often shown in existing work, even though its cause can be attributed to surface tension, is capillary action. If the surface energy density at the solid-vapor interface γ^{BV} is bigger than the surface energy density at the solid-fluid interface γ^{BF}, fluid is pulled into narrow spaces due to the surface tension pushing towards an energetically more favorable state. The thinner the gap, the higher the distance the fluid can travel until gravitational forces balance surface tension forces. As we can see in fig. 11, our surface tension formulation is able to replicate this effect. Fluid pillars rise in the capillaries, and we observe higher pillars the thinner the capillaries are. Additionally, we want to mention that in this scenario the parametrization of our surface tension formulation is intuitive and values for γ^{BV}, γ^{BF} and γ^{FV} can be directly looked up in literature (e.g. Speight [2017]). This is in contrast to many other approaches where surface tension parametrization has either no direct relation to actual physical forces [Akinci et al. 2013; Becker and Teschner 2007; Jeske et al. 2023], or parameters must be newly translated for each simulation setting to achieve the desired forces [He et al. 2015b]. We discuss this further in section 6.2.

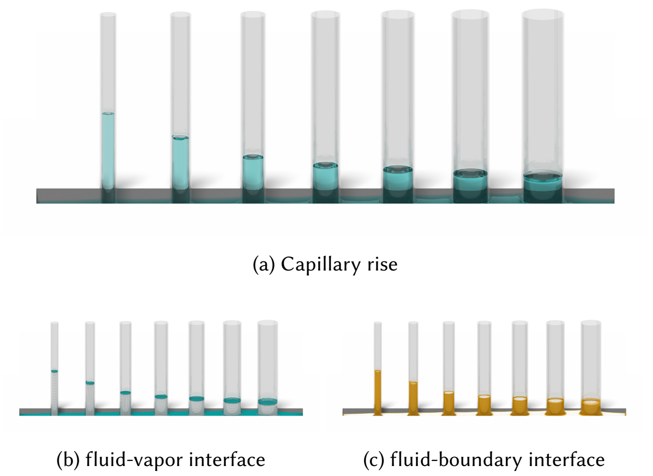


Fig. 11. Fluid rises upwards in 0.15 m high capillaries due to a boundary-vapor surface energy density $\gamma^{BV} = 0.04\,\mathrm{N\,m^{-1}}$ that is larger than the boundary-fluid surface energy density $\gamma^{BF} = 0\,\mathrm{N\,m^{-1}}$. The fluid pillars grow until gravitational forces match the boundary surface tension forces. In capillaries with smaller radius, this allows the fluid to rise higher compared to fluid in thicker capillaries. Figure 11b displays the underlying particle representation and the fluid-vapor interface detection. Particles colored in cyan have S^{FV} close to one and as such exert fluid-vapor surface tension forces onto neighboring particles, particles in white have small S^{FV}. Similarly, fig. 11c visualizes the fluid-boundary interface detection. Here, particles colored yellow have S^{FB} close to one. Note that particles at the edge between fluid, boundary and vapor can be part of the fluid-vapor and the fluid-boundary interface simultaneously.

6.1.4 Parametrization. To get a better feeling for the surface tension parameters γ^{FV}, γ^{FF}, γ^{FB}, γ^{BV}, γ^{BF}, we present two scenarios where we focus on the individual effects of some selected parameter configurations. Four tumbling boards are displayed in fig. 12. At the very top of each board, droplets are spawned whereby each droplet is considered as an individual fluid. The blue fluid droplets in the left board are given no surface tension at all. As such, they are able to merge into each other and splash when hitting an obstacle on their way downwards. The green droplets have a positive fluid-vapor surface energy density γ^{FV}. As such, surface tension tries to minimize the fluid-vapor interface area. This causes the droplets to stick to each other and to the boundary. For the third configuration, we now also increase the fluid-boundary interface energy density γ^{FB}. The orange droplets are still minimizing their fluid-vapor surface because of $\gamma^{FV} > 0$, but they no longer stick to the boundary since creating fluid-boundary interfaces now requires some potential energy too. The red droplets are additionally given a non-zero fluid-fluid interface energy density γ^{FF}. This way, droplets do not merge anymore since surface tension forces work against the formation of fluid-fluid interfaces. Each droplet now tries to minimize its own surface due to γ^{FV}, but without creating interfaces to other fluids and the boundary. We can see that our surface tension parametrization is able to produce vastly different fluid behavior while still being quite intuitive to use.

Next, we take a closer look at the two remaining parameters γ^{BV} and γ^{BF}. A droplet resting on a solid creates some contact angle θ between the solid surface and the fluid surface. The Young equation relates θ to surface energy densities γ^{FV}, γ^{BV} and γ^{BF} with [Young 1805]

$$\cos\theta = \frac{\gamma^{\mathrm{BV}} - \gamma^{\mathrm{BF}}}{\gamma^{\mathrm{FV}}}. \tag{28}$$

Figure 13 visualizes the droplet shape for three different parameter configurations. In all configurations we have $\gamma^{\mathrm{FV}} > 0$ so that the fluid behaves like an actual droplet which tends to reduce its free surface area. For the first parameter setting, we choose $\gamma^{\mathrm{FV}} = \gamma^{\mathrm{BF}} > 0$ and $\gamma^{\mathrm{BV}} = 0$. In this case, spanning a fluid-vapor surface costs as much energy as a boundary-fluid interface, so surface tension forces minimize the sum of fluid-vapor and boundary-fluid interface area. As a result, the droplet forms a sphere that rests on the solid with contact angle θ approaching 180°, as predicted by the Young equation. In the second setting, the boundary-fluid interface energy density γ^{BF} is reduced to zero. Forming a fluid-solid interface now costs no energy causing the surface tension forces to minimize the fluid-vapor interface area only. The Young equation predicts a contact angle of 90°, which is in good accordance to the observed simulation result in fig. 13b. As the last configuration, we now increase the boundary-vapor surface energy density γ^{BV} to a value slightly bigger than γ^{FV}. Surface tension forces push to minimize the boundary-vapor surface area and since $\gamma^{\mathrm{BV}} > \gamma^{\mathrm{FV}}$ this force component is stronger than the forces trying to minimize the fluid-vapor interface area. In fig. 13c we can see that our surface tension formulation is able to replicate the expected complete wetting of a surface up until the fluid sheet is only one particle thick. In the last two figs. 13d and 13e we reverse the parameter changes. However, due to our employed friction forces, the droplet is not able to pull itself together in one piece. Instead, it breaks up into smaller droplets which each on their own exhibit the expected contact angles. Existing surface tension and adhesion approaches often neglect the distinction between boundary-fluid and boundary-vapor interfaces [Akinci et al. 2013; Becker and Teschner 2007; He et al. 2015b; Jeske et al. 2023; Morris 2000; Müller et al. 2003; Wang et al. 2017]. This way, they are not able to define droplet behavior at the boundary through a physically meaningful parametrization of γ^{BF} and γ^{BV} as we do in fig. 13.

6.2 Analytical Evaluation

In the previous section we validated our experiments by eyeballing the results and decided that they match the expected outcome. Now, we want to measure the strength of surface tension forces depending on the curvature of a droplet. Goal is to demonstrate that our surface tension solver produces the right amount of surface tension for all curvatures. We compare our formulation to a pairwise approach as employed in Tartakovsky and Meakin [2005], Becker and Teschner [2007], Jeske et al. [2023], and Yang et al. [2016, 2017] and the robust CSF formulation proposed by He et al. [2015b]. Similar to Huber et al. [2015], we do not consider the additional air pressure, two scale pressure estimation and anisotropic pressure filtering that was also introduced in He et al. [2015b]. Instead, we found that using a kernel support radius of $3h$ alleviates any problems concerning tensile instability. Since surface tension coefficients are not directly comparable between existing surface tension formulations [Akinci et al. 2013; He et al. 2015b; Huber et al. 2015; Jeske et al. 2023], for the pairwise surface tension model and the robust CSF formulation we chose the coefficients such that they match the desired result best. In this experiment, we measure the average pressure inside a spherical droplet. The theoretical relation between surface tension γ^{FV} and pressure p derived from the Young-Laplace equation [Gennes et al.

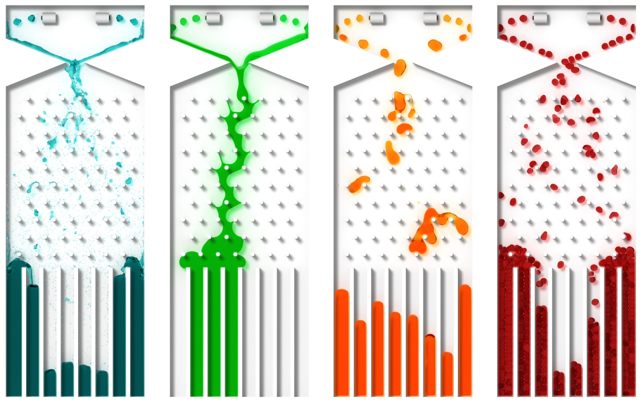
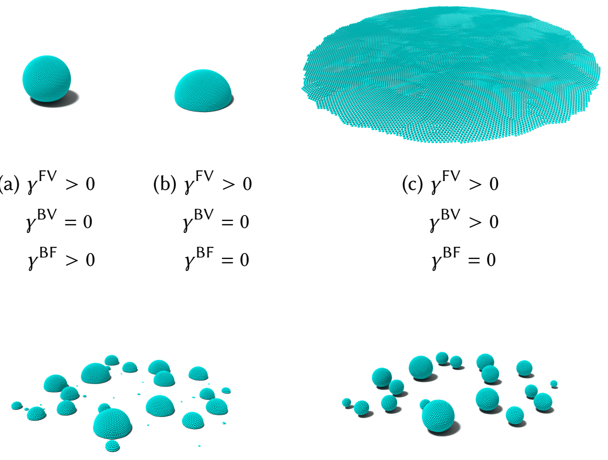

(a) $\gamma^{\mathrm{FV}} = 0$ $\gamma^{\mathrm{FF}} = 0$ $\gamma^{\mathrm{FB}} = 0$ (b) $\gamma^{\mathrm{FV}} > 0$ $\gamma^{\mathrm{FF}} = 0$ $\gamma^{\mathrm{FB}} = 0$ (c) $\gamma^{\mathrm{FV}} > 0$ $\gamma^{\mathrm{FF}} = 0$ $\gamma^{\mathrm{FB}} > 0$ (d) $\gamma^{\mathrm{FV}} > 0$ $\gamma^{\mathrm{FF}} > 0$ $\gamma^{\mathrm{FB}} > 0$

Fig. 12. Droplets tumble down a board with obstacles. To demonstrate the versatility of fluid behavior that is possible with our surface tension formulation, we simulate four different parameter settings. On the left, fluid-vapor surface energy density γ^{FV}, fluid-fluid interface energy density γ^{FF} and fluid-boundary interface energy density γ^{FB} are set equal to zero. The result is a homogeneous fluid that shows no surface tension effects. Second from left, we increase γ^{FV} while γ^{FF} and γ^{FB} remain zero. Thus, the droplets tend to minimize their free surface by merging with each other and sticking to the boundary. For the third configuration, we now also increase γ^{FB} so that γ^{FF} is the only interface energy density that remains zero. We can see that droplets still merge with each other, but due to the increased energy density at the fluid-boundary interface, droplets no longer stick to the boundary. In the rightmost configuration, to prevent droplets from merging with each other, we increase γ^{FF}. Individual droplets now bounce off each other since fluid-fluid interfaces hold potential energy. The tumbling board model was created by iolalla.


(a) $\gamma^{\mathrm{FV}} > 0$, $\gamma^{\mathrm{BV}} = 0$, $\gamma^{\mathrm{BF}} > 0$ (b) $\gamma^{\mathrm{FV}} > 0$, $\gamma^{\mathrm{BV}} = 0$, $\gamma^{\mathrm{BF}} = 0$ (c) $\gamma^{\mathrm{FV}} > 0$, $\gamma^{\mathrm{BV}} > 0$, $\gamma^{\mathrm{BF}} = 0$

(d) $\gamma^{\mathrm{FV}} > 0$, $\gamma^{\mathrm{BV}} = 0$, $\gamma^{\mathrm{BF}} = 0$ (e) $\gamma^{\mathrm{FV}} > 0$, $\gamma^{\mathrm{BV}} = 0$, $\gamma^{\mathrm{BF}} > 0$

Fig. 13. The contact angles between a fluid droplet and the boundary depends on the surface energy density parameter configuration. We start this scenario with a hydrophobic surface that has a boundary-fluid interface energy density γ^{BF} much higher than the boundary-vapor interface energy density γ^{BV}. In this configuration the droplet tends to minimize the boundary-fluid interface area, resulting in the desired hydrophobic material behavior. Next, in fig. 13b we equalize γ^{BF} and γ^{BV}. The droplet can now freely minimize its free surface, and as such develops a half-spherical shape. In fig. 13c we increase γ^{BV} to a value bigger than γ^{FV}. Now, a complete wetting of the surface is the energetically most favorable state. Our surface tension formulation is able to replicate the wetting by deforming the droplet into a fluid sheet with a thickness of a single particle. We then reverse the parameter configuration by setting γ^{BV} back to zero. Same as in fig. 13b, in fig. 13d the droplet minimizes its free surface. However, due to friction forces the thin fluid sheet is not able to contract into one droplet, but instead breaks apart into multiple smaller ones. Last, fig. 13e shows the smaller droplets resting on the original hydrophobic surface material.

2004] is given by

$$p = 2\frac{\gamma^{\mathrm{FV}}}{r} \tag{29}$$

with droplet radius r. As we see, the smaller r is, the larger the pressure p inside the droplet. In fig. 14, the measured average pressure is plotted for a range of droplet radii r. We can see that the pairwise approach causes an average pressure inside the fluid droplets that is approximately constant for all radii r. For small droplets consisting out of very few particles, the surface tension forces even reduce in magnitude. We suspect that this is caused by insufficiently filled particle neighborhoods. For a constant particle size as shown in fig. 14a and an appropriate parameter setting, our implementation of robust CSF [He et al. 2015b] is able to match the expected pressure quite well. However, when the droplet radius is changed by varying the particle size as done in fig. 14b, the surface tension parameter employed in robust CSF [He et al. 2015b] needs to be reevaluated for each particle resolution. Otherwise, surface tension forces scale incorrectly. We want to mention that He et al. [2015b] seem to be aware of this as they titled their parameter that scales surface tension forces a *squared gradient energy coefficient* with unit N instead of the commonly used $[\gamma] = \mathrm{N\,m^{-1}} = \mathrm{J\,m^{-2}}$. In both constellations, our surface tension formulation matches the desired pressure. Only for small droplets made out of few particles, in fig. 14a we observe a slight underestimation of surface tension forces. We believe this is caused by an insufficient number of neighboring particles, similar to the pairwise approach.

6.2.1 Visual Example. Even though pairwise approaches do not scale surface tension forces correctly, earlier work shows that they usually can produce visually satisfying and expected surface tension effects [Akinci et al. 2013; Huber et al. 2015; Jeske et al. 2023; Tartakovsky and Meakin 2005; Yang et al. 2016, 2017]. Here, we want to give a counterexample to demonstrate a case where pairwise surface tension simulations produce a different result compared to a surface tension force that correctly scale with curvature. For this, we consider two droplets of different sizes which are connected through a thin tube. Due to the pressure difference in the droplets, the smaller droplet should merge into the bigger one. As we can see in fig. 15, our surface tension formulation can replicate this dynamic while the pairwise approach struggles to.

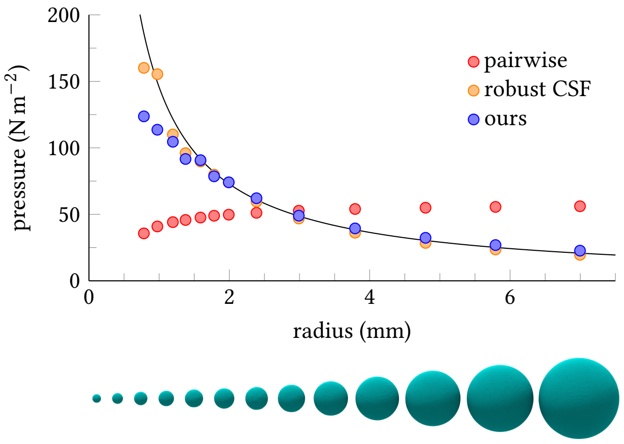
6.3 Friction

Our friction formulation at the fluid-boundary interface allows us to simulate droplets resting on inclined surfaces. As a first proof of concept, in fig. 16 we show a range of droplets resting on sloped planes. Over time, we reduce the coefficient of friction μ, causing the droplets to roll downwards. As one would expect, bigger droplets and steeper slopes require larger μ to allow the droplet to stick to the boundary.

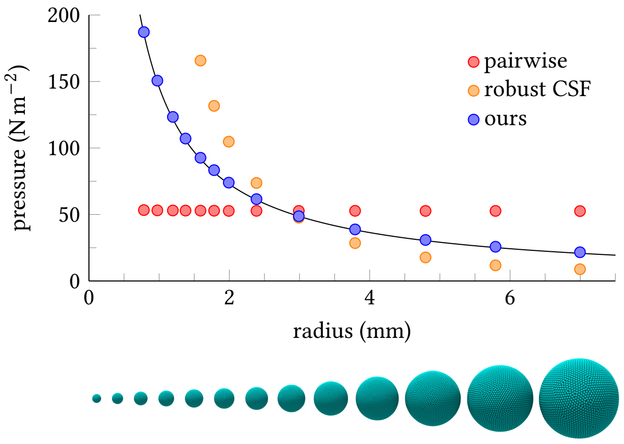
6.3.1 Window. To point up the effects of our friction force on the behavior of fluids in a more practical scenario, we simulate rain falling onto a window. As fig. 17 depicts, without friction forces, the small droplets simply slide down the window pane. When including friction forces, the droplets stick to the pane, allowing them to accumulate into bigger droplet. After reaching a critical mass, larger droplets are able to roll down the window. We can also observe that friction causes droplets to leave wet trails behind them, which mark preferred paths for other droplets to roll down the pane.

6.4 Unified Solver

In section 4 we repeatedly emphasized that in our simulation method pressure, friction and surface tension forces are solved in a unified manner such that they can be consistent with each other. Otherwise, constraints enforced by one solver, such as incompressibility or correct surface tension, may be violated by another solver [Zhang et al. 2012]. The benefits of a combined pressure and friction computation are already discussed in Probst and Teschner [2023]. For us, it is more interesting to analyze the interplay between the pressure and surface tension solver as pressure and surface tension forces counteract each other to a large degree. In fig. 18 we simulate a droplet under gravity that lies on a flat surface. Figures 18a and 18b

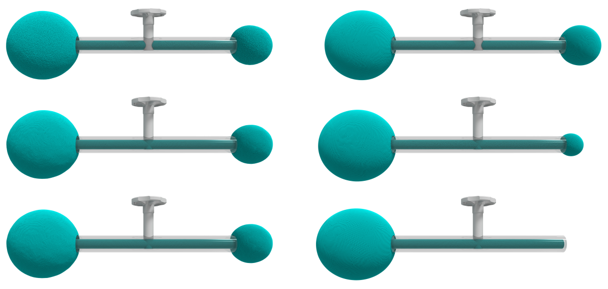


(a) constant particle size, increasing number of particles



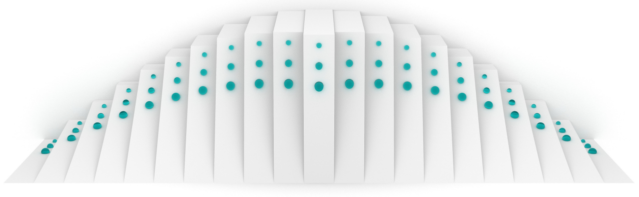
(b) increasing particle size, constant number of particles

Fig. 14. The average pressure inside a spherical droplet with $\gamma^{\mathrm{FV}} = 0.072\,\mathrm{N\,m^{-1}}$ depending on its radius. Pressure counteracts the surface tension forces acting at the droplet and as such can be used as a measuring tool for the surface tension strength. The black line represents the expected pressure $p = 2\gamma^{\mathrm{FV}}/r$ with droplet radius r. Smaller droplets have a higher average pressure than larger droplets. As we can see, pairwise surface tension formulations such as employed in Becker and Teschner [2007], Akinci et al. [2013] and Jeske et al. [2023] fail to capture this property, average pressure is approximately constant for all r. Additionally, as demonstrated in fig. 14a, for droplets made out of few particles only, the pairwise approach can suffer from missing particle neighbor contributions causing weaker surface tension forces. For a constant particle size (fig. 14a) and an appropriate parameter setting, our implementation of robust CSF [He et al. 2015b] is able to match the analytic pressure inside the droplets quite well for all radii r. However, when varying particle size (fig. 14b), the surface tension parametrization needs to be adapted to each particle resolution, otherwise surface tension forces scale incorrectly with droplet radius r as we can see in fig. 14b. Our surface tension formulation matches the desired pressure in both constellations. Only for small droplets consisting out of very few particles, in fig. 14a we observe a slight underestimation of surface tension forces. Similar to the pairwise approach, we suspect that this is caused by an insufficient number of neighboring particles.

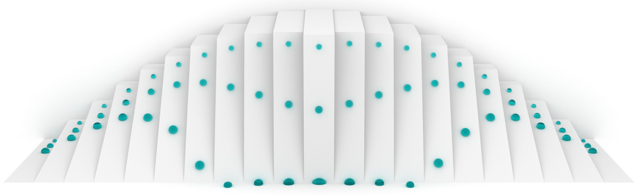


(a) pairwise (b) ours

Fig. 15. Two droplets of different sizes are connected after the valve wheel is turned open. Due to higher internal pressure, the smaller droplet is expected to merge into the larger one. However, independent of surface curvature, pairwise surface tension approaches produce equal pressure in both droplets, as already illustrated in fig. 14. Thus, on the left-hand side we can see that the fluid inside the droplets is not accelerated in either direction. In contrast, using our surface tension formulation, the smaller droplet experiences higher surface tension forces due to higher curvature, and its fluid is correctly pushed into the larger droplet. We set $\gamma^{\mathrm{FV}} = 1\,\mathrm{N\,m^{-1}}$, the bigger droplet has a diameter of 0.02 m. The surface tension coefficient for the pairwise approach was chosen such that the average pressure inside the fluid is approximately the same as in our approach. The valve wheel model was created by JohnEdwa.

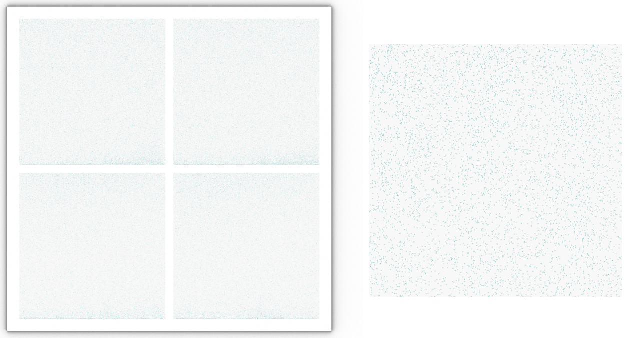


(a) $\mu = 0.05$

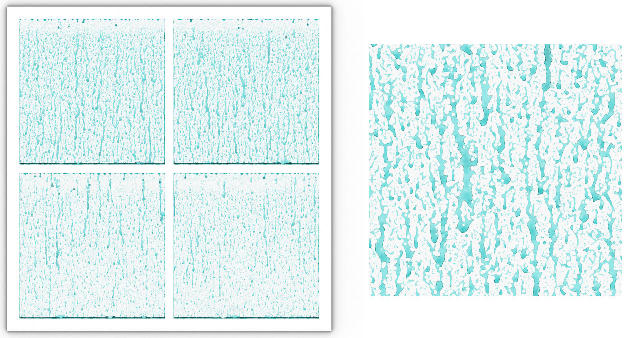


(b) $\mu = 0.02$

Fig. 16. Droplets of different sizes are able to rest on a range of slopes due to friction between the droplets and the underlying surface. The outer surfaces are placed horizontally, the slope in the middle is vertical. All droplets are given the same surface tension parametrization $\gamma^{\mathrm{FV}} = 4\,\mathrm{N\,m^{-1}}$, $\gamma^{\mathrm{FF}} = 0\,\mathrm{N\,m^{-1}}$, and $\gamma^{\mathrm{FB}} = 0.5\,\mathrm{N\,m^{-1}}$ and coefficient of friction μ. The bigger droplets have a diameter of approximately 9 mm, the middle ones are 7 mm and the smallest droplets on the top measure 5 mm. Over time, the value of μ is continuously reduced. We observe that larger droplets on steeper slopes are the first to roll down the ramp. Smaller droplets experience less gravitational pull and greater surface tension forces at the same time, allowing them to stick to the sloped plane for a longer time.

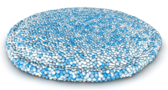


(a) no friction

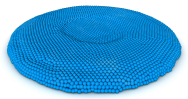


(b) with friction

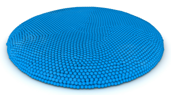
Fig. 17. Droplets with $1.5\,\mathrm{N\,m^{-1}}$ rain against a 1.5 m wide window. Without friction between the droplets and the window pane, the small droplets immediately slide down the pane. With friction, the droplets stick to the window and accumulate into bigger droplets over time. After reaching a critical mass, the droplets roll down the pane. In the accompanying video, we can observe that when droplets merge, the resulting bigger droplets are able to pick up more speed. Friction forces also cause droplets to leave wet trails behind them. Fluid covering the window allows subsequent fluid particles to flow on top of it, causing less frictional resistance compared to a direct fluid-window contact. As such, the wet trails are preferred paths for other droplets that roll down the window.



(a) friction and pressure before surface tension



(b) surface tension before friction and pressure



(c) surface tension, friction and pressure combined

Fig. 18. Comparison between different solver constellations. A droplet is placed on a flat surface where it should come to rest. Velocities are color coded with white particles having high velocity. In figs. 18a and 18b we used our surface tension solver, but separated it from the pressure and friction solver. We can clearly see in fig. 18a that when computing pressure and friction forces before surface tension forces, the fluid is not able to come to rest. A different problem arises when first computing surface tension forces as shown in fig. 18b. Here, surface tension forces are not able to shape the droplet as one would expect, instead particles seem to interlock and nonphysical dents remain in the droplet's surface. Only a combined computation of surface tension, pressure and friction produces the desired result.

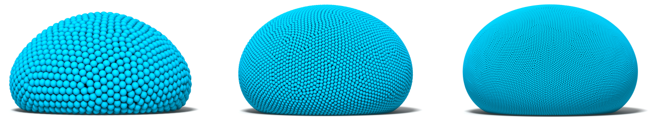


Fig. 19. Performance and scalability analysis by simulating a droplet resting on a flat surface. The droplet always has the same radius, but particle size h and timestep Δt is varied between measurements in table 3.

are the result of a simulation where we separated our surface tension solver from the pressure solver. We can see in fig. 18a, when computing surface tension after pressure forces, the surface tension disturbs the particle volume prediction the pressure solver relies on. As a consequence, the fluid particles stay in motion, and the droplet is not able to come to rest. In contrast, when computing pressure after surface tension is applied, the pressure forces seem to hinder the surface tension. While the droplet is now able to come to rest, unexpected dents remain in the droplet's surface. Only a combined computation of forces, as shown in fig. 18c, results in the desired droplet shape.

6.5 Solver Performance

This section further analyzes the scalability and convergence behavior of our solver. We again chose to simulate a droplet resting on a flat surface as our test scenario as visualized in fig. 19. The average required solver iterations and computation time per simulated second are measured and displayed in table 3 for a range of different timesteps Δt and particle resolutions h. We also distinguish between the Jacobi solver as shown in algorithm 1 and a solver version with NNCG acceleration as described in appendix B. As expected, the number of required Jacobi and NNCG iterations increase with larger timestep and higher particle count. Thus, the highest possible timestep does not always yield the best simulation performance. In most cases, the NNCG solver outperforms the Jacobi solver by requiring fewer iterations, which is especially significant for more demanding configurations. This is also reported by Probst and Teschner [2023] who compare the performance of Jacobi and NNCG for the computation of pressure forces. Only in some simple constellations, the NNCG solver causes some overhead in iteration count when compared the Jacobi solver.

Next, using the same test scenario, we measured the required computation times of the individual steps in our simulation method. The result is given in table 4. A major part of the time is spent inside the solver loop. This is expected for an implicit force solver with an average of 18 solver iterations. Within the loop, the update of V^{err}, $-\partial/\partial\mathbf{x}\,E$ and $\mathbf{v}^{\mathrm{tang}}$ requires the most computation time. In contrast,

Table 3. Average required solver iterations and computation time per simulated second for a simulation of a sessile droplet as shown in fig. 19. We measured solver performance for particle resolution h equal to 1.0 mm, 0.5 mm and 0.25 mm, timesteps Δt set to 0.04 ms, 0.08 ms and 0.12 ms, with and without NNCG acceleration. As expected, higher timesteps and more particles inside the droplet require higher iteration count. The biggest timestep is not always the best choice regarding simulation performance as an increased number of required solver iterations might outweigh larger timesteps. In most constellations, the NNCG solver is able to significantly reduce solver iterations, especially when considering more demanding configurations. Only in simple scenarios, the NNCG solver sometimes seems to cause some overhead in iteration count compared to the basic Jacobi solver.

			Jacobi		NNCG	
h	fluid particles	Δt	iterations	time per simulated s	iterations	time per simulated s
1.0 mm	4,100	0.04 ms	2	0.12 h	2	0.13 h
		0.08 ms	2	0.06 h	3	0.08 h
		0.12 ms	7	0.09 h	6	0.08 h
0.5 mm	33,300	0.04 ms	2	0.20 h	4	0.27 h
		0.08 ms	12	0.29 h	8	0.21 h
		0.12 ms	18	0.28 h	9	0.16 h
0.25 mm	267,700	0.04 ms	12	2.62 h	8	1.91 h
		0.08 ms	26	2.53 h	12	1.32 h
		0.12 ms	42	2.67 h	18	1.25 h

Table 4. Required computation times of the individual steps of our simulation method. The measurement was done for the same scenario shown in fig. 19 with 267,700 fluid particles and $\Delta t = 0.12$ ms. We employ an OpenCL implementation of the neighbor search proposed by Band et al. [2020] and the implicit viscosity solver from Weiler et al. [2018] running on an NVIDIA GeForce RTX 4090 graphics card. Our own solver is implemented on the GPU as well. As expected for an implicit method, a major part of the computation time is spent inside the solver loop. On average, the solver did 18 Jacobi iterations accelerated with NNCG. Within the loop, the update of V^{err}, $-\partial/\partial\mathbf{x}\, E$ and $\mathbf{v}^{\text{tang}}$ is the most involved. We can also see that little time is required for the NNCG update as no SPH interpolations are computed here. Thus, the small computational overhead due to the NNCG extension is easily compensated by its increased convergence speed, as already indicated in table 3.

	time per step	ratio
neighbors search	26 ms	4.4 %
gravity and viscosity	23 ms	3.9 %
diagonal elements α	45 ms	7.7 %
solver loop	487 ms	83.7 %
update $\mathbf{F}^{\text{P}}$, $\mathbf{F}^{N}$ and $\mathbf{v}$	125 ms	21.5 %
update S	111 ms	19.0 %
update V^{err}, $-\frac{\partial}{\partial \mathbf{x}}E$ and $\mathbf{v}^{\text{tang}}$	243 ms	41.7 %
update p, $\mathbf{F}^{\text{ST}}$ and $\mathbf{F}^{\text{F}}$	2 ms	0.3 %
NNCG update	7 ms	1.2 %
integrate $\mathbf{x}$	1 ms	0.2 %
total	581 ms	100 %

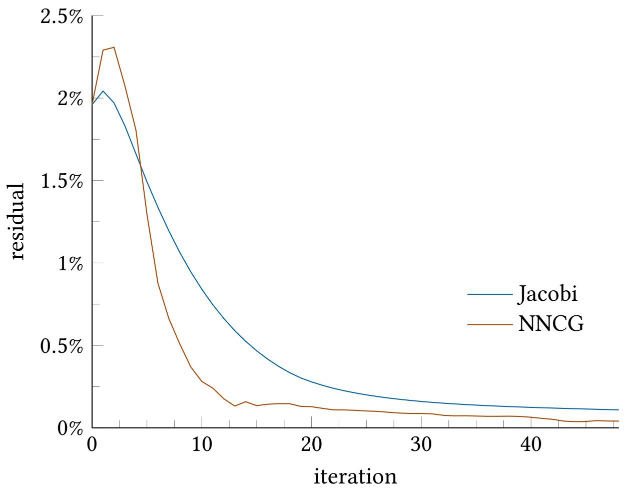


Fig. 20. Convergence behavior of the Jacobi solver and the NNCG solver. The same test configuration as in tables 3 and 5 and shown in fig. 19 is used here. As we can see, the NNCG solver converges faster compared to the Jacobi solver, but Jacobi iterates are converging in a smoother fashion. During the first few iterates the error initially grows for both solvers which can be explained by surface tension and pressure forces counteracting each other. Still, the solver is able to find a combined solution of forces, as we can see by the continuously decreasing residual afterward.

computations related to the NNCG acceleration are relatively cheap as no SPH interpolations have to be carried out here. As such, the small increase in computation time per solver iteration is usually overcompensated by the reduction of required iterations through NNCG.

Last, we compare the convergence behavior of the Jacobi solver with the NNCG accelerated version in fig. 20. Again, the sessile droplet was used as a test setting. We can see that the residual error initially increases in the first few solver iterations. This can be explained by the fact that newly computed pressure and surface tension forces counteract each other and drive up each other's residual. In subsequent iterations however, the forces converge towards a combined solution indicated by the continuously decreasing residual. The NNCG solver converges significantly faster towards a solution than the Jacobi solver. On the other hand, the Jacobi iterates seem to converge in a smoother, more predictable manner.

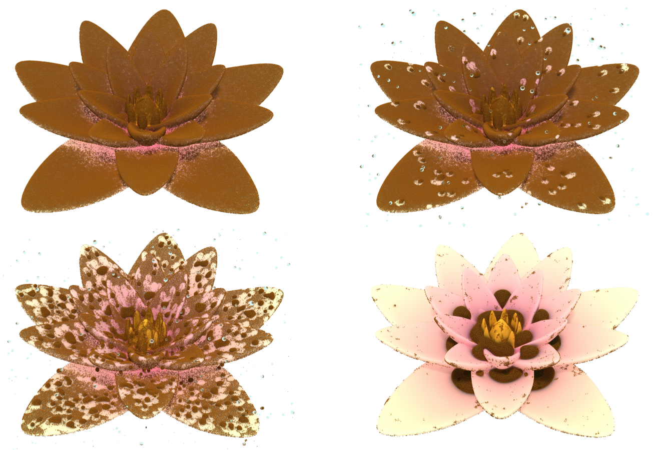
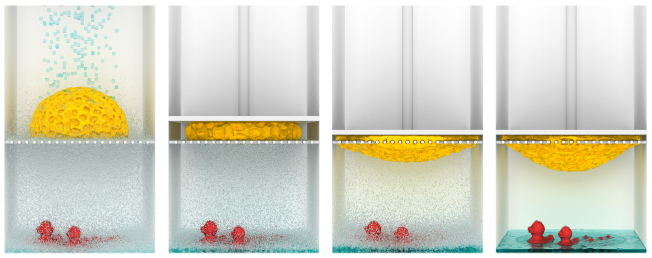
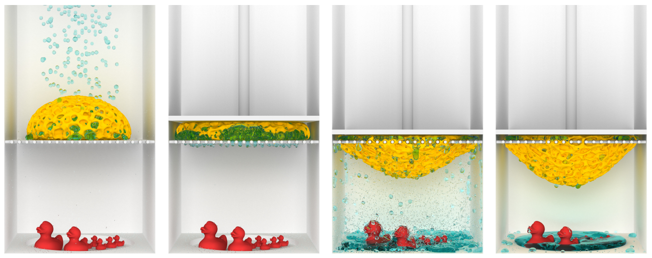


Fig. 21. A water lily is cleaned of dirt by rain. Initially, dirt particles cover the whole 0.2 m wide flower. When it starts to rain, water droplets fall onto the petals. Smaller droplets are able to stick due to the surrounding dirt, but as they grow larger they start to roll down the leaf, picking up any dirt particles in their way. This way, over time the flower is washed clean of the dirt. The particles representing dirt are given a relatively large coefficient of friction $\mu = 0.8$ and zero interface energy densities γ. The rain droplets have $\gamma^{\mathrm{FV}} = 0.14\,\mathrm{N\,m^{-1}}$ to model the surface tension and $\gamma^{\mathrm{FB}} = 0.28\,\mathrm{N\,m^{-1}}$ to replicate the hydrophobic property of the flower. The parameter γ^{FF} is set to zero to make sure that dirt particles are able to stick to the fluid droplets. With this simulation example we want to demonstrate how intuitive it is for a user of our simulation method to parametrize the surface tension and friction computation, even when considering a complex interplay between various materials and fluid phases. The water lily model was created by guppyk.



(a) no surface tension



(b) with surface tension

Fig. 22. Droplets rain on an elastic sponge model. When considering surface tension forces with $\gamma^{\mathrm{FV}} = 0.01\,\mathrm{N\,m^{-1}}$ as done in fig. 22b, the sponge is able to soak up the droplets. By squeezing the sponge, the fluid is forced out of the sponge and drips through the grate. Neglecting surface tension forces as shown in fig. 22a results in cubic droplets that disintegrate into particle spray when colliding with the sponge's surface. Note that the employed particle-based linear elasticity model proposed by Peer et al. [2018] is not able to completely prevent the sponge from penetrating through the grate. The sponge model was originally created by feklee and further modified by us. The rubber duck model was made by willie.

6.6 Versatility

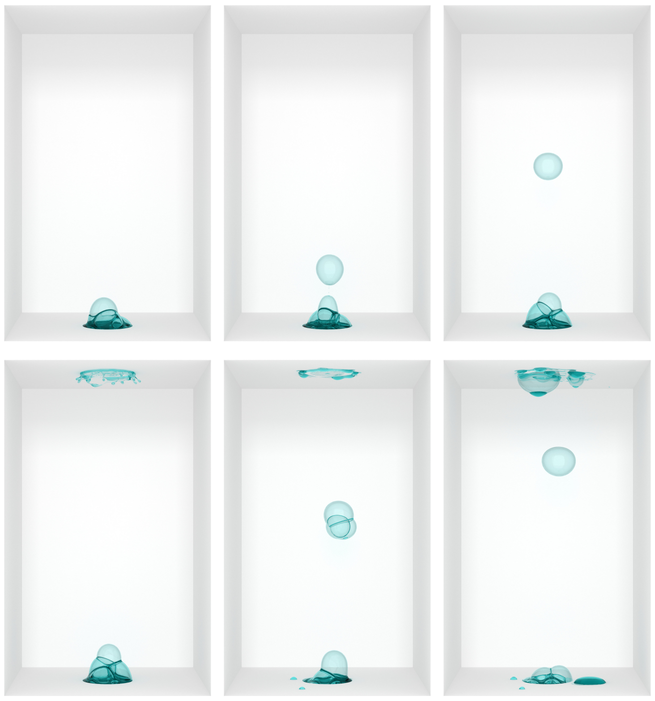
In this last section, we want to showcase a selection of scenarios that shall demonstrate the flexibility and versatility of our friction and surface tension solver. For a realistic assessment of the computational costs, in table 5 we list the computation times required for the following five simulations.

6.6.1 Water Lily. Starting with the simulation of a cleaning process of a water lily, fig. 21 shows the flower which is initially covered in dirt. The dirt particles are simulated as a fluid with high friction and no surface tension. When it starts to rain, fluid droplets hit the flower petals. Due to the hydrophobic petal surface and high γ^{FV} of the droplets, the dirt particles stick to the droplets. Small droplets are stopped by the covering dirt particles, but as they grow larger they are able to roll down the flower, picking up any dirt particles in their way. After some time, most of the dirt particles are cleaned of the petals. This example indicates how effortless a user can parametrize our surface tension and friction force in order to achieve a complex interplay between various fluid phases and materials.

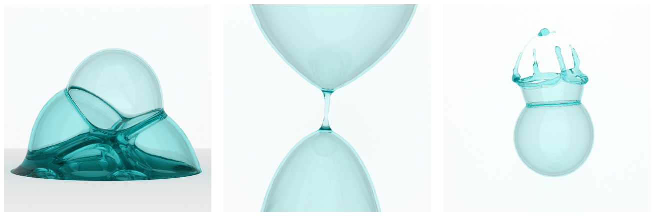
6.6.2 Sponge. Sponges are able to soak up water effectively due to their large hydrophilic surface areas. Capillary forces caused by surface tension pull the fluid into one of many small cavities inside the sponge. We are able to replicate this phenomenon with our surface tension formulation and an elastic sponge that is simulated with the linear elasticity solver proposed by Peer et al. [2018]. As we can see in fig. 22, only when considering surface tension forces, the sponge soaks up droplets until it is almost completely filled with water. To better visualize the volume of the absorbed fluid, we then squeeze the sponge to eject most of the water caught inside.

6.6.3 Bubbles. Soap Bubbles are fragile structures made of a gas volume that is captured inside a thin sheet of soapy fluid. The fluid experiences surface tension forces which determine the shape of the bubbles. Single bubbles are spherical while multiple touching bubbles form characteristic metastable clusters. Our unified solver is able to replicate the behavior of soap bubbles as shown in fig. 23. For this demonstration, we create gas bubbles modeled by a second particle fluid with lower rest density inside a water basin. The gas bubbles first accumulate on the fluid surface (fig. 23b), and then pinch off to rise up (fig. 23c). Too large strain can cause bubbles to burst, as depicted in fig. 23d. Even though surface thickness is often less than 2 particle diameters, bubbles can be robustly simulated using our surface tension formulation.

6.6.4 Lucy. Being a purely particle-based simulation method, our pressure, surface tension and friction solver naturally handles complex boundary geometry without difficulties. For demonstration,

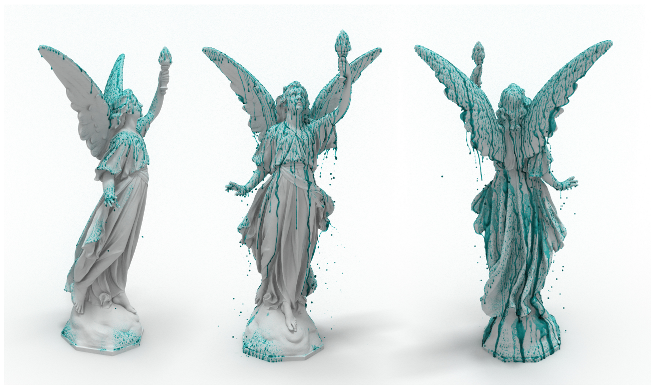


(a)

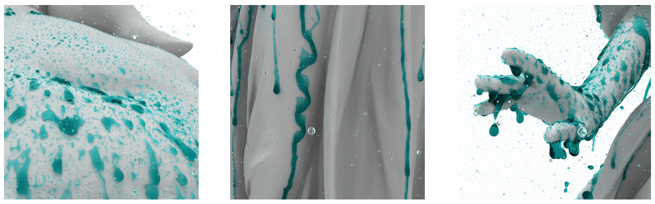


(b) (c) (d)

Fig. 23. We create soap bubbles by generating gas inside a pool of fluid. The gas is sampled with fluid particles that are given a low rest density such that they are pulled upwards by buoyancy forces, resulting in a cluster of bubbles on the fluid surface as displayed in fig. 23b. Figure 23c shows a close-up image of a bubble detaching from the cluster. Strain on a bubble's surface can become too large, causing it to burst midair as shown in fig. 23d. When hitting the ceiling, some bubbles pop while others are able to stay intact. Even though the thickness of bubbles is often less than two particle diameters, our unified fluid solver is able to robustly replicate the formation, motion and bursting of the shown gas bubbles. The soap has $\gamma^{FV} = 0.1\,\mathrm{N\,m^{-1}}$, $\gamma^{FF} = 0.1\,\mathrm{N\,m^{-1}}$ towards the gas inside the bubbles and $\gamma^{FB} = 0\,\mathrm{N\,m^{-1}}$ towards the boundary.



(a)



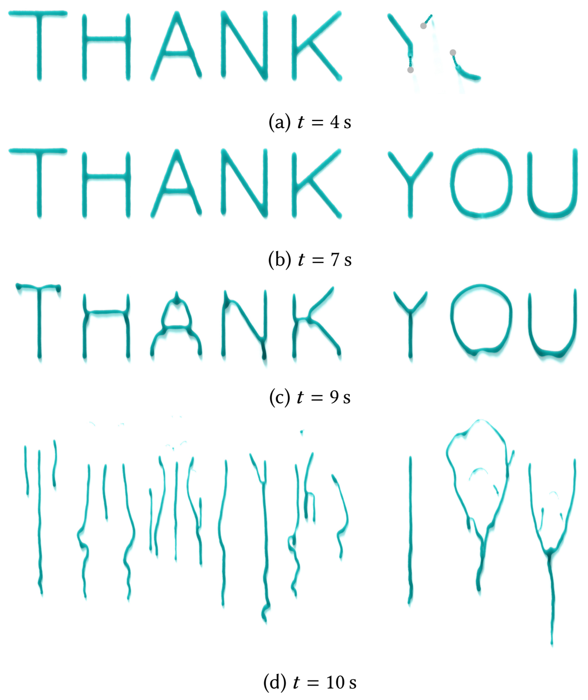
(b) (c) (d)

Fig. 24. Rainfall on a Lucy statue with a height of 1 m. We can see in figs. 24a and 24b that droplets first accumulate on exposed surfaces. Due to our friction and surface tension force, small droplets are able to truly stick to the statue. When droplets have grown sufficiently large, they start rolling downwards. In some places, merging droplets form continuous fluid streams. Figure 24c shows the formation of fluid streams similar to fig. 17 due to fluid particles experiencing less frictional resistance when the boundary is already covered by fluid. These streams sometimes shape themselves in a wavy pattern one may recognize from rain water flowing down a window at relatively high speed. In fig. 24d we can see that droplets also adhere to the statue due to $\gamma^{FV} = 1\,\mathrm{N\,m^{-1}} > \gamma^{FB} = 0\,\mathrm{N\,m^{-1}}$. The Lucy statue is originally from the Stanford 3D Scanning Repository. We use a version modified by Melllla.

we simulate a Lucy statue standing in the rain. Multiple effects one intuitively recognizes from real-life experience can also be observed in fig. 24: the small rain droplets colliding with exposed surfaces first stick to the statue due to surface tension and friction. By merging with other droplets they grow larger and eventually start rolling downwards. In the accompanying video one can see how droplets gain speed when picking up additional volume on their way down. Due to our friction force, rolling droplets leave behind wet trails. Other droplets tend to follow these trails as they impose less frictional resistance compared to a dry surface area. Small water streams are formed in places where many droplets meet. The streams sometimes shape themselves in a wavy pattern as shown in fig. 24c. In fig. 24d we can see that droplets need to gain a critical mass before they are able to overcome adhesion forces and detach from the statue. As a whole, such thin film fluid effects have so far been shown in only a few publications such as Stomakhin et al. [2019] and Stomakhin et al. [2023]. Towards the end of the simulation, the water on the statue is sampled by over 10 million actively simulated fluid particles.

6.6.5 *Inkjet Printing.*

Inkjet printers propel small ink droplets onto paper to create text and images. We mimic this mechanic, but instead of using paper that soaks up the ink, friction forces are employed to keep the ink in place on the vertical plane. Figure 25 shows the printing process and the final text. After the whole text is printed,



(a) $t = 4\,\mathrm{s}$

(b) $t = 7\,\mathrm{s}$

(c) $t = 9\,\mathrm{s}$

(d) $t = 10\,\mathrm{s}$

Fig. 25. Inkjet printing by shooting many ink droplets with $\gamma^{\mathrm{FV}} = 1\,\mathrm{N\,m^{-1}}$ onto a piece of paper. Surface tension and friction forces keep the ink in place on the vertical plane. After the text is printed, we decrease the coefficient of friction μ causing the letters to merge into fluid streams that run down the surface.

friction coefficient μ is reduced. As a consequence, the ink flows downwards with gravity.

7 Limitations and Future Work

Our surface tension computation requires a kernel support radius $\hbar$ larger than the one employed in the pressure computation, as illustrated in section 5.1. In case that $\hbar = 2h$ is used in the pressure solver, we propose $\hbar = 3h$ as a kernel support radius for all SPH approximations associated with surface tension. On the one hand, this reliably detects all particles belonging to an interface. Further, it alleviates the need for specialized kernel function shapes [Akinci et al. 2013; Jeske et al. 2023] and additional stabilization steps such as the consideration of air pressure [He et al. 2015b; Schechter and Bridson 2012], two-scale pressure estimation [He et al. 2015b] and anisotropic pressure filtering [He et al. 2015b] in order to prevent tensile instability. On the other hand however, in three dimensions, our pressure and friction computation on average only has to consider thirty neighbors in their SPH interpolations, while a kernel support of $3h$ used to compute surface tension is equivalent to an average of around one hundred neighbors inside the support radius. This causes the surface tension calculations to be computationally heavier compared to pressure and friction. Future work might be able to further improve simulation performance by finding ways to reduce the required surface tension support radius $\hbar$.

Some Coulomb friction formulations distinguish between static friction where the tangential relative velocity at contact $\mathbf{v}^{\mathrm{tang}}$ is zero and kinetic friction where $\mathbf{v}^{\mathrm{tang}} \neq \mathbf{0}$. The value given to the coefficient of friction μ depends on this distinction [Baraff 1991, 1993; McHale et al. 2022]. To keep things simple, in our implementation we follow Erleben [2017], Andrews and Erleben [2021] and others [Bender et al. 2014; Macklin et al. 2019; Peiret et al. 2019], and only considered one value for μ. We believe, however, that if desired, our method could be easily extended to support static and kinetic friction coefficients μ.

Finally, table 5 shows that small rain particles travelling at high speeds in figs. 17, 21 and 24 significantly impact simulation performance. Future work might be able to implement an individual timestep method [Goswami and Batty 2014; He et al. 2015a; Hernquist and Katz 1989; Hut and McMillan 1986; Makino et al. 2006; Saitoh and Makino 2009] to separate the timestep used to integrate positions of free-falling fluid particles from the timestep used for particles with high neighbor interaction.

8 Conclusion

In this paper, we presented an implicit fluid solver that considers pressure, surface tension and friction forces within an SPH environment. While the employed pressure computation is a variant of IISPH [Ihmsen et al. 2014a], new formulations have been developed for the surface tension and friction solver. Our surface tension combines the robust and intuitive properties of existing pairwise tension formulations with the physical correctness of CSF approaches. We achieve this by considering number densities to detect particles belonging to an interface and compute surface tension forces depending on a particle's contribution to the interface. The experiments in section 6 show that our surface tension not only produces visually pleasing results, but also causes the correct amount of tension for all surface curvatures.

Using Coulomb friction at the SPH fluid-boundary interface appears to be a newer concept, only few related works can be found in the literature. Still, our tests demonstrated that our final friction solver allows us to replicate well-known real life phenomena. This includes, in particular, lifelike looking stick-slip transitions of droplets at inclined surfaces, which have not yet gained much attention within the SPH community, despite their significant impact on perceived realism of droplet behavior.

Following the recent trend to solve forces in a unified manner, our solver simultaneously computes pressure, surface tension and friction forces that are consistent with each other. Our experiments indicate that this improves simulation stability and correctness of results, allowing us to simulate the behavior of fluids in complex and challenging scenarios.

Acknowledgments

We thank FIFTY2 Technology for providing an academic license of PreonLab [FIFTY2 Technology 2024], which was used to generate all shown images of simulations. We acknowledge support by the state of Baden-Württemberg through bwHPC. The tumbling board model by iolalla from fig. 12 is licensed under CC BY-SA 3.0. The valve wheel model by JohnEdwa from fig. 15 is licensed under CC

Table 5. Computational costs of the simulation scenarios presented in section 6.6. From left to right we list the maximum number of fluid particles present in the simulation, number of boundary particles, particle size h, average timestep Δt, average solver iterations, computation time required per simulation step and computation time required per simulated second. All computations are performed on a 24-core 5.4 GHz Intel Core i9-13900K workstation. The SPH neighbor search and basic particle accelerations such as gravity are performed on the CPU, an OpenCL implementation of our solver runs on an NVIDIA GeForce RTX 4090 graphics card. Note the relatively small timestep Δt used in the *Water Lily* and *Lucy* scenario. In both cases, the simulation requires a low Δt to catch the collision between small fluid particles raining with high speed on the solid boundaries. Together with the large number of simulated particles, this results in significant computation times. We want to mention that fast moving particles threaten the performance of numerical simulations in general [Makino et al. 2006; Saitoh and Makino 2009] and are not a difficulty in proposed method in particular.

						computation time per	
	fluid particles	boundary particles	h	Δt	solver iter.	step	simulated s
Water Lily	1,300,000	4,100,000	0.1 mm	0.10 ms	8	2.0 s	5.6 h
Sponge	1,400,000	1,600,000	0.5 mm	0.33 ms	10	2.1 s	1.8 h
Soap Bubbles	4,600,000	1,700,000	0.5 mm	0.26 ms	10	6.2 s	6.6 h
Lucy	10,200,200	21,000,000	1.0 mm	0.04 ms	8	7.0 s	48.8 h
Inkjet Printing	500,000	26,200,000	0.7 mm	1.00 ms	15	1.1 s	0.3 h

BY 4.0. The water lily model by guppyk from fig. 21 is licensed under CC BY 4.0. The sponge model by feklee, modified by us and used in fig. 22 to generate the elastic object, is licensed under CC BY 4.0. The rubber duck model by willie used in the same scenario is licensed under CC0 1.0. The Lucy statue used in figs. 1 and 24, originally from the Stanford 3D Scanning Repository and modified by Mellla, is licensed under the MIT License.

References

Stefan Adami, Xiangyu Hu, and Nikolaus Adams. 2010. A new surface-tension formulation for multi-phase SPH using a reproducing divergence approximation. *J. Comput. Phys.* 229, 13 (2010), 5011–5021. https://doi.org/10.1016/j.jcp.2010.03.022

Nadir Akinci, Gizem Akinci, and Matthias Teschner. 2013. Versatile Surface Tension and Adhesion for SPH Fluids. *ACM Trans. Graph.* 32, 6, Article 182 (nov 2013), 8 pages. https://doi.org/10.1145/2508363.2508395

Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile Rigid-Fluid Coupling for Incompressible SPH. *ACM Trans. Graph.* 31, 4, Article 62 (jul 2012), 8 pages. https://doi.org/10.1145/2185520.2185558

Michael Andersen, Sarah Niebe, and Kenny Erleben. 2017. A fast linear complementarity problem solver for fluid animation using high level algebra interfaces for GPU libraries. *Computers & Graphics* 69 (2017), 36–48. https://doi.org/10.1016/j.cag.2017.09.006

Sheldon Andrews and Kenny Erleben. 2021. Contact and Friction Simulation for Computer Graphics. In *ACM SIGGRAPH 2021 Courses* (Virtual Event, USA) *(SIGGRAPH '21)*. Association for Computing Machinery, New York, NY, USA, Article 2, 124 pages. https://doi.org/10.1145/3450508.3464571

Erin Arai, Alexandre Tartakovsky, R. Glynn Holt, Sheryl Grace, and Emily Ryan. 2020. Comparison of surface tension generation methods in smoothed particle hydrodynamics for dynamic systems. *Computers & Fluids* 203 (2020), 104540. https://doi.org/10.1016/j.compfluid.2020.104540

Stefan Band, Christoph Gissler, Markus Ihmsen, Jens Cornelis, Andreas Peer, and Matthias Teschner. 2018a. Pressure Boundaries for Implicit Incompressible SPH. *ACM Trans. Graph.* 37, 2, Article 14 (feb 2018), 11 pages. https://doi.org/10.1145/3180486

Stefan Band, Christoph Gissler, Andreas Peer, and Matthias Teschner. 2018b. MLS pressure boundaries for divergence-free and viscous SPH fluids. *Computers & Graphics* 76 (2018), 37–46. https://doi.org/10.1016/j.cag.2018.08.001

Stefan Band, Christoph Gissler, and Matthias Teschner. 2020. Compressed Neighbour Lists for SPH. *Computer Graphics Forum* 39, 1 (2020), 531–542. https://doi.org/10.1111/cgf.13890

David Baraff. 1991. Coping with friction for non-penetrating rigid body simulation. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*. Association for Computing Machinery, New York, NY, USA, 31–41. https://doi.org/10.1145/122718.122722

David Baraff. 1993. Non-penetrating rigid body simulation. In *Proceedings of Eurographics '93 State of the Art Reports*.

Christopher Batty, Andres Uribe, Basile Audoly, and Eitan Grinspun. 2012. Discrete viscous sheets. *ACM Trans. Graph.* 31, 4, Article 113 (jul 2012), 7 pages. https://doi.org/10.1145/2185520.2185609

Markus Becker and Matthias Teschner. 2007. Weakly Compressible SPH for Free Surface Flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '07)*. Eurographics Association, Goslar, DEU, 209–217. https://doi.org/10.2312/SCA/SCA07/209-218

Jan Bender, Kenny Erleben, and Jeff Trinkle. 2014. Interactive Simulation of Rigid Body Dynamics in Computer Graphics. *Computer Graphics Forum* 33, 1 (2014), 246–270. https://doi.org/10.1111/cgf.12272

Jan Bender and Dan Koschier. 2015. Divergence-Free Smoothed Particle Hydrodynamics. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '15)*. Association for Computing Machinery, New York, NY, USA, 147–155. https://doi.org/10.1145/2786784.2786796

Jan Bender, Tassilo Kugelstadt, Marcel Weiler, and Dan Koschier. 2019. Volume Maps: An Implicit Boundary Representation for SPH. In *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games* (Newcastle upon Tyne, United Kingdom) *(MIG '19)*. Association for Computing Machinery, New York, NY, USA, Article 26, 10 pages. https://doi.org/10.1145/3359566.3360077

Jan Bender, Tassilo Kugelstadt, Marcel Weiler, and Dan Koschier. 2020. Implicit Frictional Boundary Handling for SPH. *IEEE Transactions on Visualization and Computer Graphics* 26, 10 (2020), 2982–2993. https://doi.org/10.1109/TVCG.2020.3004245

Jan Bender, Lukas Westhofen, and Stefan Rhys Jeske. 2023. Consistent SPH Rigid-Fluid Coupling. In *Vision, Modeling, and Visualization*. The Eurographics Association. https://doi.org/10.2312/vmv.20231244

Michael Berry. 1971. The molecular mechanism of surface tension. *Physics Education* 6, 2 (mar 1971), 79. https://doi.org/10.1088/0031-9120/6/2/001

Kirsten Bobzin, Hendrik Heinemann, Kevin Jasutyn, Stefan Rhys Jeske, Jan Bender, Sergej Warkentin, Oleg Mokrov, Rahul Sharma, and Uwe Reisgen. 2023. Modeling the Droplet Impact on the Substrate with Surface Preparation in Thermal Spraying with SPH. *Journal of Thermal Spray Technology* (jan 2023). https://doi.org/10.1007/s11666-023-01534-0

Kenneth Bodin, Claude Lacoursiere, and Martin Servin. 2012. Constraint Fluids. *IEEE Transactions on Visualization and Computer Graphics* 18, 3 (March 2012), 516–526. https://doi.org/10.1109/TVCG.2011.29

Landon Boyd and Robert Bridson. 2012. MultiFLIP for energetic two-phase fluid simulation. *ACM Trans. Graph.* 31, 2, Article 16 (apr 2012), 12 pages. https://doi.org/10.1145/2159516.2159522

Jeremiah. Brackbill, Douglas Kothe, and Zemach Charles. 1992. A Continuum Method for Modeling Surface Tension. *J. Comput. Phys.* 100 (07 1992). https://doi.org/10.1016/0021-9991(92)90240-Y

Robert Bridson. 2008. *Fluid Simulation.* A. K. Peters, Ltd., USA.

Richard W Cottle, Jong-Shi Pang, and Richard E Stone. 2009. *The linear complementarity problem.* SIAM. https://doi.org/10.1137/1.9780898719000

Hadrien Courtecuisse and Jérémie Allard. 2009. Parallel Dense Gauss-Seidel Algorithm on Many-Core Processors. In *2009 11th IEEE International Conference on High Performance Computing and Communications*. 139–147. https://doi.org/10.1109/HPCC.2009.51

Matthew E. Cross and Emma V. E. Plunkett. 2014. *Principles of surface tension.* Cambridge University Press, 57–58.

Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double bubbles sans toil and trouble: discrete circulation-preserving vortex sheets for soap films and foams. *ACM Trans. Graph.* 34, 4, Article 149 (jul 2015), 9 pages. https://doi.org/10.1145/2767003

Mathieu Desbrun and Marie-Paule Gascuel. 1996. Smoothed Particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation '96*, Ronan Boulic and Gerard Hégron (Eds.). Springer Vienna, Vienna, 61–76.

Riqiang Duan, Chen Sun, and Shengyao Jiang. 2020. A new surface tension formulation for particle methods. *International Journal of Multiphase Flow* 124 (2020), 103187. https://doi.org/10.1016/j.ijmultiphaseflow.2019.103187

Kenny Erleben. 2013. Numerical Methods for Linear Complementarity Problems in Physics-Based Animation. In *ACM SIGGRAPH 2013 Courses* (Anaheim, California) *(SIGGRAPH '13)*. Association for Computing Machinery, New York, NY, USA, Article 8, 42 pages. https://doi.org/10.1145/2504435.2504443

Kenny Erleben. 2017. Rigid Body Contact Problems Using Proximal Operators. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '17)*. Association for Computing Machinery, New York, NY, USA, Article 13, 12 pages. https://doi.org/10.1145/3099564.3099575

FIFTY2 Technology. 2024. PreonLab. www.fifty2.eu

Makoto Fujisawa and Kenjiro T. Miura. 2015. An Efficient Boundary Handling with a Modified Density Calculation for SPH. *Computer Graphics Forum* 34, 7 (2015), 155–162. https://doi.org/10.1111/cgf.12754

Nan Gao, Florian Geyer, Dominik W. Pilat, Sanghyuk Wooh, Doris Vollmer, Hans-Jürgen Butt, and Rüdiger Berger. 2018. How drops start sliding over solid surfaces. *Nature Physics* 14, 2 (01 Feb 2018), 191–196. https://doi.org/10.1038/nphys4305

Pierre-Gilles Gennes, Françoise Brochard-Wyart, David Quéré, Alex Reisinger, and Benjamin Widom. 2004. *Capillarity and wetting phenomena: drops, bubbles, pearls, waves.* Springer, New York, NY. https://doi.org/10.1007/978-0-387-21656-0

Ruth A. Gingold and Joseph J. Monaghan. 1977. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 181, 3 (12 1977), 375–389. https://doi.org/10.1093/mnras/181.3.375

Christoph Gissler, Andreas Henne, Stefan Band, Andreas Peer, and Matthias Teschner. 2020. An implicit compressible SPH solver for snow simulation. *ACM Trans. Graph.* 39, 4, Article 36 (aug 2020), 16 pages. https://doi.org/10.1145/3386569.3392431

Christoph Gissler, Andreas Peer, Stefan Band, Jan Bender, and Matthias Teschner. 2019. Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling. *ACM Trans. Graph.* 38, 1, Article 5 (Jan. 2019), 13 pages. https://doi.org/10.1145/3284980

Prashant Goswami and Christopher Batty. 2014. Regional Time Stepping for SPH. In *Eurographics 2014 - Short Papers*, Eric Galin and Michael Wand (Eds.). The Eurographics Association. https://doi.org//10.2312/egsh.20141011

Steffen Hardt and Glen McHale. 2022. Flow and Drop Transport Along Liquid-Infused Surfaces. *Annual Review of Fluid Mechanics* 54, Volume 54, 2022 (2022), 83–104. https://doi.org/10.1146/annurev-fluid-030121-113156

Liangliang He, Xiaojuan Ban, Xu Liu, and Xiaokun Wang. 2015a. Individual Time Stepping for SPH Fluids. In *EG 2015 - Short Papers*, B. Bickel and T. Ritschel (Eds.). The Eurographics Association. https://doi.org/10.2312/egsh.20151010

Xiaowei He, Huamin Wang, Fengjun Zhang, Hongan Wang, Guoping Wang, and Kun Zhou. 2015b. Robust Simulation of Sparsely Sampled Thin Features in SPH-Based Free Surface Flows. *ACM Trans. Graph.* 34, 1, Article 7 (dec 2015), 9 pages. https://doi.org/10.1145/2682630

Lars Hernquist and Neal Katz. 1989. TREESPH: A Unification of SPH with the Hierarchical Tree Method. *The Astrophysical journal. Supplement series* 70 (June 1989), 419. https://doi.org/10.1086/191344

Jeong-Mo Hong, Ho-Young Lee, Jong-Chul Yoon, and Chang-Hun Kim. 2008. Bubbles alive. In *ACM SIGGRAPH 2008 Papers* (Los Angeles, California) *(SIGGRAPH '08)*. Association for Computing Machinery, New York, NY, USA, Article 48, 4 pages. https://doi.org/10.1145/1399504.1360647

Xiangyu Hu and Nikolaus Adams. 2006. A multi-phase SPH method for macroscopic and mesoscopic flows. *J. Comput. Phys.* 213, 2 (2006), 844–861. https://doi.org/10.1016/j.jcp.2005.09.001

Libo Huang, Torsten Hädrich, and Dominik L. Michels. 2019. On the accurate large-scale simulation of ferrofluids. *ACM Trans. Graph.* 38, 4, Article 93 (jul 2019), 15 pages. https://doi.org/10.1145/3306346.3322973

Markus Huber, Stefan Reinhardt, Daniel Weiskopf, and Bernhard Eberhardt. 2015. Evaluation of Surface Tension Models for SPH-Based Fluid Animations Using a Benchmark Test. In *Workshop on Virtual Reality Interaction and Physical Simulation*, Fabrice Jaillet, Florence Zara, and Gabriel Zachmann (Eds.). The Eurographics Association. https://doi.org/10.2312/vriphys.20151333

Piet Hut and Stephen McMillan. 1986. *The Use of supercomputers in stellar dynamics : proceedings of a workshop held at the Institute for advanced study Princeton, USA, 1986.* Springer-Verlag, Berlin.

David A. B. Hyde, Steven W. Gagniere, Alan Marquez-Razon, and Joseph Teran. 2020. An Implicit Updated Lagrangian Formulation for Liquids with Large Surface Energy. *ACM Trans. Graph.* 39, 6, Article 183 (nov 2020), 13 pages. https://doi.org/10.1145/3414685.3417845

Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. 2014a. Implicit Incompressible SPH. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (2014), 426–435. https://doi.org/10.1109/TVCG.2013.105

Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. 2014b. SPH Fluids in Computer Graphics. In *Eurographics 2014 - State of the Art Reports*, Sylvain Lefebvre and Michela Spagnuolo (Eds.). The Eurographics Association. https://doi.org/10.2312/egst.20141034

Sadashige Ishida, Masafumi Yamamoto, Ryoichi Ando, and Toshiya Hachisuka. 2017. A hyperbolic geometric flow for evolving films and foams. *ACM Trans. Graph.* 36, 6, Article 199 (nov 2017), 11 pages. https://doi.org/10.1145/3130800.3130835

Stefan Rhys Jeske, Lukas Westhofen, Fabian Löschner, José Antonio Fernández-fernández, and Jan Bender. 2023. Implicit Surface Tension for SPH Fluid Simulation. *ACM Trans. Graph.* 43, 1, Article 13 (nov 2023), 14 pages. https://doi.org/10.1145/3631936

Dan Koschier and Jan Bender. 2017. Density Maps for Improved SPH Boundary Handling. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '17)*. Association for Computing Machinery, New York, NY, USA, Article 1, 10 pages. https://doi.org/10.1145/3099564.3099565

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2022. A Survey on SPH Methods in Computer Graphics. *Computer Graphics Forum* 41, 2 (2022), 737–760. https://doi.org/10.1111/cgf.14508

Benny Lautrup. 2011. *Physics of Continuous Matter: Exotic and Everyday Phenomena in the Macroscopic World.* Taylor & Francis Inc. https://doi.org/10.1201/9781439894200

Shusen Liu, Xiaowei He, Wencheng Wang, and Enhua Wu. 2022. Adapted SIMPLE Algorithm for Incompressible SPH Fluids With a Broad Range Viscosity. *IEEE Transactions on Visualization and Computer Graphics* 28, 9 (2022), 3168–3179. https://doi.org/10.1109/TVCG.2021.3055789

Fabian Löschner, José Antonio Fernández-Fernández, Stefan Rhys Jeske, Andreas Longva, and Jan Bender. 2023. Micropolar Elasticity in Physically-Based Animation. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6, 3, Article 46 (aug 2023), 24 pages. https://doi.org/10.1145/3606922

Leon B Lucy. 1977. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal, vol. 82, Dec. 1977, p. 1013-1024.* 82 (1977), 1013–1024. https://doi.org/10.1086/112164

Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviychuk. 2019. Non-smooth Newton Methods for Deformable Multi-body Dynamics. *ACM Trans. Graph.* 38, 5, Article 140 (oct 2019), 20 pages. https://doi.org/10.1145/3338695

Miles Macklin and Matthias Müller. 2013. Position Based Fluids. *ACM Trans. Graph.* 32, 4, Article 104 (jul 2013), 12 pages. https://doi.org/10.1145/2461912.2461984

Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. 2014. Unified particle physics for real-time applications. *ACM Trans. Graph.* 33, 4, Article 153 (jul 2014), 12 pages. https://doi.org/10.1145/2601097.2601152

Junichiro Makino, Piet Hut, Murat Kaplan, and Hasan Saygın. 2006. A time-symmetric block time-step algorithm for N-body simulations. *New Astronomy* 12, 2 (2006), 124–133. https://doi.org/10.1016/j.newast.2006.06.003

Glen McHale, Nan Gao, Gary G. Wells, Hernán Barrio-Zhang, and Rodrigo Ledesma-Aguilar. 2022. Friction Coefficients for Droplets on Solids: The Liquid–Solid Amontons' Laws. *Langmuir* 38, 14 (12 Apr 2022), 4425–4433. https://doi.org/10.1021/acs.langmuir.2c00178

Oleg Mokrov, Sergej Warkentin, Lukas Westhofen, Stefan Jeske, Jan Bender, Rahul Sharma, and Uwe Reisgen. 2024. Simulation of wire metal transfer in the cold metal transfer (CMT) variant of gas metal arc welding using the smoothed particle hydrodynamics (SPH) approach. *Materialwissenschaft und Werkstofftechnik* 55, 1 (2024), 62–71. https://doi.org/10.1002/mawe.202300166

Joseph J. Monaghan. 1992. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics* 30, 1 (1992), 543–574. https://doi.org/10.1146/annurev.aa.30.090192.002551

Joseph P. Morris. 2000. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids* 33, 3 (2000), 333–353. https://doi.org/10.1002/1097-0363(20000615)33:3<333::AID-FLD11>3.0.CO;2-7

Joseph P. Morris, Patrick J. Fox, and Yi Zhu. 1997. Modeling Low Reynolds Number Incompressible Flows Using SPH. *J. Comput. Phys.* 136, 1 (1997), 214–226. https://doi.org/10.1006/jcph.1997.5776

Matthias Müller, David Charypar, and Markus Gross. 2003. Particle-Based Fluid Simulation for Interactive Applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03)*. Eurographics Association, Goslar, DEU, 154–159.

Matthias Müller, Simon Schirm, Matthias Teschner, Bruno Heidelberger, and Markus Gross. 2004. Interaction of fluids with deformable solids. *Computer Animation and Virtual Worlds* 15, 3-4 (2004), 159–171. https://doi.org/10.1002/cav.18

Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization* (second ed.). Springer, New York, NY, USA.

Severin Nugent and Harald Alois Posch. 2000. Liquid drops and surface tension with smoothed particle applied mechanics. *Phys. Rev. E* 62 (Oct 2000), 4968–4975. Issue 4. https://doi.org/10.1103/PhysRevE.62.4968

Jens Orthmann, Hendrik Hochstetter, Julian Bader, Serkan Bayraktar, and Andreas Kolb. 2013. Consistent surface model for SPH-based fluid transport. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Anaheim, California) *(SCA '13)*. Association for Computing Machinery, New York, NY, USA, 95–103. https://doi.org/10.1145/2485895.2485902

Andreas Peer, Christoph Gissler, Stefan Band, and Matthias Teschner. 2018. An Implicit SPH Formulation for Incompressible Linearly Elastic Solids. *Computer Graphics*

Forum 37, 6 (2018), 135–148. https://doi.org/doi.org/10.1111/cgf.13317

Andreas Peer, Markus Ihmsen, Jens Cornelis, and Matthias Teschner. 2015. An Implicit Viscosity Formulation for SPH Fluids. *ACM Trans. Graph.* 34, 4, Article 114 (jul 2015), 10 pages. https://doi.org/10.1145/2766925

Andreas Peer and Matthias Teschner. 2017. Prescribed Velocity Gradients for Highly Viscous SPH Fluids with Vorticity Diffusion. *IEEE Transactions on Visualization and Computer Graphics* 23, 12 (dec 2017), 2656–2662. https://doi.org/10.1109/TVCG.2016.2636144

Albert Peiret, Sheldon Andrews, József Kövecses, Paul G. Kry, and Marek Teichmann. 2019. Schur Complement-based Substructuring of Stiff Multibody Systems with Contact. *ACM Trans. Graph.* 38, 5, Article 150 (oct 2019), 17 pages. https://doi.org/10.1145/3355621

Ettore Pennestri, Valerio Rossi, Pietro Salvini, and Pier Paolo Valentini. 2016. Review and comparison of dry friction force models. *Nonlinear Dynamics* 83 (03 2016). https://doi.org/10.1007/s11071-015-2485-3

Morten Poulsen, Sarah Maria Niebe Abel, and Kenny Erleben. 2010. Heuristic convergence rate improvements of the projected Gauss-Seidel method for frictional contact problems. In *WSCG 2010*, Vaclav Skala (Ed.). Vaclav Skala - Union Agency, 135–142. 18th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2010 ; Conference date: 01-02-2010 Through 04-02-2010.

Timo Probst and Matthias Teschner. 2023. Monolithic Friction and Contact Handling for Rigid Bodies and Fluids Using SPH. *Computer Graphics Forum* 42, 1 (2023), 155–179. https://doi.org/10.1111/cgf.14727

Bo Ren, Chenfeng Li, Xiao Yan, Ming C. Lin, Javier Bonet, and Shi-Min Hu. 2014. Multiple-Fluid SPH Simulation Using a Mixture Model. *ACM Trans. Graph.* 33, 5, Article 171 (sep 2014), 11 pages. https://doi.org/10.1145/2645703

Bo Ren, Tailing Yuan, Chenfeng Li, Kun Xu, and Shi-Min Hu. 2018. Real-Time High-Fidelity Surface Flow Simulation. *IEEE Transactions on Visualization and Computer Graphics* 24, 8 (2018), 2411–2423. https://doi.org/10.1109/TVCG.2017.2720672

Takayuki R. Saitoh and Junichiro Makino. 2009. A Necessary Condition for Individual Time Steps in SPH Simulations. *The Astrophysical Journal* 697, 2 (may 2009), L99. https://doi.org/10.1088/0004-637X/697/2/L99

Hagit Schechter and Robert Bridson. 2012. Ghost SPH for animating water. *ACM Trans. Graph.* 31, 4, Article 61 (jul 2012), 8 pages. https://doi.org/10.1145/2185520.2185557

Thorsten Schindler, Binh Nguyen, and Jeff Trinkle. 2011. Understanding the difference between prox and complementarity formulations for simulation of systems with contact. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1433–1438. https://doi.org/10.1109/IROS.2011.6094779

Morten Silcowitz, Sarah Niebe, and Kenny Erleben. 2009. Nonsmooth Newton Method for Fischer Function Reformulation of Contact Force Problems for Interactive Rigid Body Simulation. In *Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2009)*, Hartmut Prautzsch, Alfred Schmitt, Jan Bender, and Matthias Teschner (Eds.). The Eurographics Association. https://doi.org/10.2312/PE/vriphys/vriphys09/105-114

Morten Silcowitz-Hansen, Sarah Niebe, and Kenny Erleben. 2010. A Nonsmooth Nonlinear Conjugate Gradient Method For Interactive Contact Force Problems. *Vis. Comput.* 26, 6–8 (June 2010), 893–901. https://doi.org/10.1007/s00371-010-0502-6

Barbara Solenthaler and Renato Pajarola. 2008. Density contrast SPH interfaces. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Dublin, Ireland) *(SCA '08)*. Eurographics Association, Goslar, DEU, 211–218.

Barbara Solenthaler and Renato Pajarola. 2009. Predictive-Corrective Incompressible SPH. In *ACM SIGGRAPH 2009 Papers* (New Orleans, Louisiana) *(SIGGRAPH '09)*. Association for Computing Machinery, New York, NY, USA, Article 40, 6 pages. https://doi.org/10.1145/1576246.1531346

James G. Speight (Ed.). 2017. *Lange's Handbook of Chemistry* (17th edition ed.). McGraw-Hill Education, New York.

Alexey Stomakhin, Steve Lesser, Joel Wretborn, Sean Flynn, Johnathan Nixon, Nicholas Illingworth, Adrien Rollet, Kevin Blom, and Douglas Mchale. 2023. Pahi: A Unified Water Pipeline and Toolset. In *Proceedings of the 2023 Digital Production Symposium* (Los Angeles, CA, USA) *(DigiPro '23)*. Association for Computing Machinery, New York, NY, USA, Article 11, 13 pages. https://doi.org/10.1145/3603521.3604291

Alexey Stomakhin, Andrew Moffat, and Gary Boyle. 2019. A practical guide to thin film and drips simulation. In *ACM SIGGRAPH 2019 Talks* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 72, 2 pages. https://doi.org/10.1145/3306307.3328141

Tetsuya Takahashi and Christopher Batty. 2020. Monolith: A Monolithic Pressure-Viscosity-Contact Solver for Strong Two-Way Rigid-Rigid Rigid-Fluid Coupling. *ACM Trans. Graph.* 39, 6, Article 182 (nov 2020), 16 pages. https://doi.org/10.1145/3414685.3417798

Tetsuya Takahashi, Yoshinori Dobashi, Issei Fujishiro, Tomoyuki Nishita, and Ming C. Lin. 2015. Implicit Formulation for SPH-based Viscous Fluids. *Computer Graphics Forum* (2015). https://doi.org/10.1111/cgf.12578

Alexandre Tartakovsky and Paul Meakin. 2005. Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Phys. Rev. E* 72 (Aug 2005), 026301. Issue 2. https://doi.org/10.1103/PhysRevE.72.026301

Alexandre M. Tartakovsky, Nathaniel Trask, Kai Pan, Bruce Jones, Wenxiao Pan, and John Williams. 2016. Smoothed particle hydrodynamics and its applications for multiphase flow and reactive transport in porous media. *Computational Geosciences* 20, 4 (01 Aug 2016), 807–834. https://doi.org/10.1007/s10596-015-9468-9

Nils Thürey, Chris Wojtan, Markus Gross, and Greg Turk. 2010. A multiscale approach to mesh-based surface tension flows. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) *(SIGGRAPH '10)*. Association for Computing Machinery, New York, NY, USA, Article 48, 10 pages. https://doi.org/10.1145/1833349.1778785

Hui Wang, Yongxu Jin, Anqi Luo, Xubo Yang, and Bo Zhu. 2020. Codimensional surface tension flow using moving-least-squares particles. *ACM Trans. Graph.* 39, 4, Article 42 (aug 2020), 16 pages. https://doi.org/10.1145/3386569.3392487

Xiaokun Wang, Xiao-Juan Ban, Zhang Yalan, Si-Nuo Liu, and Peng-Fei Ye. 2017. Surface Tension Model Based on Implicit Incompressible Smoothed Particle Hydrodynamics for Fluid Simulation. *Journal of Computer Science and Technology* 32 (11 2017), 1186–1197. https://doi.org/10.1007/s11390-017-1793-0

Marcel Weiler, Dan Koschier, Magnus Brand, and Jan Bender. 2018. A Physically Consistent Implicit Viscosity Solver for SPH Fluids. *Computer Graphics Forum (Eurographics)* 37, 2 (2018), 145–155.

Rene Winchenbach, Rustam Akhunov, and Andreas Kolb. 2020. Semi-Analytic Boundary Handling below Particle Resolution for Smoothed Particle Hydrodynamics. *ACM Trans. Graph.* 39, 6, Article 173 (nov 2020), 17 pages. https://doi.org/10.1145/3414685.3417829

Jingrui Xing, Liangwang Ruan, Bin Wang, Bo Zhu, and Baoquan Chen. 2022. Position-Based Surface Tension Flow. *ACM Trans. Graph.* 41, 6, Article 244 (nov 2022), 12 pages. https://doi.org/10.1145/3550454.3555476

Lijing Yang, Milad Rakhsha, and Dan Negrut. 2019. Comparison of Surface Tension Models in Smoothed Particles Hydrodynamics Method *(International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. Volume 6: 15th International Conference on Multibody Systems, Nonlinear Dynamics, and Control)*. V006T09A028. https://doi.org/10.1115/DETC2019-98124

Tao Yang, Ming C. Lin, Ralph R. Martin, Jian Chang, and Shi-Min Hu. 2016. Versatile interactions at interfaces for SPH-based simulations. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Zurich, Switzerland) *(SCA '16)*. Eurographics Association, Goslar, DEU, 57–66.

Tao Yang, Ralph R. Martin, Ming C. Lin, Jian Chang, and Shi-Min Hu. 2017. Pairwise Force SPH Model for Real-Time Multi-Interaction Applications. *IEEE Transactions on Visualization and Computer Graphics* 23, 10 (2017), 2235–2247. https://doi.org/10.1109/TVCG.2017.2706289

Thomas Young. 1805. An Essay on the Cohesion of Fluids. *Philosophical Transactions of the Royal Society of London* 95 (1805), 65–87. https://doi.org/10.1098/rstl.1805.0005

Jihun Yu, Chris Wojtan, Greg Turk, and Chee Yap. 2012. Explicit Mesh Surfaces for Particle Based Fluids. *Computer Graphics Forum* 31, 2pt4 (2012), 815–824. https://doi.org/10.1111/j.1467-8659.2012.03062.x

Yizhong Zhang, Huamin Wang, Shuai Wang, Yiying Tong, and Kun Zhou. 2012. A Deformable Surface Model for Real-Time Water Drop Animation. *IEEE Transactions on Visualization and Computer Graphics* 18, 8 (2012), 1281–1289. https://doi.org/10.1109/TVCG.2011.141

Bo Zhu, Minjae Lee, Ed Quigley, and Ronald Fedkiw. 2015. Codimensional non-Newtonian fluids. *ACM Trans. Graph.* 34, 4, Article 115 (jul 2015), 9 pages. https://doi.org/10.1145/2766981

Bo Zhu, Ed Quigley, Matthew Cong, Justin Solomon, and Ronald Fedkiw. 2014. Codimensional surface tension flow on simplicial complexes. *ACM Trans. Graph.* 33, 4, Article 111 (jul 2014), 11 pages. https://doi.org/10.1145/2601097.2601201

Fernando Zorilla, Marcel Ritter, Johannes Sappl, Wolfgang Rauch, and Matthias Harders. 2020. Accelerating Surface Tension Calculation in SPH via Particle Classification and Monte Carlo Integration. *Computers* 9, 2 (2020), 23. https://doi.org/10.3390/computers9020023

A Diagonal Elements

The goal of this section is to give a detailled description of how we can translate the fixed-point iteration from eq. (18) into a relaxed Jacobi update step by choosing appropriate values for α_f^{P}, α_f^{ST} and α_f^{F}.

A.1 Pressure

As an estimate of a safe update step size, Jacobi pressure solvers use a value for α_f^{P} that describes how sensitive the volume error V_f^{err} reacts on changes of pressure p_f [Ihmsen et al. 2014a; Koschier et al.

2022]:

$$\alpha_f^{\mathrm{P}} = \left(\frac{\partial}{\partial p_f} \ V_f^{\mathrm{err}}(t+\Delta t) \right)^{-1}. \tag{30}$$

If V_f^{err} changes much for small variations of p_f, α_f^{P} will be small. As a consequence, one fixed-point iteration of eq. (18a) will only cause a small difference in the iterates of p_f. This way, the solver is able to adapt its step size to the sensitivity of the underlying system. In order to compute α_f^{P} we need to evaluate $\frac{\partial}{\partial p_f} \ V_f^{\mathrm{err}}(t+\Delta t)$ using eqs. (4), (20) and (21):

$$\begin{aligned} &\frac{\partial}{\partial p_f} \ V_f^{\mathrm{err}}(t+\Delta t) \\ =& -V_f^0 \sum_{f_f} \Delta t \left(\frac{\partial}{\partial p_f} \mathbf{v}_f(t+\Delta t) - \frac{\partial}{\partial p_f} \mathbf{v}_{f_f}(t+\Delta t) \right) \cdot \nabla W_{f,f_f}^{\mathrm{P}}(t) \\ & - V_f^0 \sum_{f_b} \Delta t \left(\frac{\partial}{\partial p_f} \mathbf{v}_f(t+\Delta t) - \frac{\partial}{\partial p_f} \mathbf{v}_{f_b}(t+\Delta t) \right) \cdot \nabla W_{f,f_b}^{\mathrm{P}}(t) \end{aligned} \tag{31}$$

with

$$\begin{aligned} \frac{\partial}{\partial p_f} \ \mathbf{v}_f(t+\Delta t) =& -\frac{\Delta t}{m_f} V_f^0 \sum_{f_f} V_{f_f}^0 \nabla W_{f,f_f}^{\mathrm{P}}(t) \\ & -\frac{\Delta t}{m_f} V_f^0 \sum_{f_b} V_{f_b}^0 \nabla W_{f,f_b}^{\mathrm{P}}(t) \end{aligned} \tag{32a}$$

$$\frac{\partial}{\partial p_f} \ \mathbf{v}_{f_f}(t+\Delta t) = -\frac{\Delta t}{m_f} V_{f_f}^0 V_f^0 \nabla W_{f_f,f}^{\mathrm{P}}(t) \tag{32b}$$

$$\frac{\partial}{\partial p_f} \ \mathbf{v}_{f_b}(t+\Delta t) = \mathbf{0}. \tag{32c}$$

A.1.1 Summary. One can compute α_f^{P} for a fluid particle f by iterating twice over its neighbors. First, $\partial/\partial p_f \mathbf{v}_f(t+\Delta t)$ is evaluated as shown in eq. (32a). Afterwards, $\partial/\partial p_f V_f^{\mathrm{err}}$ can be computed using eq. (31) where $\partial/\partial p_f \mathbf{v}_{f_f}(t+\Delta t)$ and $\partial/\partial p_f \mathbf{v}_{f_b}(t+\Delta t)$ are computed on the fly with eqs. (32b) and (32c). The whole procedure is also summarized in algorithm 2.

```
1 foreach fluid particle $f$ do
2     Compute $\frac{\partial}{\partial p_f} \mathbf{v}_f(t+\Delta t)$          ▹ eq. (32a)
3     Compute $\frac{\partial}{\partial p_f} V_f^{\mathrm{err}}(t+\Delta t)$   ▹ eqs. (31), (32b) and (32c)
4     Compute $\alpha_f^{\mathrm{P}}$                                        ▹ eq. (30)
```

Algorithm 2: The computation procedure for all α_f^{P}

A.2 Surface Tension

Similar to α_f^{P}, α_f^{ST} governs the step size used in a fixed-point iteration of surface tension forces $\mathbf{F}^{\mathrm{ST}}$ described in eq. (18b). In a standard Jacobi solver, α_f^{ST} is chosen as

$$\alpha_f^{\mathrm{ST}} = \left(\frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} \ \left(\mathbf{F}_f^{\mathrm{ST}} + \frac{\partial}{\partial \mathbf{x}_f} E(t+\Delta t) \right) \right)^{-1}. \tag{33}$$

This works well when assuming that the solver is able to converge exactly. In reality however, this is rarely the case. Typically, we exit the solver loop shown in algorithms 1 and 5 when the remaining residual is sufficiently small, even though no exact solution has yet been found. In the case of surface tension, this residual can cause forces $\mathbf{F}^{\mathrm{ST}}$ to not conserve momentum exactly. We can prevent the violation of momentum conservation by ensuring that the Jacobi update for surface tension forces $\mathbf{F}^{\mathrm{ST}}$ uses an equal step size for all fluid particles f, which is the case if α_f^{ST} is given the same value for all f. To make sure that the solving process remains stable, in our implementation we choose α^{ST} for all fluid particles to be equal to the lowest α_f^* of all fluid particles f:

$$\alpha^{\mathrm{ST}} = \min_f \alpha_f^* \tag{34a}$$

$$\alpha_f^* = \left(\frac{1}{3} \mathrm{tr} \left[\frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} \ \left(\mathbf{F}_f^{\mathrm{ST}} + \frac{\partial}{\partial \mathbf{x}_f} E(t+\Delta t) \right) \right] \right)^{-1} \tag{34b}$$

where tr $(\mathbf{x})$ returns the trace of $\mathbf{x}$. Note that in contrast to eq. (33), we consider the trace to obtain a scalar value whose magnitude can be compared in order to find the smallest α_f^*. In our experiments we found that α_f^* is typically close to one for all fluid particles f. Thus, by using the smallest α_f^*, only an insignificant amount of solver convergence speed is sacrificed. To compute α^{ST}, we reconsider eqs. (20) and (22) to (24):

$$\begin{aligned} &\frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} \left(\mathbf{F}_f^{\mathrm{ST}} + \frac{\partial}{\partial \mathbf{x}_f} E(t+\Delta t) \right) \\ =& \ \mathbf{I} + \sum_{f_f} \left(A_f^0 V_{f_f}^0 Y_f^{\mathrm{FV}} \frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} S_f^{\mathrm{FV}}(t+\Delta t) \right. \\ & \qquad \left. + A_{f_f}^0 V_f^0 Y_{f_f}^{\mathrm{FV}} \frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} S_{f_f}^{\mathrm{FV}}(t+\Delta t) \right) \nabla W_{f,f_f}^{\mathrm{ST}}(t) \\ & + \sum_{f_f \notin F} \left(- A_f^0 V_{f_f}^0 Y_f^{\mathrm{FF}} \frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} S_f^{\mathrm{FF}}(t+\Delta t) \right. \\ & \qquad \left. - A_{f_f}^0 V_f^0 Y_{f_f}^{\mathrm{FF}} \frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} S_{f_f}^{\mathrm{FF}}(t+\Delta t) \right) \nabla W_{f,f_f}^{\mathrm{ST}}(t) \\ & + \sum_{f_b} \left(A_f^0 V_{f_b}^0 Y_f^{\mathrm{FV}} \frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} S_f^{\mathrm{FV}}(t+\Delta t) \right. \\ & \qquad \left. + A_{f_b}^0 V_f^0 Y_{f_b}^{\mathrm{BV}} \frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} S_{f_b}^{\mathrm{BV}}(t+\Delta t) \right) \nabla W_{f,f_b}^{\mathrm{ST}}(t) \\ & + \sum_{f_b} \left(- A_f^0 V_{f_b}^0 Y_f^{\mathrm{FB}} \frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} S_f^{\mathrm{FB}}(t+\Delta t) \right. \\ & \qquad \left. - A_{f_b}^0 V_f^0 Y_{f_b}^{\mathrm{BF}} \frac{\partial}{\partial \mathbf{F}_f^{\mathrm{ST}}} S_{f_b}^{\mathrm{BF}}(t+\Delta t) \right) \nabla W_{f,f_b}^{\mathrm{ST}}(t). \end{aligned} \tag{35}$$

The exact derivative of $S(t+\Delta t)$ is given by

$$\frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}} S(t+\Delta t) = \begin{cases} 0 & \text{if } C(t+\Delta t) < 0 \\ \left(\frac{1}{\sqrt{C^2(t+\Delta t)+\epsilon^2}} - \frac{C^2(t+\Delta t)}{(C^2(t+\Delta t)+\epsilon^2)^{\frac{3}{2}}} \right) \frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}} C(t+\Delta t) & \text{otherwise.} \end{cases} \tag{36}$$

However, since $S(t+\Delta t)$ is nonlinear in $C(t+\Delta t)$, its derivative changes with $C(t+\Delta t)$. In order to obtain a value for α^{ST} that is constant during the solving process, in our implementation we approximate $C(t+\Delta t)$ with $C(t)$:

$$\frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}_f} S_j(t+\Delta t) \approx \begin{cases} 0 & \text{if } C_j(t) < 0 \\ \left(\frac{1}{\sqrt{C_j^2(t)+\epsilon^2}} - \frac{C_j^2(t)}{\left(C_j^2(t)+\epsilon^2\right)^{\frac{3}{2}}} \right) \frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}_f} C_j(t+\Delta t) & \text{otherwise.} \end{cases} \tag{37}$$

where j acts as a placeholder for f, f_f and f_b. The only unknown terms remaining are $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f C_f(t+\Delta t)$ for the fluid-vapor interface (FV), the fluid-fluid interface (FF) and the fluid-boundary interface (FB), as well as $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f C_{f_f}(t+\Delta t)$ with $f_f \notin F$ for the fluid-vapor interface (FV) and the fluid-fluid interface (FF), and last $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f C_{f_b}(t+\Delta t)$ for the boundary-vapor (BV) and the boundary-fluid interface (BF):

$$\frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}_f} C^{\mathrm{FV}}_f(t+\Delta t) = -\frac{\Delta t^2}{m_f} \sum_{f_f} V^0_{f_f} \nabla W^{\mathrm{ST}}_{f,f_f} - \frac{\Delta t^2}{m_f} \sum_{f_b} V^0_{f_b} \nabla W^{\mathrm{ST}}_{f,f_b} \tag{38a}$$

$$\frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}_f} C^{\mathrm{FF}}_f(t+\Delta t) = \frac{\Delta t^2}{m_f} \sum_{f_f \notin F} V^0_{f_f} \nabla W^{\mathrm{ST}}_{f,f_f}(t) \tag{38b}$$

$$\frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}_f} C^{\mathrm{FB}}_f(t+\Delta t) = \frac{\Delta t^2}{m_f} \sum_{f_b} V^0_{f_b} \nabla W^{\mathrm{ST}}_{f,f_b}(t) \tag{38c}$$

$$\frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}_f} C^{\mathrm{FV}}_{f_f}(t+\Delta t) = \frac{\Delta t^2}{m_f} V^0_f \nabla W^{\mathrm{ST}}_{f_f,f}(t) \tag{38d}$$

$$\frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}_f} C^{\mathrm{FF}}_{f_f}(t+\Delta t) = -\frac{\Delta t^2}{m_f} V^0_f \nabla W^{\mathrm{ST}}_{f_f,f}(t) \tag{38e}$$

$$\frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}_f} C^{\mathrm{BV}}_{f_b}(t+\Delta t) = \frac{\Delta t^2}{m_f} V^0_f \nabla W^{\mathrm{ST}}_{f_b,f}(t) \tag{38f}$$

$$\frac{\partial}{\partial \mathbf{F}^{\mathrm{ST}}_f} C^{\mathrm{BF}}_{f_b}(t+\Delta t) = -\frac{\Delta t^2}{m_f} V^0_f \nabla W^{\mathrm{ST}}_{f_b,f}(t). \tag{38g}$$

Note that eq. (38e) is only true for particles $f_f \notin F$, but looking at eq. (35) it is clear that we only ever evaluate $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f C^{\mathrm{FF}}_{f_f}(t+\Delta t)$ for $f_f \notin F$.

A.2.1 Summary. Similar to pressure, we can compute α^*_f for a particle f by iterating twice over its neighbors. First, the terms $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f C^{\mathrm{FV}}_f(t+\Delta t)$, $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f C^{\mathrm{FF}}_f(t+\Delta t)$ and $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f C^{\mathrm{FB}}_f(t+\Delta t)$ are evaluated using eqs. (38a) to (38c). They can be directly transformed into $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f S^{\mathrm{FV}}_f(t+\Delta t)$, $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f S^{\mathrm{FF}}_f(t+\Delta t)$ and $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f S^{\mathrm{FB}}_f(t+\Delta t)$ with eq. (37). In the second step, we can then evaluate eq. (35) where the expressions $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f S^{\mathrm{FV}}_{f_f}(t+\Delta t)$ and $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f S^{\mathrm{FF}}_{f_f}(t+\Delta t)$ as well as $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f S^{\mathrm{BV}}_{f_b}(t+\Delta t)$ and $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f S^{\mathrm{BF}}_{f_b}(t+\Delta t)$ are computed on the fly using eqs. (37) and (38d) to (38g). Last, we search for the smallest α^*_f to find α^{ST} as shown in eq. (34). The whole procedure is again summarized in algorithm 3.

```
1 foreach fluid particle $f$ do
2     Compute $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f S^{\mathrm{FV}}_f(t+\Delta t)$      ▷ eqs. (37) and (38a)
3     Compute $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f S^{\mathrm{FF}}_f(t+\Delta t)$      ▷ eqs. (37) and (38b)
4     Compute $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f S^{\mathrm{FB}}_f(t+\Delta t)$      ▷ eqs. (37) and (38c)
5     Compute $\alpha^*_f$      ▷ eqs. (34b), (35), (37) and (38d) to (38g)
6 $\alpha^{\mathrm{ST}} \leftarrow \min_f \alpha^*_f$
```

Algorithm 3: The computation procedure for α^{ST}. All quantities that are explicitly mentioned are computed and temporarily stored as they are repeatedly used in subsequent steps. Quantities not mentioned, such as $\partial/\partial \mathbf{F}^{\mathrm{ST}}_f C^{\mathrm{FV}}_{f_f}(t+\Delta t)$ for example, are computed on the fly as soon as they are required.

A.3 Friction

The update of friction forces during a Jacobi step as displayed in eq. (18c) requires a value for α^{F}_f. This value needs to be a scalar [Probst and Teschner 2023] and should relate the change in tangential relative velocity between fluid particle f and the boundary to its friction force $\mathbf{F}^{\mathrm{F}}_f$. Note that even though we directly solve for forces $\mathbf{F}^{\mathrm{F}}_f$ as we did in the surface tension computation, since momentum conservation is less of a concern when interacting with the static boundary, small residual errors in the friction computation are not a serious problem. This is why for the friction solver, we are able to use an individual value α^{F}_f that is optimal for each respective f. In our implementation, we set

$$\alpha^{\mathrm{F}}_f = \left(\frac{1}{3} \mathrm{tr}\left[\frac{\partial}{\partial \mathbf{F}^{\mathrm{F}}_f} \mathbf{v}^{\mathrm{tang}}_f(t+\Delta t) \right] \right)^{-1} \tag{39}$$

where $\partial/\partial \mathbf{F}^{\mathrm{F}}_f \mathbf{v}^{\mathrm{tang}}_f(t+\Delta t)$ is computed with

$$\partial/\partial \mathbf{F}^{\mathrm{F}}_f \mathbf{v}^{\mathrm{tang}}_f(t+\Delta t) = \left(\mathbf{I} - \frac{\mathbf{F}^N_f \mathbf{F}^{N\top}_f}{|\mathbf{F}^N_f|^2} \right) \partial/\partial \mathbf{F}^{\mathrm{F}}_f \mathbf{v}^{\mathrm{rel}}_f(t+\Delta t) \tag{40a}$$

$$\partial/\partial \mathbf{F}^{\mathrm{F}}_f \mathbf{v}^{\mathrm{rel}}_f(t+\Delta t) = \frac{\Delta t^2}{m_f} \mathbf{I} \sum_{f_b} V^0_{f_b} W^{\mathrm{F}}_{f,f_b}(t) \tag{40b}$$

as shown in eq. (26). We mentioned earlier in section 4.2.3 that the magnitude of $\mathbf{F}^N_f$ changes during the solving procedure. Its direction on the other hand stays constant, allowing us to precompute α^{F}_f for the whole solving process. The computation of α^{F}_f is again summarized in algorithm 4.

```
foreach fluid particle $f$ do
    Compute $\partial/\partial\mathbf{F}^{\mathrm{F}}_f\mathbf{v}^{\mathrm{rel}}_f(t+\Delta t)$        ▷ eq. (40b)
    Compute $\partial/\partial\mathbf{F}^{\mathrm{F}}_f\mathbf{v}^{\mathrm{tang}}_f(t+\Delta t)$       ▷ eqs. (14) and (40a)
    Compute $\alpha^{\mathrm{F}}_f$                                   ▷ eq. (39)
```

Algorithm 4: The computation procedure for all α^{F}_f

B Nonsmooth Nonlinear Conjugate Gradient

Jacobi methods are known for their simplicity and flexibility, but they also often suffer from slow convergence speed, especially when applied to poorly conditioned problems [Cottle et al. 2009; Erleben 2013; Poulsen et al. 2010; Silcowitz et al. 2009]. To improve convergence behavior, Silcowitz-Hansen et al. [2010] proposed a nonsmooth nonlinear conjugate gradient (NNCG) method that is used on top of the Jacobi iterates shown in algorithm 1. The NNCG algorithm considers differences between iterates $k+1$ and k of pressure $\mathbf{p}$, surface tension $\mathbf{F}^{\mathrm{ST}}$ and friction $\mathbf{F}^{\mathrm{F}}$ as residuals $\mathbf{r}$:

$$\begin{aligned}\left(\mathbf{r}^{\mathrm{P}}_k,\mathbf{r}^{\mathrm{ST}}_k,\mathbf{r}^{\mathrm{F}}_k\right) &\equiv \left(\mathbf{p}_{k+1}-\mathbf{p}_k,\mathbf{F}^{\mathrm{ST}}_{k+1}-\mathbf{F}^{\mathrm{ST}}_k,\mathbf{F}^{\mathrm{F}}_{k+1}-\mathbf{F}^{\mathrm{F}}_k\right)\\ &= \mathrm{Jacobi}\left(\mathbf{p}_k,\mathbf{F}^{\mathrm{ST}}_k,\mathbf{F}^{\mathrm{F}}_k\right)-\left(\mathbf{p}_k,\mathbf{F}^{\mathrm{ST}}_k,\mathbf{F}^{\mathrm{F}}_k\right).\end{aligned} \tag{41}$$

Here, the function Jacobi returns the iterates of $\mathbf{p}$, $\mathbf{F}^{\mathrm{ST}}$ and $\mathbf{F}^{\mathrm{F}}$ after applying one Jacobi solver iteration as described in lines 6 to 22. This allows us to define a function f as

$$f(\mathbf{r}^{\mathrm{P}}_k,\mathbf{r}^{\mathrm{ST}}_k,\mathbf{r}^{\mathrm{F}}_k) \equiv \frac{1}{2}|\mathbf{r}^{\mathrm{P}}_k|^2+\frac{1}{2}|\mathbf{r}^{\mathrm{ST}}_k|^2+\frac{1}{2}|\mathbf{r}^{\mathrm{F}}_k|^2. \tag{42}$$

Note that searching for roots of f is equivalent to solving the original fixed-point problem from eq. (18) by finding values for $\mathbf{p}_k$, $\mathbf{F}^{\mathrm{ST}}_k$ and $\mathbf{F}^{\mathrm{F}}_k$ that satisfy

$$\mathrm{Jacobi}\left(\mathbf{p}_k,\mathbf{F}^{\mathrm{ST}}_k,\mathbf{F}^{\mathrm{F}}_k\right)-\left(\mathbf{p}_k,\mathbf{F}^{\mathrm{ST}}_k,\mathbf{F}^{\mathrm{F}}_k\right)=\mathbf{0}. \tag{43}$$

The Fletcher-Reeves nonlinear conjugate gradient method can be employed to find a local minimum of f since it only requires information about the gradient $\boldsymbol{\nabla}f$ with respect to $\mathbf{p}_k$, $\mathbf{F}^{\mathrm{ST}}_k$ and $\mathbf{F}^{\mathrm{F}}_k$ [Andrews and Erleben 2021; Nocedal and Wright 2006; Silcowitz-Hansen et al. 2010], which is equal to the negative residual vector $-(\mathbf{r}^{\mathrm{P}}_k,\mathbf{r}^{\mathrm{ST}}_k,\mathbf{r}^{\mathrm{F}}_k)$. The final NNCG solver algorithm used in our simulation is illustrated in algorithm 5.

```
$\mathbf{p}_0 \leftarrow \mathbf{0}$
$\mathbf{F}^{\mathrm{ST}}_0 \leftarrow \mathbf{0}$
$\mathbf{F}^{\mathrm{F}}_0 \leftarrow \mathbf{0}$
$\mathbf{p}_1,\mathbf{F}^{\mathrm{ST}}_1,\mathbf{F}^{\mathrm{F}}_1 \leftarrow \mathrm{Jacobi}\,(\mathbf{p}_0,\mathbf{F}^{\mathrm{ST}}_0,\mathbf{F}^{\mathrm{F}}_0)$        ▷ algorithm 1
$\mathbf{r}^{\mathrm{P}}_0 \leftarrow \mathbf{p}_1-\mathbf{p}_0$
$\mathbf{r}^{\mathrm{ST}}_0 \leftarrow \mathbf{F}^{\mathrm{ST}}_1-\mathbf{F}^{\mathrm{ST}}_0$
$\mathbf{r}^{\mathrm{F}}_0 \leftarrow \mathbf{F}^{\mathrm{F}}_1-\mathbf{F}^{\mathrm{F}}_0$
$\mathbf{s}^{\mathrm{P}}_0 \leftarrow -\mathbf{r}^{\mathrm{P}}_0$
$\mathbf{s}^{\mathrm{ST}}_0 \leftarrow -\mathbf{r}^{\mathrm{ST}}_0$
$\mathbf{s}^{\mathrm{F}}_0 \leftarrow -\mathbf{r}^{\mathrm{F}}_0$
$k \leftarrow 1$
while not converged do
    $\mathbf{p}_{k+1},\mathbf{F}^{\mathrm{ST}}_{k+1},\mathbf{F}^{\mathrm{F}}_{k+1} \leftarrow \mathrm{Jacobi}\,(\mathbf{p}_k,\mathbf{F}^{\mathrm{ST}}_k,\mathbf{F}^{\mathrm{F}}_k)$   ▷ algorithm 1
    $\mathbf{r}^{\mathrm{P}}_k \leftarrow \mathbf{p}_{k+1}-\mathbf{p}_k$
    $\mathbf{r}^{\mathrm{ST}}_k \leftarrow \mathbf{F}^{\mathrm{ST}}_{k+1}-\mathbf{F}^{\mathrm{ST}}_k$
    $\mathbf{r}^{\mathrm{F}}_k \leftarrow \mathbf{F}^{\mathrm{F}}_{k+1}-\mathbf{F}^{\mathrm{F}}_k$
    $\beta^{\mathrm{P}} \leftarrow \mathbf{r}^{\mathrm{P}}_k/\mathbf{r}^{\mathrm{P}}_{k-1}$
    $\beta^{\mathrm{ST}} \leftarrow \mathbf{r}^{\mathrm{ST}}_k/\mathbf{r}^{\mathrm{ST}}_{k-1}$
    $\beta^{\mathrm{F}} \leftarrow \mathbf{r}^{\mathrm{F}}_k/\mathbf{r}^{\mathrm{F}}_{k-1}$
    if $\beta^{\mathrm{P}} > 1$ then
        $\mathbf{s}^{\mathrm{P}}_k \leftarrow \mathbf{0}$
    else
        $\mathbf{p}_{k+1} \leftarrow \mathbf{p}_{k+1}+\beta^{\mathrm{P}}\mathbf{s}^{\mathrm{P}}_{k-1}$
        $\mathbf{s}^{\mathrm{P}}_k \leftarrow \beta^{\mathrm{P}}\mathbf{s}^{\mathrm{P}}_{k-1}-\mathbf{r}^{\mathrm{P}}_k$
    if $\beta^{\mathrm{ST}} > 1$ then
        $\mathbf{s}^{\mathrm{ST}}_k \leftarrow \mathbf{0}$
    else
        $\mathbf{F}^{\mathrm{ST}}_{k+1} \leftarrow \mathbf{F}^{\mathrm{ST}}_{k+1}+\beta^{\mathrm{ST}}\mathbf{s}^{\mathrm{ST}}_{k-1}$
        $\mathbf{s}^{\mathrm{ST}}_k \leftarrow \beta^{\mathrm{ST}}\mathbf{s}^{\mathrm{ST}}_{k-1}-\mathbf{r}^{\mathrm{ST}}_k$
    if $\beta^{\mathrm{F}} > 1$ then
        $\mathbf{s}^{\mathrm{F}}_k \leftarrow \mathbf{0}$
    else
        $\mathbf{F}^{\mathrm{F}}_{k+1} \leftarrow \mathbf{F}^{\mathrm{F}}_{k+1}+\beta^{\mathrm{F}}\mathbf{s}^{\mathrm{F}}_{k-1}$
        $\mathbf{s}^{\mathrm{F}}_k \leftarrow \beta^{\mathrm{F}}\mathbf{s}^{\mathrm{F}}_{k-1}-\mathbf{r}^{\mathrm{F}}_k$
    $k \leftarrow k+1$
```

Algorithm 5: Our implementation of the NNCG solver proposed by Silcowitz-Hansen et al. [2010] to simultaneously compute strongly coupled pressure $\mathbf{p}$, surface tension $\mathbf{F}^{\mathrm{ST}}$ and friction $\mathbf{F}^{\mathrm{F}}$. The Jacobi iteration shown in algorithm 1 is used as a basic building block to estimate $\boldsymbol{\nabla}f$ which is required to find a local minimum of our objective function f (see eq. (42)).

C Conservation of Momentum

Here, we want to give a short proof that shows how our surface tension force $\mathbf{F}^{\mathrm{ST}}$ described in section 3.2.1 conserves momentum exactly. Interactions with kinematic boundaries are ignored since in this case momentum conservation is violated on purpose by not considering forces acting on the boundary. Starting with conservation of linear momentum, we need to show that

$$\sum_f \mathbf{F}^{\mathrm{ST}}_f = \mathbf{0}. \tag{44}$$

Plugging in the definition of $\mathbf{F}^{\mathrm{ST}}$ from eq. (11) gives us

$$\begin{aligned}\mathbf{0}=\sum_f \mathbf{F}^{\mathrm{ST}}_f=\sum_f\Bigg[&\sum_{ff}\left(A^0_f V^0_{ff}\gamma^{\mathrm{FV}}_f S^{\mathrm{FV}}_f + A^0_{ff}V^0_f\gamma^{\mathrm{FV}}_{ff}S^{\mathrm{FV}}_{ff}\right)\boldsymbol{\nabla}W^{\mathrm{ST}}_{f,ff}\\ &+\sum_{ff\neq f}\left(-A^0_f V^0_{ff}\gamma^{\mathrm{FF}}_f S^{\mathrm{FF}}_f - A^0_{ff}V^0_f\gamma^{\mathrm{FF}}_{ff}S^{\mathrm{FF}}_{ff}\right)\boldsymbol{\nabla}W^{\mathrm{ST}}_{f,ff}\Bigg].\end{aligned} \tag{45}$$

For $f_f = f$, the contribution to the double summation is zero as $\nabla W^{\mathrm{ST}}_{f,f} = \mathbf{0}$. For each fluid particle neighbor pair i and j, the double summations always consider the contribution for $f = i$ and $f_f = j$ as well as for $f = j$ and $f_f = i$ exactly once. Since $\nabla W^{\mathrm{ST}}_{f,f_f} = -\nabla W^{\mathrm{ST}}_{f_f,f}$ and as such

$$\begin{aligned}&\left(A^0_f V^0_{f_f} \Upsilon^{\mathrm{FV}}_f S^{\mathrm{FV}}_f + A^0_{f_f} V^0_f \Upsilon^{\mathrm{FV}}_{f_f} S^{\mathrm{FV}}_{f_f}\right) \nabla W^{\mathrm{ST}}_{f,f_f}\\ &+\left(-A^0_f V^0_{f_f} \Upsilon^{\mathrm{FF}}_f S^{\mathrm{FF}}_f - A^0_{f_f} V^0_f \Upsilon^{\mathrm{FF}}_{f_f} S^{\mathrm{FF}}_{f_f}\right) \nabla W^{\mathrm{ST}}_{f,f_f}\\ =& -\left(A^0_{f_f} V^0_f \Upsilon^{\mathrm{FV}}_{f_f} S^{\mathrm{FV}}_{f_f} + A^0_f V^0_{f_f} \Upsilon^{\mathrm{FV}}_f S^{\mathrm{FV}}_f\right) \nabla W^{\mathrm{ST}}_{f_f,f}\\ &-\left(-A^0_{f_f} V^0_f \Upsilon^{\mathrm{FF}}_{f_f} S^{\mathrm{FF}}_{f_f} - A^0_f V^0_{f_f} \Upsilon^{\mathrm{FF}}_f S^{\mathrm{FF}}_f\right) \nabla W^{\mathrm{ST}}_{f_f,f},\end{aligned} \tag{46}$$

the two contributions add up to zero. We see that each contribution to the double summation is either zero for $f = f_f$, or is canceled out by the contribution to the double summation with flipped indices. Thus, the double summation in eq. (45) indeed equals zero and linear momentum is conserved. Angular momentum is conserved by surface tension forces if the total torque exerted on all fluid particles equals zero for an arbitrary $\mathbf{r} \in \mathbb{R}^3$:

$$\sum_f \left(\mathbf{x}_f - \mathbf{r}\right) \times \mathbf{F}^{\mathrm{ST}}_f = \mathbf{0}. \tag{47}$$

We plug in the definition of our surface tension forces $\mathbf{F}^{\mathrm{ST}}$ (eq. (11)) which gives us

$$\begin{aligned}\mathbf{0} &= \sum_f \left(\mathbf{x}_f - \mathbf{r}\right) \times \mathbf{F}^{\mathrm{ST}}_f\\ &= \sum_f \Bigg[\sum_{f_f} \left(A^0_f V^0_{f_f} \Upsilon^{\mathrm{FV}}_f S^{\mathrm{FV}}_f + A^0_{f_f} V^0_f \Upsilon^{\mathrm{FV}}_{f_f} S^{\mathrm{FV}}_{f_f}\right) \left(\mathbf{x}_f - \mathbf{r}\right) \times \nabla W^{\mathrm{ST}}_{f,f_f}\\ &\qquad + \sum_{f_f \notin F} \left(-A^0_f V^0_{f_f} \Upsilon^{\mathrm{FF}}_f S^{\mathrm{FF}}_f - A^0_{f_f} V^0_f \Upsilon^{\mathrm{FF}}_{f_f} S^{\mathrm{FF}}_{f_f}\right) \left(\mathbf{x}_f - \mathbf{r}\right) \times \nabla W^{\mathrm{ST}}_{f,f_f} \Bigg].\end{aligned} \tag{48}$$

It is easy to see that for $f = f_f$ the contribution to the double sum again equals zero due to $\nabla W^{\mathrm{ST}}_{f,f} = \mathbf{0}$. Similar to before, we now want to demonstrate that for each neighboring fluid particle pair, their two respective contributions to the double summation cancel out each other. We do this by showing

$$\begin{aligned}&\left(A^0_f V^0_{f_f} \Upsilon^{\mathrm{FV}}_f S^{\mathrm{FV}}_f + A^0_{f_f} V^0_f \Upsilon^{\mathrm{FV}}_{f_f} S^{\mathrm{FV}}_{f_f}\right) \left(\mathbf{x}_f - \mathbf{r}\right) \times \nabla W^{\mathrm{ST}}_{f,f_f}\\ &+\left(-A^0_f V^0_{f_f} \Upsilon^{\mathrm{FF}}_f S^{\mathrm{FF}}_f - A^0_{f_f} V^0_f \Upsilon^{\mathrm{FF}}_{f_f} S^{\mathrm{FF}}_{f_f}\right) \left(\mathbf{x}_f - \mathbf{r}\right) \times \nabla W^{\mathrm{ST}}_{f,f_f}\\ =& -\left(A^0_{f_f} V^0_f \Upsilon^{\mathrm{FV}}_{f_f} S^{\mathrm{FV}}_{f_f} + A^0_f V^0_{f_f} \Upsilon^{\mathrm{FV}}_f S^{\mathrm{FV}}_f\right) \left(\mathbf{x}_{f_f} - \mathbf{r}\right) \times \nabla W^{\mathrm{ST}}_{f_f,f}\\ &-\left(-A^0_{f_f} V^0_f \Upsilon^{\mathrm{FF}}_{f_f} S^{\mathrm{FF}}_{f_f} - A^0_f V^0_{f_f} \Upsilon^{\mathrm{FF}}_f S^{\mathrm{FF}}_f\right) \left(\mathbf{x}_{f_f} - \mathbf{r}\right) \times \nabla W^{\mathrm{ST}}_{f_f,f}.\end{aligned} \tag{49}$$

Flipping the kernel gradients with $\nabla W^{\mathrm{ST}}_{f_f f} = -\nabla W^{\mathrm{ST}}_{f f_f}$ and moving all terms to one side gives us

$$\begin{aligned}\mathbf{0} =& \left(A^0_f V^0_{f_f} \Upsilon^{\mathrm{FV}}_f S^{\mathrm{FV}}_f + A^0_{f_f} V^0_f \Upsilon^{\mathrm{FV}}_{f_f} S^{\mathrm{FV}}_{f_f}\right) \left(\mathbf{x}_f - \mathbf{r} - \mathbf{x}_{f_f} + \mathbf{r}\right) \times \nabla W^{\mathrm{ST}}_{f,f_f}\\ &+\left(-A^0_f V^0_{f_f} \Upsilon^{\mathrm{FF}}_f S^{\mathrm{FF}}_f - A^0_{f_f} V^0_f \Upsilon^{\mathrm{FF}}_{f_f} S^{\mathrm{FF}}_{f_f}\right) \left(\mathbf{x}_f - \mathbf{r} - \mathbf{x}_{f_f} + \mathbf{r}\right) \times \nabla W^{\mathrm{ST}}_{f,f_f}.\end{aligned} \tag{50}$$

Since kernel gradients $\nabla W^{\mathrm{ST}}_{f,f_f}$ by construction point in the same or opposite direction as $\mathbf{x}_f - \mathbf{x}_{f_f}$, the cross products equal zero. Thus, eqs. (49) and (50) are valid and each contribution to the total torque computed in eq. (48) adds up to zero per particle pair, causing the total torque due to surface tension to be equal to zero as well. In summary, we were able to show that both linear and angular momentum are exactly conserved by our surface tension force.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009