Implicit Surface Tension for SPH Fluid Simulation

STEFAN RHYS JESKE, RWTH Aachen University, Germany
LUKAS WESTHOFEN, RWTH Aachen University, Germany
FABIAN LÖSCHNER, RWTH Aachen University, Germany
JOSÉ ANTONIO FERNÁNDEZ-FERNÁNDEZ, RWTH Aachen University, Germany
JAN BENDER, RWTH Aachen University, Germany

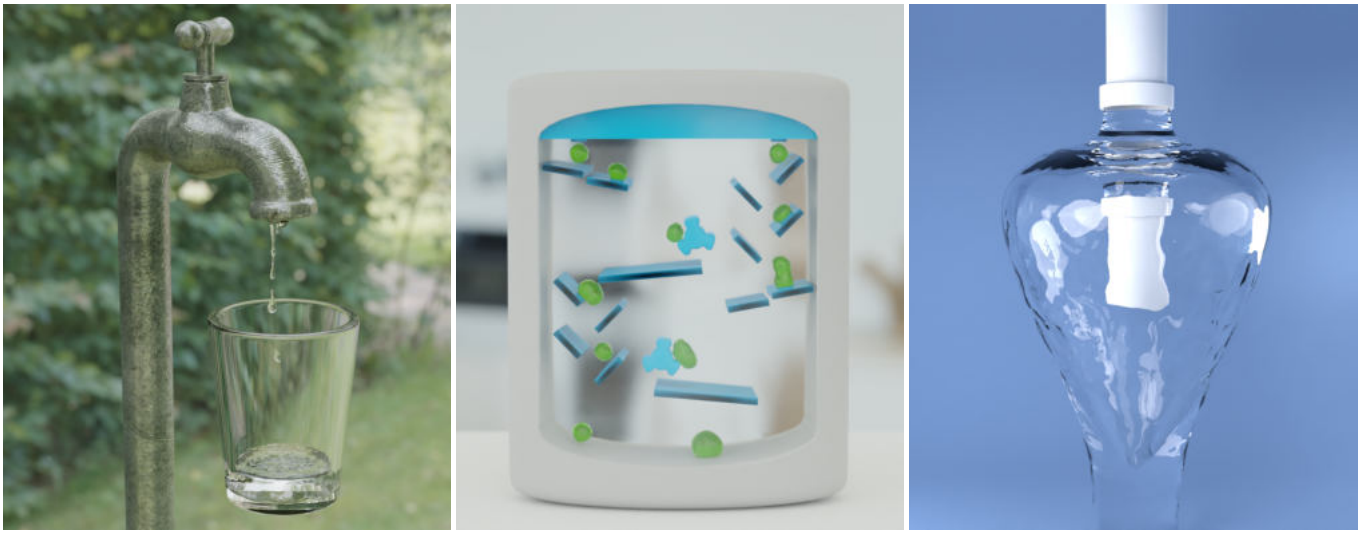

Fig. 1. *Left:* A dripping water faucet showing realistic stream and droplet formation for small scale water effects. *Middle:* A child's toy in which droplets fall through a maze of obstacles due to gravity. A reservoir at the top of the toy causes droplets to be formed through funnels, falling through the maze and recombining into a single fluid mass at the bottom. *Right:* A water-bell, which is formed due to surface tension when two opposite and vertical water streams meet. Our surface tension method force is able to maintain the shape, while the surface at its thinnest point is just 2-3 particles thick.

The numerical simulation of surface tension is an active area of research in many different fields of application and has been attempted using a wide range of methods. Our contribution is the derivation and implementation of an implicit cohesion force based approach for the simulation of surface tension effects using the Smoothed Particle Hydrodynamics (SPH) method. We define a continuous formulation inspired by the properties of surface tension at the molecular scale which is spatially discretized using SPH. An adapted variant of the linearized backward Euler method is used for time discretization, which we also strongly couple with an implicit viscosity model. Finally, we extend our formulation with adhesion forces for interfaces with rigid objects.

Existing SPH approaches for surface tension in computer graphics are mostly based on explicit time integration, thereby lacking in stability for challenging settings. We compare our implicit surface tension method to these approaches and further evaluate our model on a wider variety of complex scenarios, showcasing its efficacy and versatility. Among others, these include but are not limited to simulations of a water crown, a dripping faucet and a droplet-toy.

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

Additional Key Words and Phrases: surface tension, cohesion, adhesion, smoothed particle hydrodynamics, fluid simulation

ACM Reference Format:
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.* 1, 1, Article 1 (January 2023), 14 pages. https://doi.org/10.1145/3631936

1 INTRODUCTION

Surface tension effects can be readily observed in everyday situations at home or in nature. A dripping water faucet, the formation and merging of soap bubbles or the movement of water striders on a pond are just a few examples which are impossible without surface tension. There exist many use cases for simulating these effects in the field of computer animation, whether it is for small scale water features, or artistic effects like droplets running down a soda can. Even outside these applications, modeling surface tension is crucial for specific engineering problems, for example in additive

Authors' addresses: Stefan Rhys Jeske, jeske@cs.rwth-aachen.de, RWTH Aachen University, Aachen, Germany; Lukas Westhofen, l.westhofen@cs.rwth-aachen.de, RWTH Aachen University, Aachen, Germany; Fabian Löschner, loeschner@cs.rwth-aachen.de, RWTH Aachen University, Aachen, Germany; José Antonio Fernández-Fernández, fernandez@cs.rwth-aachen.de, RWTH Aachen University, Aachen, Germany; Jan Bender, bender@cs.rwth-aachen.de, RWTH Aachen University, Aachen, Germany.

manufacturing. Overall, the numerical simulation of surface tension is an active area of research in various fields.

At its core, surface tension is the result of unbalanced molecular attraction forces at the fluid interface. This imbalance results in forces minimizing the surface area of the fluid and tensile stresses parallel to the surface, which can roughly be compared to a kind of "elastic" membrane. Of course this is a highly simplified description and actually incorporating surface tension in continuum mechanical fluid simulations poses several challenges. First of all, surface tension is not inherent to models based on the Navier-Stokes equations and has to be modelled explicitly in contrast to molecular simulations, where surface tension arises directly from the intermolecular potentials. Furthermore, depending on the choice of the numerical discretization method, the obvious fact that surface tension effects specifically apply to the surface of the fluid can cause nontrivial accuracy or stability issues as well as complicate the coupling with the rest of the fluid model.

Previous SPH surface tension methods in computer graphics have, for the most part, used explicit time integration. This becomes an issue when aiming to simulate cases with large surface tension coefficients, as they end up requiring small time steps in order to obtain a stable simulation. In addition, the simulation of certain effects containing surface tension, such as the formation of thin sheets during splashing, requires a delicate balance of forces and stability of surface tension that has not yet been shown using explicit SPH surface tension methods. As such, we aim in this paper to develop an implicit scheme for computing surface tension forces in order to improve simulation stability and to enable the use of large force coefficients without resorting to reducing time step sizes. To this end we present the following contributions.

First, we derive a cohesion force which is inspired by the balance of forces at the molecular scale. This force is defined such that it only acts on the surface of the fluid while cancelling out in the interior. This property is fulfilled by making use of the symmetric SPH kernel function, enabling our model to seamlessly integrate with existing SPH solvers. After discretizing with SPH we apply the backward Euler method for time integration, yielding a non-linear system of equations. For the system we subsequently propose a non-trivial linearization for improved solver performance.

Second, we extend our surface tension force by an adhesion term for interaction with rigid objects and further strongly couple it with a state-of-the-art implicit viscosity force [Weiler et al. 2018]. The strong coupling is achieved by solving a single monolithic linear system. This results in more stable simulations compared to solving surface tension and viscosity sequentially by avoiding coupling artifacts, which become more noticeable for cases with both larger surface tension and viscosity coefficients. The monolithic solver is able to directly find a solution satisfying both equations, instead of having to iterate between them or favoring one over the other by solving them in succession.

Finally, we demonstrate that using our method, we are able to simulate a large variety of surface tension effects. These range from cohesion dominant cases, such as droplet formation and fluid crowns, to adhesion dominant cases, such as covered spheres and surface wetting, as well as cases requiring a delicate balance between the two, such as a catenoid between two rigid tori. In Fig. 1, a selection of these complex surface tension effects is shown.

2 RELATED WORK

Over the years a number of different approaches have been developed to simulate surface tension using a variety of different discretizations. As the goal of our work is to present an improved and implicit surface tension model for SPH simulation, we first present a brief review of SPH surface tension approaches. Afterwards, we outline the broader context of how surface tension can be simulated using different discretization and simulation methods. Finally, we summarize the findings most relevant to our work.

2.1 Particle-Based Surface Tension

To this day, SPH fluid simulation [Desbrun and Gascuel 1996; Gingold and Monaghan 1977; Lucy 1977; Monaghan 1992] is an established method in a wide variety of fields, ranging from computational physics and engineering [Farrokhpanah et al. 2021; Jeske et al. 2022; Komen et al. 2020] to physical simulation and animation. Recent advances include the simulation of snow [Gissler et al. 2020], the simulation of deformable objects [Kugelstadt et al. 2021; Peer et al. 2018], and the strong coupling between fluids and rigid bodies [Gissler et al. 2019]. SPH simulation methods have made significant progress over the years, and as such there have also been a variety of methods attempting to capture the effects of surface tension. For a general overview of SPH methods in computer graphics, the reader is referred to the state-of-the-art reports by Koschier et al. [2022] and Ihmsen et al. [2014b]. An overview of surface tension formulations, as well as resulting numerical models is given by Popinet [2018].

At a microscopic scale, surface tension is known to originate from an imbalance of molecular forces at fluid interfaces [Nelkon 1969]. As SPH is a particle based Lagrangian discretization method, some works have attempted to emulate this molecular force by introducing attractive forces between particles [Becker and Teschner 2007; Clavet et al. 2005; Tartakovsky and Meakin 2005; Yang et al. 2016b, 2017]. These works have partially been documented to suffer from increased particle clustering at the surface [Huber et al. 2015]. Yang et al. [2016b; 2017] show that using a significantly larger support radius can alleviate the particle clustering problems in some of the previous works and propose an improved particle-particle interaction force for surface tension. They use a Jacobi iteration-based strong coupling between pressure and non-pressure forces (including surface tension). This however comes at the cost of significantly increased computational complexity.

At the macroscopic scale, surface tension is generally derived through the thermodynamic equilibrium across the fluid interface [Landau and Lifshitz 2013]. Derived from this is the Young-Laplace equation that describes forces acting on the surface of the fluid. Brackbill et al. [1992] reformulated this equation using a mollifier function for volumetric simulations. This is called the Continuum Surface Force (CSF) approach. This approach was initially adopted for SPH simulations by Morris [2000], and subsequently by several others [He et al. 2014; Hu and Adams 2006; Müller et al. 2003; Zhang 2010; Zorilla et al. 2020]. It integrates well with the SPH framework

and is able to produce comparable results to cohesion based approaches when tuned correctly. Nevertheless, the computation of the curvature with SPH is known to be prone to errors [Morris 2000], especially in the interior of the fluid, requiring the usage of non-trivial thresholds or explicit surface classification to alleviate this. Notably, Zorilla et al. [2020] present an improved, but computationally expensive approach for computing accurate curvature values. Finally, Akinci et al. [2013] present a hybrid approach utilizing both curvature (through non-normalized normal vectors) and cohesion forces aiming to combine the best of both methods. This method is further built upon by X.K. Wang et al. [2017], by coupling it with the IISPH method [Ihmsen et al. 2014a]. Comparisons of select models based on the microscale considerations (inter-particle force) and models based on the CSF approach are presented by Huber et al. [2015] and Yang et al. [2019].

2.2 Further Approaches

Grid-Based. Similar to Lagrangian particle methods, the fluid interfaces which are required for surface tension are often only implicitly defined in grid-based methods. The surfaces can be reconstructed by using an interface tracking technique such as the level-set method [Aanjaneya et al. 2013; Kang et al. 2008; Ni et al. 2020; Patkar et al. 2013; Zheng et al. 2009] or the volume of fluid method [Albadawi et al. 2013; Hong and Kim 2003, 2005]. However, while grid-based Eulerian methods allow for changing topology, the resolution is often constrained by the necessity to discretize a very large region outside of the actual region of interest. A recent work by Chen et al. [2020] addresses this issue by proposing a cut-cell method able to resolve the liquid surface at sub-grid scale. The level-set method has also been successfully applied to the simulation of soap bubbles [Kang et al. 2008] as well as air bubbles and bubble clusters [Zheng et al. 2009].

Mesh-Based. Lagrangian mesh based approaches typically facilitate accurate computation of surface tension forces by explicitly modeling the surface of the fluid. There are a number of mesh-based approaches incorporating surface tension effects focusing on surface only (non-volumetric) discretizations of fluids. These include the simulation of droplets covering other objects [Zhang et al. 2012], the simulation of viscous thin sheets [Batty et al. 2012] and threads [Bergou et al. 2010], thin films on other surfaces [Azencot et al. 2015] and surface only discretizations of bubbles and thin films [Da et al. 2015; Ishida et al. 2020, 2017]. Zhu et al. [2015; 2014] propose a codimensional simulation method to allow for changing surface topology. Finally, Misztal et al. [2014, 2012] derive and implement a fully coupled solver containing pressure, viscosity and implicit surface tension on moving unstructured tetrahedral meshes.

Hybrid. Hybrid methods typically aim to make use of Lagrangian particles and Eulerian grids together, combining benefits of both approaches to capture incredibly diverse effects. Zheng et al. [2015] apply a hybrid particle marker-and-cell grid to better capture free surface flows. Hong et al. [2008] propose a novel approach of using SPH to simulate air bubbles within an Eulerian grid based simulator. A versatile surface tension approach based on the minimization of an energy which is proportional to the surface area of the liquid is proposed by Hyde et al. [2020]. This approach can be used within the material-point method or particle-in-cell approach and has been further extended to model phenomena with spatially varying surface tension coefficients [Chen et al. 2021]. Boyd and Bridson [2012] propose a FLIP method which explicitly reconstructs the fluid surface for accurate surface tension computation. A hybrid method which uses a Lagrangian surface mesh instead of particles is introduced by Schroeder et al. [2012], in which the surface mesh is coupled to an Eulerian discretization. A similar approach has also been proposed by Ruan et al. [2021] and has been extended to strongly couple with solid objects interacting with the surface.

Recently, Xing et al. [2022] introduced a hybrid position-based method for simulating small-scale surface tension phenomena, by detecting surface particles, reconstructing a local triangle mesh and formulating surface tension constraints.

2.3 Summary

While a number of particle-based surface tension models have already been proposed, there have always been drawbacks and limitations to what effects a single method was able to capture. Most previous SPH based methods modelled surface tension with explicit forces, which can cause stability issues with large parameter values or large time steps. We propose an implicit surface tension method in SPH-based fluid simulation to address this problem. This includes an implicit approach for boundary adhesion, as well as strong coupling with an implicit velocity solver.

As opposed to the very recently published method of Xing et al. [2022], we do not have to resort to detecting surface particles explicitly, or potentially costly reconstructions of local meshes. Surface particles are implicitly detected using our proposed cohesion kernel and surface forces constructed such that surface area is reduced.

Compared to the approach of Yang et al. [2016b, 2017], we are able to implicitly integrate surface tension forces using comparatively small particle neighborhoods of only 30-40 particles. We also use conjugate gradient (CG) iteration to solve a linear system instead of nonlinear Jacobi iteration, which also has some more favorable convergence properties.

Ultimately, we show that the implicit nature of our method enables us to plausibly simulate a wide variety of surface tension phenomena within a purely particle-based surface tension model.

3 METHOD

As surface tension is fundamentally an effect caused by unbalanced molecular forces at the micro-scale (see Fig. 2), this results in a net force at the fluid interface. This can be thought of as "tightening" the surface, forming a kind of elastic membrane with the tendency to reduce the surface area of the interface.

In the following, we will derive our surface tension force from continuum micro-scale considerations. We chose this approach as the transfer of surface tension originating from molecular cohesive forces to the domain of SPH particle simulation is the more natural transition, resulting in a native formulation. Meanwhile, curvature based approaches require specialized solutions to prevent non-zero curvature values in the interior of the fluid. Inspired by the molecular cohesive forces we define a continuous model for surface tension

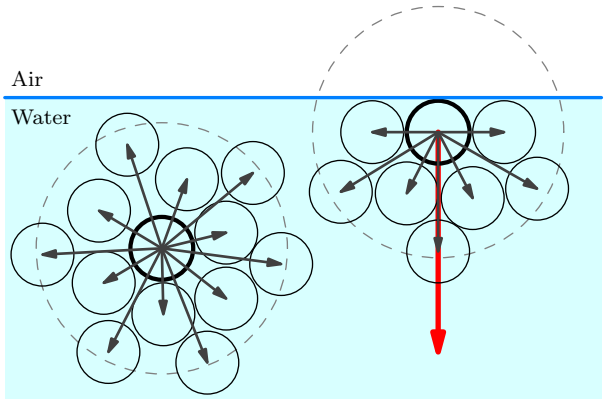

Fig. 2. Surface Tension force as a result of micro-scale force imbalances. Interior particles are completely surrounded in all directions, so that pairwise cohesive forces (grey arrows) add up to zero. Particles at the surface have no neighboring particles in a specific region, such that summation of pairwise cohesive forces (grey arrows) causes a net force (red arrow) in the direction of the fluid interior.

forces, which we discretize in space using SPH. For discretization in time we propose an adapted variant of the linearized backward Euler method. We strongly couple this linear system with implicit viscosity forces and further extend it to be able to include adhesion effects at solid boundaries.

We use the well-established operator splitting approach for SPH, which means that we separately compute and integrate pressure and non-pressure forces. This weak coupling between pressure and non-pressure forces is the current standard for SPH simulation [Ihmsen et al. 2014b; Koschier et al. 2022], and we have not found it to be prohibitive for the effects we aimed to simulate with our method, especially when using the implicit pressure solver DFSPH [Bender and Koschier 2017]. Nevertheless, we also present a short description of an iteration-based strong pressure coupling approach, as well as how it relates to the weak pressure-coupled algorithm in Sec. 3.5. A comparison and brief discussion between the coupling schemes is then provided in Sec. 4.1.

Finally, we propose a particular weighting function for our cohesion force approach, which relates how strongly particles interact with each other depending on their distance vector. We construct it in a way to prevent previously mentioned clustering artifacts at the surface.

3.1 Derivation

For our method we adopt the micro-scale assumption that the surface tension force results from a force imbalance at the fluid interface as shown in Fig. 2. We formalize this in the continuum as

$$\boldsymbol{f}_{st}(\boldsymbol{x}) = -\sigma \int_{\Omega} \frac{\boldsymbol{x}-\boldsymbol{y}}{||\boldsymbol{x}-\boldsymbol{y}||} W(\boldsymbol{x}-\boldsymbol{y};\tilde{h}) d\boldsymbol{y}, \tag{1}$$

where W is a compactly supported weighting function on the integration domain Ω, and $\sigma \in \mathbb{R}^+$ denotes the surface tension coefficient. Note that σ is not directly proportional to the otherwise commonly used surface tension parameters derived for curvature-based surface tension, although it shares the same units [N m^{-1}].

W further depends on the distance vector between evaluation point $\boldsymbol{x}$ and integration point $\boldsymbol{y}$, and the smoothing length $\tilde{h}$ (which is equivalent to the corresponding SPH quantity).

Intuitively speaking, this integral accumulates force contributions of all points $\boldsymbol{y}$ in the domain in the direction of evaluation point $\boldsymbol{x}$ and weights them through evaluation of the function W. As long as W is an even function, the total force will be zero when integrating over a symmetric domain, e.g., the interior of the fluid. Only when integrating over non-symmetric domains, e.g., at the fluid surface, does the integral evaluate to a non-zero value. In addition, W can be adjusted to exhibit different attractive or repulsive properties. For example, at the molecular level W might be chosen such that the net force correspond to the force resulting from the Lennard-Jones [1931] potential. This potential is an even function yielding an attractive force when molecules are further than some molecule specific distance apart, diminishing smoothly with increasing distance. For smaller distances it results in a rapidly increasing repulsive force, proportional to the inverse distance.

In our work we choose W^{cubic} to be the cubic spline kernel [Monaghan 1992] often used in SPH simulations, with a small modification resulting in

$$W(\boldsymbol{x}-\boldsymbol{y};\tilde{h}) = ||\boldsymbol{x}-\boldsymbol{y}|| W^{\text{st}}(\boldsymbol{x}-\boldsymbol{y};\tilde{h}), \tag{2}$$

$$W^{st}(\boldsymbol{x}-\boldsymbol{y};\tilde{h}) = \theta \begin{cases} W^{\text{cubic}}(\boldsymbol{x}-\boldsymbol{y};\tilde{h}), & \text{for } ||\boldsymbol{x}-\boldsymbol{y}|| > d \\ W^{\text{cubic}}\left(\frac{\boldsymbol{x}-\boldsymbol{y}}{||\boldsymbol{x}-\boldsymbol{y}||} d;\tilde{h}\right), & \text{otherwise}. \end{cases} \tag{3}$$

Here d denotes the particle diameter, such that the value of W^{st}, and implicitly also the value of W, is clamped to a constant value as the distance between particles decreases below d. In all of our simulations we set $\tilde{h} = 2d$. The additional factor $\theta = \frac{10}{7}$ (for 3D simulations) is derived, such that the modified cubic-spline kernel function is again normalized, i.e. the integral of the kernel over the compact support is equal to one.

The multiplication by the distance $||\boldsymbol{x}-\boldsymbol{y}||$ and the clamping of the cubic-spline kernel ensures, that the attraction is maximal at one particle diameter distance d and linearly tends to zero as particles get closer. As such, the SPH discretization of Eq. (1) is given by

$$\boldsymbol{f}_{\text{st}}(\boldsymbol{x}_i) = -\sigma \sum_{j \in \mathcal{N}_i} \frac{m_j}{\rho_j} (\boldsymbol{x}_i - \boldsymbol{x}_j) W^{\text{st}}(\boldsymbol{x}_i - \boldsymbol{x}_j;\tilde{h}), \tag{4}$$

where m_j is the mass and ρ_j is the density of discrete particle j in the neighborhood $\mathcal{N}_i$ around position $\boldsymbol{x}_i$. Note that due to the dependency on the density ρ_j, the term also depends on all positions $\boldsymbol{x}_k$ in the neighborhood $k \in \mathcal{N}_j$, since

$$\rho_j = \sum_{k \in \mathcal{N}_j} m_k W^{\text{cubic}}(\boldsymbol{x}_j - \boldsymbol{x}_k;\tilde{h}). \tag{5}$$

This dependency has considerable significance for our implicit integration scheme which we introduce in the following. The formulation in Eq. (4) is very similar to the surface tension force employed by Becker and Teschner [2007], yet differs in derivation and construction of the custom clamped kernel function.

In the following, we will use the shorthands $W_{ij} = W^{\text{st}}(\boldsymbol{x}_i - \boldsymbol{x}_j;\tilde{h})$ and $\boldsymbol{x}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$. In order to symmetrize Eq. (4), we introduce $\overline{\rho}_{ij} = \frac{1}{2}(\rho_i + \rho_j)$ and $\overline{m}_{ij} = \frac{1}{2}(m_i + m_j)$ as the averaged densities

and masses. This yields the symmetric force

$$\overline{\boldsymbol{f}}_{\text{st}}(\boldsymbol{x}_i) = -\sum_{j\in\mathcal{N}_i} \underbrace{\sigma\frac{\overline{m}_{ij}}{\overline{\rho}_{ij}}\boldsymbol{x}_{ij}W_{ij}}_{\overline{\boldsymbol{f}}_{ij,\text{st}}} . \tag{6}$$

Finally, the acceleration on particle i resulting from the discretized surface tension force can be written as

$$\boldsymbol{a}_{i,\text{st}} = -\frac{1}{m_i}\sum_{j\in\mathcal{N}_i}\overline{\boldsymbol{f}}_{ij,\text{st}} = -\frac{\sigma}{m_i}\sum_{j\in\mathcal{N}_i}\frac{\overline{m}_{ij}}{\overline{\rho}_{ij}}\boldsymbol{x}_{ij}W_{ij} . \tag{7}$$

Discretizing this equation in time using the forward Euler method yields

$$\boldsymbol{a}_{i,\text{st}} = \frac{\boldsymbol{v}_i^{t+1}-\boldsymbol{v}_i^t}{\Delta t} = -\frac{\sigma}{m_i}\sum_{j\in\mathcal{N}_i}\frac{\overline{m}_{ij}}{\overline{\rho}_{ij}^t}\boldsymbol{x}_{ij}^t W_{ij}^t , \tag{8}$$

resulting in a closed expression for $\boldsymbol{v}_i^{t+1}$. Discretizing with the backward Euler method instead, this equation becomes

$$\boldsymbol{a}_{i,\text{st}} = \frac{\boldsymbol{v}_i^{t+1}-\boldsymbol{v}_i^t}{\Delta t} = -\frac{\sigma}{m_i}\sum_{j\in\mathcal{N}_i}\frac{\overline{m}_{ij}}{\overline{\rho}_{ij}^{t+1}}\boldsymbol{x}_{ij}^{t+1}W_{ij}^{t+1}, \tag{9}$$

which results in a non-linear system for the velocity $\boldsymbol{v}_i^{t+1}$, since the (non-linear) kernel function and the density both depend on $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \Delta t\boldsymbol{v}_i^{t+1}$.

3.2 Implicit Surface Tension

In the following we describe the implicit time integration in more detail. Using an implicit method improves the stability significantly when using large surface tension forces in combination with larger time steps. This enables the simulation of complex surface tension effects that explicit methods struggle to handle at typical time step sizes. The explicit version in Eq. (8) directly yields a closed solution for velocity $\boldsymbol{v}_i^{t+1}$, while the implicit version in Eq. (9) requires solving a non-linear system.

The usage of a full non-linear solver, e.g. Newton's method, is not well suited for our specific problem. The combination of Newton iterations together with a sparse linear system solver quickly becomes prohibitively expensive, both in terms of computational complexity as well as potentially memory requirements of storing the Jacobian and intermediate results for line-search. More crucially, a full Newton's method for surface tension would entail many more iterations over all particles, its neighbors and neighbors of neighbors than just a linear system solve, as well as the need to run the costly neighborhood search and to evaluate new densities in each Newton iteration. As such, we instead describe our solution method for linearized implicit surface tension forces in the following.

Naive Linearization. To be able to solve the non-linear system we choose to employ a linear approximation. The most obvious approach to this would be to simply linearize Eq. (9) in velocity

$$\begin{aligned}\boldsymbol{a}_{i,\text{st}}(\boldsymbol{x}^{t+1}(\boldsymbol{v}^{t+1})) = {}& \boldsymbol{a}_{i,\text{st}}(\boldsymbol{x}^{t+1}(\boldsymbol{v}^t)) \\ & + \nabla_{\boldsymbol{v}}\,\boldsymbol{a}_{i,\text{st}}(\boldsymbol{x}^{t+1}(\boldsymbol{v}))\big|_{\boldsymbol{v}^t}\,(\boldsymbol{v}^{t+1}-\boldsymbol{v}^t) \\ & + O(\Delta\boldsymbol{v}^2) .\end{aligned} \tag{10}$$

Solving this linear system can be viewed as solving the non-linear system using a single Newton iteration. However, in our experiments we quickly disregarded this approach as untenable. The derivative has to be computed with respect to the velocities of all particles in the immediate neighborhood, as well es their immediate neighbors due to the dependencies mentioned previously. Using an iterative and matrix-free linear system solver, this led to significantly worse performance and scaling, quickly dominating all other terms in the SPH solver. In addition, we found that the quality and stability of the simulations using this approach were, if anything, slightly inferior to the following approach.

Improved Linearization. Instead of directly linearizing Eq. (9) in velocity, we instead split the term into two parts g and h

$$\frac{\boldsymbol{v}_i^{t+1}-\boldsymbol{v}_i^t}{\Delta t} = -\frac{\sigma}{m_i}\sum_{j\in\mathcal{N}_i}\overline{m}_{ij}\overbrace{\frac{W_{ij}^{t+1}}{\overline{\rho}_{ij}^{t+1}}}^{g_{ij}^{t+1}}\overbrace{\boldsymbol{x}_{ij}^{t+1}}^{h_{ij}^{t+1}} , \tag{11}$$

and linearize them independently with respect to time t. Realizing that the term h only depends linearly on particle positions $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, it is linearized using backward Euler time integration. This results in

$$h_{ij}^{t+1} = \left(\boldsymbol{x}_i^t + \Delta t\boldsymbol{v}_i^{t+1} - \boldsymbol{x}_j^t - \Delta t\boldsymbol{v}_j^{t+1}\right) , \tag{12}$$

preventing any longer range dependencies on velocities outside of the immediate neighborhood of particle i. The other term g, which does have dependencies outside of the immediate neighborhood (see Sec. 3.1) is approximated using a first-order Taylor expansion at time t

$$g_{ij}^{t+1} \approx \tilde{g}_{ij}^{t+1} = g_{ij}^t + \Delta t\left.\frac{\partial g_{ij}}{\partial t}\right|_t = g_{ij}^t + \Delta t\left.\left(\frac{\partial g_{ij}}{\partial \boldsymbol{x}}\cdot\boldsymbol{v}\right)\right|_t . \tag{13}$$

This results in a formulation which only depends on quantities at time t. Therefore, it can be computed in advance of solving the linear system. The individual terms of g are

$$g_{ij}^t = \frac{W_{ij}^t}{\overline{\rho}_{ij}^t} , \tag{14}$$

and

$$\begin{aligned}\left.\left(\frac{\partial g_{ij}}{\partial \boldsymbol{x}}\cdot\boldsymbol{v}\right)\right|_t = {}& \frac{\nabla_i W_{ij}^t\cdot\boldsymbol{v}_i^t + \nabla_j W_{ij}^t\cdot\boldsymbol{v}_j^t}{\overline{\rho}_{ij}^t} \\ & - \frac{W_{ij}^t}{2}\left(\frac{\sum_{k\in\mathcal{N}_i} m_k(\boldsymbol{v}_i^t-\boldsymbol{v}_k^t)\cdot\nabla_i W_{ik}^t}{{\overline{\rho}_{ij}^t}^2}\right. \\ & \left. + \frac{\sum_{l\in\mathcal{N}_j} m_l(\boldsymbol{v}_j^t-\boldsymbol{v}_l^t)\cdot\nabla_j W_{jl}^t}{{\overline{\rho}_{ij}^t}^2}\right) .\end{aligned} \tag{15}$$

The first line in this equation corresponds to the time derivative of the kernel function W_{ij}, which has been expanded using the product rule, see Eq. (13), to the spatial derivative of the kernel function times the velocity. The spatial gradients ∇_i and ∇_j denote derivatives wrt. particle positions $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The second and third line correspond to the time derivative of the reciprocal averaged density $\overline{\rho}_{ij}$, which has been expanded analogously. These terms

can be recognized from the continuity equation as the negative divergence of the velocity field $\nabla \cdot \boldsymbol{v}$ times density ρ

$$-(\rho \nabla \cdot \boldsymbol{v})_i = \sum_{j \in \mathcal{N}_i} m_j (\boldsymbol{v}_i - \boldsymbol{v}_j) \cdot \nabla_i W_{ij} \,. \tag{16}$$

We precompute and store this scalar value per particle and thus eliminate the need for iterating over neighbors of neighboring particles in the linear solver. Now that all terms have been identified, we can assemble the final linearized system

$$\begin{aligned} &\frac{\boldsymbol{v}_i^{t+1} - \boldsymbol{v}_i^t}{\Delta t} = -\frac{\sigma}{m_i} \sum_{j \in \mathcal{N}_i} \overline{m}_{ij} \left(\boldsymbol{x}_{ij}^t + \Delta t \boldsymbol{v}_{ij}^{t+1} \right) \tilde{g}_{ij}^{t+1}, \\ \Leftrightarrow \quad &\boldsymbol{v}_i^{t+1} + \Delta t^2 \frac{\sigma}{m_i} \sum_{j \in \mathcal{N}_i} \overline{m}_{ij} \boldsymbol{v}_{ij}^{t+1} \tilde{g}_{ij}^{t+1} = \boldsymbol{v}_i^t - \Delta t \frac{\sigma}{m_i} \sum_{j \in \mathcal{N}_i} \overline{m}_{ij} \boldsymbol{x}_{ij}^t \tilde{g}_{ij}^{t+1}. \end{aligned} \tag{17}$$

This can also be written as

$$(\mathbf{I} + \Delta t \mathbf{A}) \boldsymbol{v}^{t+1} = \boldsymbol{v}^t + \boldsymbol{b}, \tag{18}$$

where

$$\begin{aligned} &\mathbf{A}_{ii} = \mathbf{I}_3 \frac{\sigma}{m_i} \Delta t \sum_{j \in \mathcal{N}_i} \overline{m}_{ij} \tilde{g}_{ij}^{t+1}, \quad \mathbf{A}_{ij} = -\mathbf{I}_3 \frac{\sigma}{m_i} \Delta t \overline{m}_{ij} \tilde{g}_{ij}^{t+1} \text{ for } i \neq j, \\ &\boldsymbol{b}_i = -\Delta t \frac{\sigma}{m_i} \sum_{j \in \mathcal{N}_i} \overline{m}_{ij} \boldsymbol{x}_{ij}^t \tilde{g}_{ij}^{t+1}. \end{aligned} \tag{19}$$

Here, $\mathbf{A}_{ij}$ denotes the matrix block corresponding to the entry of particles i and j, while $\mathbf{A}_{ii}$ denotes the diagonal matrix element and $\mathbf{I}_3$ the 3x3 identity matrix. The constructed linear system is symmetric positive definite if all particles have equal mass $m = m_i$, which allows us to use the iterative matrix-free Conjugate Gradient (CG) method to find the solution. Indeed, we only perform simulations with uniform particle masses. It would however be possible, to use a different iterative linear solver, such as BiCGStab, to solve the non-symmetric system for non-uniform particle masses.

The linear system could also be solved as three smaller systems, as the individual velocity components do not depend on each other. However, this can only be done when only solving implicit surface tension and is no longer possible when strong coupling with viscosity is desired, see Section 3.4.

We use a matrix-free CG solver, since it is more efficient both in terms of memory as well as computation time, as the linear system never actually has to be fully assembled. We also use the previous change in velocity to compute a prediction to warmstart the linear solver

$$\boldsymbol{v}_{\text{pred}} = \boldsymbol{v}^t + \left(\boldsymbol{v}^t - \boldsymbol{v}^{t-1} \right) . \tag{20}$$

We have found this to significantly reduce computation time by decreasing the number of required solver iterations. Yet in our experiments we have not found a diagonal preconditioner to yield any tangible improvement over using no preconditioner, although this may be a possible direction for further study.

3.3 Adhesion

We have formulated an implicit linear system allowing us to compute cohesive surface tension forces within the fluid itself. We extend this by defining an additional force which allows us to describe the surface tension properties between a fluid and a rigid object in contact. To achieve this we use an analogous formulation as for the previous cohesion model and introduce an adhesion coefficient σ^b, which defines how strongly the fluid is drawn to the rigid object. We further assume the usage of the *Volume Maps* boundary handling method [Bender et al. 2019]. Within this method, boundaries are simply defined by the volume V_i^b occupied within the compact support of particle i, the closest point on the boundary $\boldsymbol{x}_i^b$, and the velocity of the boundary $\boldsymbol{v}_i^b$. All other properties, such as rest density, are assumed to be identical to the fluid properties. The system in Eq. (19) can easily be extended to incorporate adhesion

$$\begin{aligned} \mathbf{A}_{ii} &= \mathbf{I}_3 \frac{\Delta t}{m_i} \left(\sum_{b \in \mathcal{N}_i^b} \overline{m}_i^b \sigma^b \tilde{g}_i^{b,t+1} + \sigma \sum_{j \in \mathcal{N}_i} \overline{m}_{ij} \tilde{g}_{ij}^{t+1} \right), \\ \boldsymbol{b}_i &= -\frac{\Delta t}{m_i} \left(\sum_{b \in \mathcal{N}_i^b} \overline{m}_i^b \sigma^b \Delta t \boldsymbol{v}_i^{b,t} \tilde{g}_i^{b,t+1} + \sigma \sum_{j \in \mathcal{N}_i} \overline{m}_{ij} \boldsymbol{x}_{ij}^t \tilde{g}_{ij}^{t+1} \right), \end{aligned} \tag{21}$$

where $b \in \mathcal{N}_i^b$ denotes all neighboring boundaries of particle i, and $\overline{m}_i^b = \frac{1}{2}(m_i + \rho_0 V_i^b)$ denotes the averaged mass. Here V_i^b corresponds to the volume occupied by the boundary in the compact support of particle i, as can be obtained from the volume maps approach by Bender et al. [2019]. Note that $\mathbf{A}_{ij}$ remains unchanged and that the linear system remains symmetric for uniform particle masses. The term $\tilde{g}_i^{b,t+1}$ looks slightly different,

$$\begin{aligned} \tilde{g}_i^{b,t+1} = &\frac{W_i^{b,t}}{\frac{1}{2}(\rho_i^t + \rho_0)} + \frac{\nabla_i W_i^{b,t} \cdot \boldsymbol{v}_i^t - \nabla_i W_i^{b,t} \cdot \boldsymbol{v}_i^{b,t}}{\frac{1}{2}(\rho_i^t + \rho_0)} \\ &- \frac{W_i^{b,t}}{2} \left(\frac{\sum_{k \in \mathcal{N}_i} m_k (\boldsymbol{v}_i^t - \boldsymbol{v}_k^t) \cdot \nabla_i W_{ik}^t}{\left(\frac{1}{2}(\rho_i^t + \rho_0) \right)^2} \right). \end{aligned} \tag{22}$$

Here the term corresponding to the density gradient of the boundary is dropped, as the density of the boundary volume is assumed to be constant and equal to the rest density. While we show only the derivation of the adhesion terms for Volume Maps boundary handling, the formulation can be adapted easily to other implicit boundary representations (e.g., [Koschier and Bender 2017]) or particle-based approaches (e.g., [Akinci et al. 2012]).

The effect of combining different adhesion coefficients with different cohesion coefficients is shown in Fig. 3. This figure shows that the droplet shape becomes more spherical with increasing surface tension coefficients. Increasing the adhesion coefficient makes the droplet flatter and spread out further across the plane, while also changing the contact angle between the droplet and plane.

3.4 Implicit Viscosity

Having obtained an implicit linear system containing both cohesive and adhesive surface tension effects, we propose to make one final modification. In the following, we introduce strong coupling of our surface tension force with the implicit viscosity force proposed by Weiler et al. [2018] by solving for both in a single linear system. In contrast to weak coupling where successive solves of the systems can be problematic for convergence, we expect the strong coupling

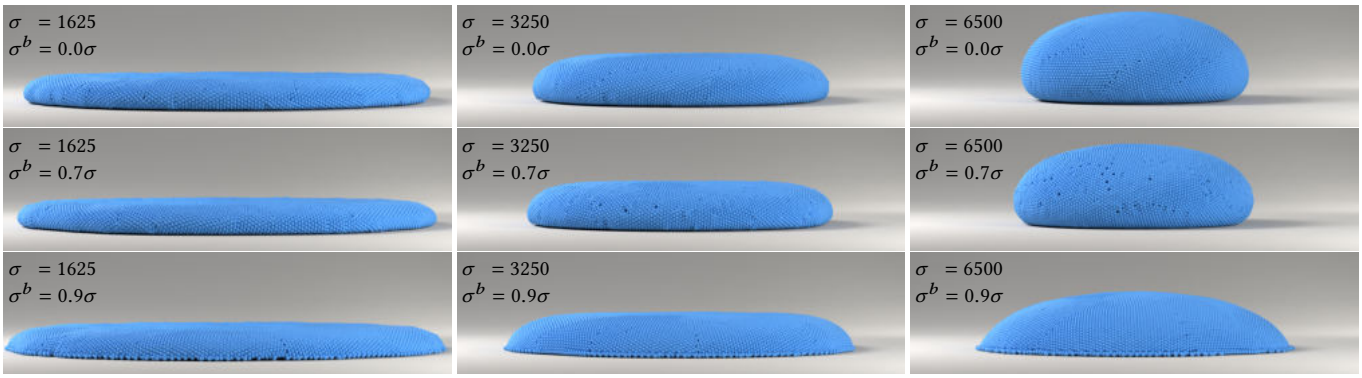

Fig. 3. Effect of droplet adhesion to a flat plane when using different cohesion and adhesion parameters. The columns from left to right show increasing surface tension coefficients $\sigma = [1625, 3250, 6500]\,\mathrm{N\,m^{-1}}$, while the rows from top to bottom show increasing adhesion coefficients $\sigma^b = [0\sigma, 0.7\sigma, 0.9\sigma]\,\mathrm{N\,m^{-1}}$ as a fraction of the surface tension coefficient.

to yield better results. In practice we observed stability benefits as well as performance benefits since fewer overall CG iterations were required (cf. Section 4).

Nevertheless, for low-viscosity fluids, the weakly coupled standard explicit viscosity model performs reasonably well. Compared to the following strong coupling approach when using surface tension however, using an explicit viscosity model is not significantly more computationally efficient. This is due to the very similar matrix structure of our implicit surface tension model and the strongly coupled viscosity model of Weiler et al. [2018].

Strongly coupling both effects can be accomplished by simply solving both linear systems simultaneously, i.e.

$$\frac{\boldsymbol{v}_i^{t+1} - \boldsymbol{v}_i^t}{\Delta t} = f_{i,\mathrm{st}}^{t+1} + f_{i,\mathrm{visc}}^{t+1}. \tag{23}$$

Because the implicit viscosity method by Weiler et al. [2018] is given in very similar notation, we extend the linear system in Eq. (18) by the contribution of viscosity

$$(\mathbf{I} + \Delta t(\mathbf{A}_{\mathrm{st}} + \mathbf{A}_{\mathrm{visc}}))\,\boldsymbol{v}^{t+1} = \boldsymbol{v}^t + \boldsymbol{b}_{\mathrm{st}}, \tag{24}$$

where the *st* subscript denotes the contributions from surface tension given by Eq. (19) and Eq. (21). The *visc* subscript denotes the contribution from viscosity forces

$$\mathbf{A}_{\mathrm{visc},ij} = 2(\delta+2)\frac{\mu\overline{m}_{ij}}{\rho_i^t\rho_j^t}\frac{\nabla W_{ij}^t \boldsymbol{x}_{ij}^{t\,T}}{||\boldsymbol{x}_{ij}^t||^2 + 0.01\tilde{h}^2},\ \mathbf{A}_{\mathrm{visc},ii} = -\sum_{j\in N_i}\mathbf{A}_{\mathrm{visc},ij}, \tag{25}$$

where μ is the dynamic viscosity coefficient, δ the number of spatial dimensions and $\tilde{h}$ the smoothing length of kernel function W. As both linear systems are symmetric positive definite individually — under the assumption of uniform particle masses — the combined system inherits this property. Therefore, we make use of the same iterative CG solver as described previously, with the same initial guess as warmstart. The full algorithm for our implicit surface tension method is shown in Algorithm 1. For all experiments we use a relative residual error tolerance $tol = 0.001$ and a maximum number of iterations $iter_{max} = 100$.

Algorithm 1 Implicit Surface Tension with Viscosity

procedure ComputeSurfaceTensionWithViscosity
  for all particles i **do**
    $(\rho\nabla\cdot\boldsymbol{v})_i^t \leftarrow$ PrecomputeDivergence ▷ Eq. (16)
    $\boldsymbol{b}_i^t \leftarrow$ ComputeCohesionAdhesionRHS ▷ Eq. (21)
    $\boldsymbol{v}_{i,\mathrm{pred}}^t \leftarrow$ ComputeWarmstart ▷ Eq. (20)
  $\boldsymbol{v}_i^{t+1} \leftarrow$ SolveCG$(\boldsymbol{b}_i^t, \boldsymbol{v}_{i,\mathrm{pred}}^t)$ ▷ Eq. (24)
  for all particles i **do**
    $\boldsymbol{a}_i \leftarrow \boldsymbol{a}_i + \frac{\boldsymbol{v}_i^{t+1} - \boldsymbol{v}_i^t}{\Delta t}$ ▷ Add acceleration

In the supplementary video we show a comparison of using varying viscosity parameters with a constant surface tension coefficient. Larger viscosity values result in a smooth surface yet lead to damped fluid motion. Lower viscosity values are less dissipative, yet result in more chaotic motion at the fluid surface. This effect can generally be observed for inviscid fluids using SPH and is already well documented by Monaghan [1989, 1992], where a proposed solution is to apply velocity smoothing in order to add coupling between velocities of neighboring particles.

3.5 Pressure Coupling

Due to the nature of surface tension forces acting in normal direction to the fluid interface, they are directly opposed to pressure forces. Surface tension forces lead to a minimization of the surface area of the fluid, while pressure forces counteract compression and, to a certain extent, expansion of the fluid. While there have been previous works investigating strong coupling of pressure and non-pressure forces [Hopp-Hirschler and Nieken 2019; Liu et al. 2022], we have not found the weak coupling of our implicit surface tension and the implicit pressure solver to cause any issues. To verify this, we extended the "DFSPH-IV" approach of Liu et al. [2022] to strongly couple the implicit pressure solver by interleaving it with our proposed implicit surface tension solver. The outline of the resulting algorithm is shown in Algorithm 2. Weakly coupled operator splitting can be recovered when setting $iter_{max} = 1$. Results comparing

Algorithm 2 Simulation Step

procedure SimStep
  for all particles i **do**
    $\mathcal{N}_i \leftarrow$ FindNeighboringParticles(i)
  for all particles i **do**
    $\rho_i \leftarrow$ ComputeDensity(i)
    $\boldsymbol{v}_i^* \leftarrow \boldsymbol{v}_i + \Delta t \boldsymbol{g}$ ▷ Add gravity accel.
    $\boldsymbol{v}_i^* \leftarrow \boldsymbol{v}_i^* + \Delta t \boldsymbol{a}_i^{np}$ ▷ Add explicit non-pressure forces
  while $iter < iter_{max}$ **do**
    $\boldsymbol{v}^* \leftarrow$ DFSPHDivergenceFree($\boldsymbol{v}^*$)
    $\boldsymbol{a}^{st} \leftarrow$ Algorithm 1 ▷ Implicit surface tension
    $\boldsymbol{v}^* \leftarrow \boldsymbol{v}^* + \Delta t \boldsymbol{a}^{st}$
    $\boldsymbol{v}^* \leftarrow$ DFSPHConstantDensity($\boldsymbol{v}^*$)
  for all particles i **do**
    $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i^*$ ▷ Update velocity
    $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \Delta t \boldsymbol{v}_i$ ▷ Update positions

explicit surface tension, weakly pressure coupled implicit surface tension and iteration-based strong pressure-coupled implicit surface tension are presented in Section 4.1.

Only for very large time steps, combined with very large surface tension coefficients, did the weakly coupled iterative solvers not converge. These configurations incurred a lot of compression from the surface tension forces. The subsequent application of the pressure solver to enforce constant density would then result in very large pressure forces and could then lead to instabilities. This was however typically easily solvable, by slightly reducing time step size, particle radius, or both. In addition, weakly coupled pressure and non-pressure solvers remain the current state of the art for SPH simulation methods [Ihmsen et al. 2014b; Koschier et al. 2022]. We leave further and more thorough investigations of strong pressure coupling as a possible direction of future work, as the selection of time step size was not a constraining factor in any of our simulations.

4 RESULTS

In this section we aim to show the efficacy of our implicit surface tension method using a wide variety of examples. To begin with, we investigate comparisons with our method. This includes a comparison of the different variations of our proposed method, namely explicit integration, implicit integration with weak pressure coupling and finally implicit integration with iteration-based strong pressure coupling. Subsequently, we directly compare our implicit method with weak pressure coupling to other surface tension methods for SPH in order to show that we are able to match, if not even outperform existing approaches.

Afterwards, we show in more detail the capabilities of our method, especially with respect to the large surface tension coefficients enabled by our implicit method. We were not able to reproduce the majority of these examples with the other approaches due to stability issues as a consequence of their explicit integration.

Unless otherwise specified, we use the *Divergence-Free SPH* pressure solver [Bender and Koschier 2017] and the implicit viscosity method of Weiler et al. [2018] in addition to the respective surface tension model. For all methods we use the implementations available in the open-source SPH framework SPlisHSPlasH [Bender et al. 2022].

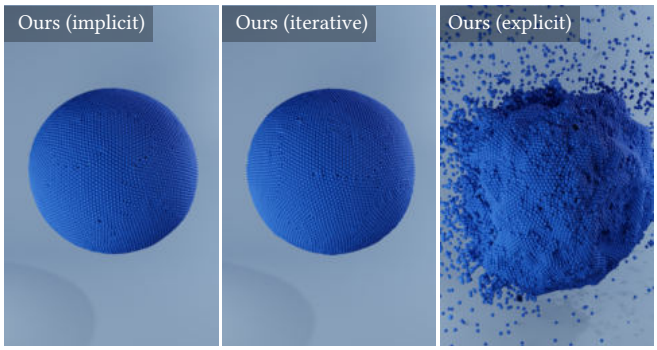

Fig. 4. Comparison of a droplet forming in zero-gravity. Our implicit method (left) remains stable and forms a smooth sphere while the explicit version of our method (right) becomes unstable with increasing surface tension parameter. The iterative method for strong coupling with pressure forces (middle) also remains stable.

4.1 Sphere Stability

Visual Comparison. To start with, and to further show the stability of the implicit formulation, we compare the implicit version of our surface tension method against the explicit version in a zero-gravity setting and without viscosity. For implicit surface tension, we present two versions: one utilizing weak pressure coupling, being commonly used in recent SPH publications, and another employing an iterative strong coupling scheme proposed by Liu et al. [2022] with $iter_{max} = 5$ iterations (see Algorithm 2). The lack of viscosity ensures that surface tension and pressure are the only forces acting on the fluid. Furthermore, we use XSPH velocity smoothing as described by Monaghan [1989] with a factor of 0.5. The visual comparison is shown in Fig. 4.

The fluid starts out as a block and the surface tension coefficient is initialized to a small value. After 2 s, this value is increased continuously — such that the block should transform into a sphere — until the explicit method becomes unstable (shown in the figure). It is clearly observable that the explicit methods forms irregular clumps at the surface of the fluid, while the implicit methods forms a smooth and stable sphere. This disparity can be attributed to the explicit method getting "stuck" in locally optimal configurations, while the solution of a linear system in the implicit methods is able to account for the global shape of the fluid. Additionally, the explicit method seems to "gain" volume, which is another artifact where the outermost layer of particles detaches from the bulk of the fluid, to form a kind of "hull".

Although both implicit methods form a stable sphere, the surface of the iteration-based strong pressure coupling appears somewhat "rougher". We attribute this to the fact, that we had to reduce the surface tension coefficient to 0.77x the coefficient of the weakly coupled simulation, to obtain comparable results. This change in apparent magnitude of effects was also documented in the work of Liu et al. [2022].

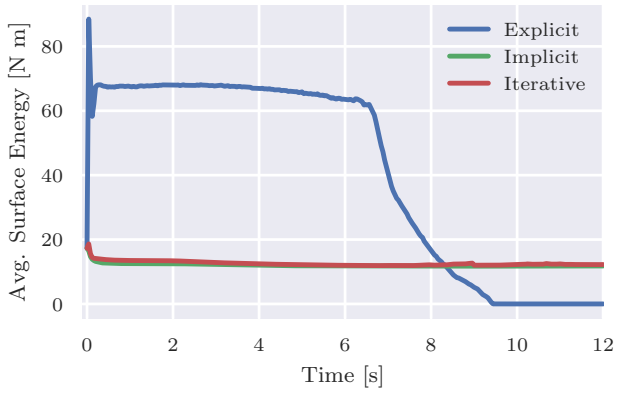

Fig. 5. Surface energy over time for the simulations in Fig. 4.

Table 1. Comparison of performance metrics for the sphere stability scene. ST abbreviates "surface tension", while DFSPH iterations denotes the number of pressure solver iterations and time coupling the total time needed to solve both surface tension and pressure.

	Explicit	Implicit	Iterative
Avg. ST Time [ms]	3.95	9.34	31.92
Avg. Time Coupling [ms]	12.61	44.74	154.51
$iter_{max}$ [-]	1	1	5
Avg. DFSPH Iterations [-]	4.13	49.89	209.75
$\sigma_{t^{start}}$ [N m^{-1}]	70000		54000
$\sigma_{t^{end}}$ [N m^{-1}]	140000		108000

Surface Energy. The evolution over time of the average free surface energy (see He et al. [2014] and Cahn and Hilliard [1958]), of the three variations is shown in Fig. 5. For the explicit solver, the surface energy oscillates in the beginning after settling into a kind of equilibrium. It slowly starts decreasing until dropping rapidly to zero as the simulation becomes unstable and particles scatter.

In contrast, both implicit methods drop quickly to a stable energy level as a sphere is formed. The energy continues to decrease slightly with increasing surface tension, as some particles in the sphere rearrange into a more optimal configuration.

Timing. To conclude the comparison, we briefly outline some of the more descriptive performance metrics in Table 1. It is clear that the explicit method is computationally the cheapest, followed by the implicit method with weak pressure coupling and finally the implicit method with strong pressure coupling. It is notable that 5 strong coupling iterations do not result in 5 times longer runtime but "only" incurs a factor of roughly 3.4. Also, due to the very large surface tension coefficient, a lot of pressure iterations and large pressure values are required to counteract the compression.

Summary. This example shows that the proposed iteration-based strong coupling between surface tension and pressure forces is possible, but not needed to maintain stable interactions between our implicit surface tension and pressure solvers, even in the absence of viscosity. Overall, we did not observe any tangible benefits when using this particular coupling approach. In specific, it did not allow larger time steps, did not result in more dynamic movement and did not improve stability over weak pressure coupling. As such, we will use our implicit solver with weak pressure coupling for all following experiments, unless otherwise stated.

Table 2. Time step sizes and average timings per simulation step of the surface tension models in two of the comparison cases. All values are given in ms and the runtimes *include the implicit viscosity solver*. The abbreviations *sc* and *wc* denote strong coupling and weak coupling with implicit viscosity respectively, while *BT* denotes the method by Becker and Teschner [2007].

	Double Droplet		Droplet Crown	
	Δt	Avg. Runtime	Δt	Avg. Runtime
Ours (sc)	2.0	1.90	2.0	22.00
Ours (wc)	2.0	3.06	2.0	36.57
Akinci et al.	1.0	2.91	2.0	32.60
BT	0.5	1.51	2.0	21.69
He et al.	0.5	1.92	2.0	32.25
Zorilla et al.	0.8	24.21	2.0	436.10

Further investigation of more advanced methods of coupling pressure and non-pressure forces might yield different results however, possibly making simulations of even stronger surface tension effects possible at even larger time steps.

4.2 Comparisons

For the following comparisons the parameters for the pressure and viscosity solver are kept consistent across the different surface tension models. The parameters of the other surface tension models were adjusted to the best of our abilities, to yield the best possible result in the given situation. Note that here, strong and weak coupling refers to coupling between surface tension and viscosity.

Double Droplet and Impact. Next, (see Fig. 6) two fluid blocks are released and kept in zero-gravity until they have (ideally) contracted into a single stable fluid mass (see Fig. 6a). In this comparison, only our method (using both strong and weak coupling of viscosity and surface tension) as well as the method of Zorilla et al. [2020] are able to obtain an actual fluid sphere from the initial double block configuration. While the method of Akinci et al. [2013] forms an ellipsoid, the method of Becker and Teschner [2007] forms almost two distinct spheres and using the method of He et al. [2014] we have unfortunately not been able to obtain a stable configuration as a single fluid mass. The reader is encouraged to refer to the supplementary video to confirm that our method also produces the most dynamic behavior during droplet formation.

Once a stable fluid mass has formed, ideally a droplet, gravity is reactivated and the droplet falls onto a flat plane (without adhesion) and should form a stable flattened droplet as well (see Fig. 6b). We can observe all methods being able to form some kind of droplet, with the exception of the one of He et al. [2014]. Yet again, we can see that both the shape of the droplets as well as the velocity distributions on the surface are the most stable for our method. It is also possible to see that the (discouraged) weak coupling of viscosity with our method introduces some small velocity oscillations in the regions of largest curvature, while the result with our proposed

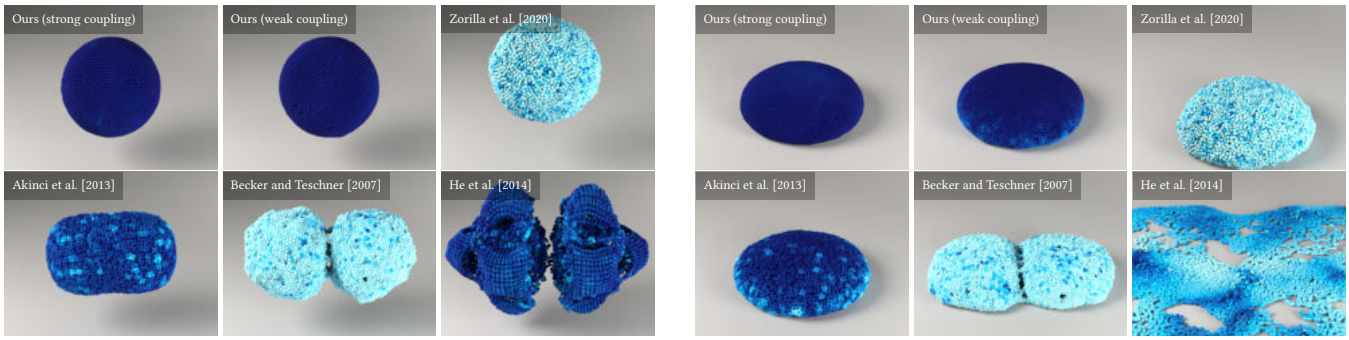

Fig. 6. Comparison of a variety of state-of-the-art SPH surface tension methods. (a) Two fluid blocks are released in zero gravity and form a droplet. (b) After reaching a stable configuration, gravity is enabled and the droplets fall onto a flat plane without adhesion. The shading of the particles shows the magnitude of the velocity, with a lighter color indicating a larger velocity value.

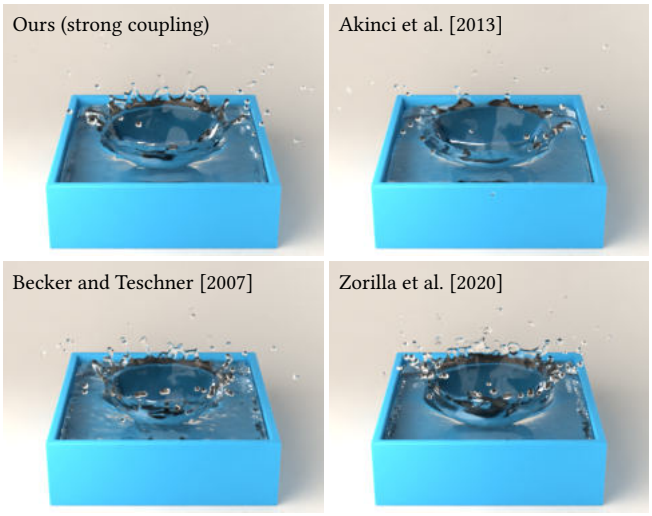

Fig. 7. The formation of a fluid crown when a ball of water is dropped into a container.

strong coupling does not. The parameters of each method were adjusted such that the simulation is stable and produces droplets of similar shape on the flat plane at the end of the simulation. For a more dynamic view of this comparison the reader is referred to the supplemental video.

Droplet Crown. For the next comparison we show another common example for surface tension simulations, namely the droplet crown. A ball of water is dropped into a still pool, which causes a splash to form in the shape of a crown. The results of this experiment are shown in Fig. 7. For the comparison the method of He et al. [2014] and our method with weak viscosity coupling are disregarded. It can be seen that all models are able to produce a fluid crown. The model of Becker and Teschner [2007] and Zorilla et al. [2020] produce more splashing individual droplets, while our method produces a slightly more cohesive crown with intricate details and fewer individual splashes. The crown for Akinci et al. [2013] is more subdued and spread out wider but still visible. Note however, that there is no absolute ground truth to what this droplet crown should look like. As such any one of the methods cannot be said to be better or more "realistic" than the others.

Table 2 shows a performance comparison of the various methods. It is evident that our method performs very competitively in terms of computational efficiency. Strong coupling especially shows significant performance improvements, which we attribute to the fact that the surface tension and the viscosity matrix have the same structure. It should further be noted, that all methods have been parallelized using OpenMP. However, our method was further optimized using SIMD vectorization while the explicit methods were not. For the double droplet comparison, we were able to use larger time steps than all other explicit methods, while we used the same time step for all methods in the droplet crown example. For the droplet crown example the surface tension coefficient was chosen to be an order of magnitude smaller than for the double droplet comparison, which is why the other explicit surface tension methods also remained stable. This comparison shows that our method is already able to perform very competitively in cases with low to moderate surface tension forces, while it is able to outperform existing approaches for larger surface tension forces.

Sphere Adhesion. For a final comparison we set up a scene in which fluid is emitted onto a sphere, coats the sphere due to adhesion and drips down at the bottom without separating from the object. We compare our method only with the method of Akinci et al. [2013] because the methods of Zorilla et al. [2020] and Becker and Teschner [2007] do not implement adhesion. The comparison is shown in Fig. 8. We can see that our method is able to cover the sphere almost perfectly uniformly, dripping down in a thin and contained stream. For the method of Akinci et al. [2013], however, we struggled to find a good balance of adhesion and cohesion forces to achieve the same effect, resulting in the clustering of particles around the bottom of the sphere and a less uniform stream dripping down. In addition, the particle coloring, indicating the velocity

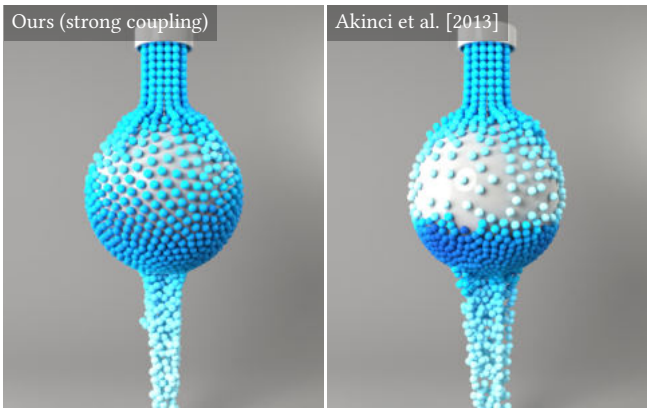

Fig. 8. A stream of fluid is emitted over the top of a sphere, adheres to and covers the object and drips down in a uniform stream.

Fig. 9. A *water bell* forms when the streams of two equal and opposite vertical emitters meet. The particle view is cut open by a clipping plane to visualize the interior. Parameters: Δt = 0.001 s, particle radius = 0.015 m, density = 1000 kg m^{-3}, ν = 0.2 m^2 s^{-1}, σ = 7560 N m^{-1}.

magnitude, shows that our method produces smooth velocity transitions, while the method of Akinci et al. [2013] produces larger fluctuations.

4.3 Further Examples

In the previous subsection we compared our method to other SPH surface tension methods and were able to confirm the effectiveness and stability of our method on common surface tension benchmarks. In the following we will instead focus on testing the limits of our method by showing its full capabilities and versatility in a number of challenging scenarios.

Water-Bell. First, we investigate two interesting effects which can be achieved using just two emitters. The first, called the *water bell*, is achieved by arranging two emitters vertically facing each other (see Fig. 1 and Fig. 9). In our example, the fluid streams meet and spread out horizontally until gravity causes them to drop down vertically. Due to the surface tension forces acting on the fluid, the vertically falling ring of fluid contracts before merging into a single stream. This results in the formation of a stable bell-like shape, which can also be observed in reality. In our experiments, we have not been able to reproduce this result with any of the other explicit SPH surface tension methods.

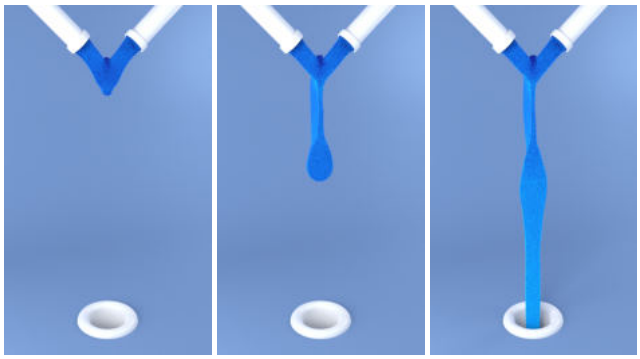

Fig. 10. Two fluid streams causing the formation of a *fluid chain*. The particle view is cut open by a clipping plane to visualize the interior. Parameters: Δt = 0.001 s, particle radius = 0.01 m, density = 1000 kg m^{-3}, ν = 0.0125 m^2 s^{-1}, σ = 8400 N m^{-1}.

Fluid Chain. The second effect, termed *fluid chain*, is achieved by arranging two fluid streams meeting at an angle from above (see Fig. 10). As the streams meet and first expand into the perpendicular direction, similar to the *water bell* in Fig. 9, a first link is formed. The second link is formed due to surface tension forces contracting the first chain link and expanding it into a plane perpendicular to the first chain link. Finally, the surface tension force balances out and contracts the fluid into a single stable stream. The size and number of chain links can be varied by changing the angle and velocity of the inflow. As with the water-bell, we have not been able to qualitatively reproduce this result with any of the other explicit SPH surface tension methods. While the method of Akinci et al. [2013] was able to reproduce a cohesive fluid "string", the chain links that formed were barely discernable at best.

Droplet Toy. In Fig. 1 we show the simulation of a child's toy containing droplets. A reservoir at the top of the toy causes droplets to drip down through funnels, entering a maze of obstacles. The droplets fall through the maze and deform dynamically, bouncing against obstacles and merging with other droplets to form larger droplets. This simulation shows the ability of our method to readily simulate cases with large surface tension coefficients (the droplets maintain an almost spherical shape at any size) and interactions with boundary objects. The key parameters for this simulation are: Δt = 0.001 s, particle radius = 0.025 m, density = 1000 kg m^{-3}, ν = 0.002 m^2 s^{-1}, $\sigma = 5 \times 10^5$ N m^{-1}, $\sigma^b = 1.25 \times 10^5$ N m^{-1}.

Dripping Faucet. Another simulation presented in Fig. 1 is the simulation of water dripping from a leaky faucet into a glass. The faucet has a diameter of 2 cm, while the glass is 14 cm tall. The simulation consists of 250k particles at the end of the simulation, with a diameter of 0.1 mm. Due to the slow emission of fluid, water gathers at the top of the faucet until the surface tension cannot counteract the gravitational force anymore. At this point, the water drips down in a small stream, which breaks up into smaller droplets due to surface tension. It should be noted that the faucet does not contain a so-called "aerator" at the tip, which is why this dripping behavior is more reminiscent of a garden hose water tap, instead

of a kitchen sink. The key parameters for this simulation are: $\Delta t = 7 \times 10^{-5}$ s, particle radius = 5×10^{-5} m, density = 1000 kg m^{-3}, $\nu = 5 \times 10^{-6}$ m^2 s^{-1}, $\sigma = 0.02$ N m^{-1}, $\sigma^b = 0.01$ N m^{-1}.

Thin-Films. Using our implicit method and a careful balance of cohesion, adhesion and viscosity forces, we show the simulation of a catenoid soap film connecting two tori in Fig. 11 on the left. The tori are first moved closer together and then apart, causing first an expansion and then a contraction of the catenoid.

Our model is also capable of simulating "soap" bubbles. The bubbles are made up of a thin soap phase, wrapped around a second much lighter "gas"-phase of particles, to emulate the internal air pressure of the bubble. For the coupling of the two phases we make use of the density-contrast multi-phase model from Solenthaler and Pajarola [2008]. A similar two-phase approach has been used by Yang et al. [2017] for the simulation of blowing bubbles, yet using a volume-fraction-based multiphase model instead of two distinct phases. We show the simulation of two and 27 colliding bubbles in the middle and right of Fig. 11 respectively.

While our model is able to keep the thin surfaces stable and intact, we have observed that the results generally lack surface detail (capillary-waves) and appear somewhat "smoothed". We attribute this to the problem of using a 3D SPH kernel for simulating a 2D phenomenon. In order to more accurately handle these cases, we believe combining our method with a co-dimensional model [Wang et al. 2020] to be a promising direction of further research. Nevertheless, we believe our surface tension method being able to stably simulate thin-film behavior to be an interesting and worthwhile result. This type of thin-film simulation is unthinkable with aforementioned existing explicit SPH surface tension approaches.

5 CONCLUSION

Previous surface tension approaches for SPH have been mostly explicit [Akinci et al. 2013; Becker and Teschner 2007; He et al. 2014; Zorilla et al. 2020], requiring small time steps to be able to simulate complex phenomena including large coefficients and intricate force interactions. Furthermore, explicit curvature based methods often suffered from inaccurate curvature computation from the color field, while explicit cohesion based methods have been known to suffer from increased particle clustering at the surface (see Sec. 2).

In this paper we have presented our model for computing surface tension forces implicitly in SPH simulations. We have derived a non-linear surface tension force from inter-particle cohesion forces, have proposed a suitable linearization and solution method, and have described how to incorporate adhesion effects. In addition, we have shown how to strongly couple our approach with a state-of-the-art implicit viscosity solver [Weiler et al. 2018].

We have demonstrated the versatility of our approach on a number of well-known benchmarks, including droplet formation, droplet impact on a flat plane, fluid crown and sphere adhesion. Going beyond these simple benchmarks, we have presented a variety of more challenging scenarios which typically required a specific balance between cohesion, adhesion and viscosity forces to function as expected.

Nevertheless, we have observed our model to also have some limitations. While we are able to simulate strong surface tension effects, we sometimes needed to use the momentum-conserving and non-dissipative XSPH velocity smoothing to avoid overly chaotic motion at the surface. This however is well established for inviscid SPH simulations, which naturally decouple velocities of neighboring particles [Monaghan 1989]. Furthermore, while we were even able to simulate thin films, these simulations were generally less predictable and harder to tune. We attribute this to the fact that we attempted to simulate a 2D phenomenon using a 3D discretization. Especially in SPH, thin films cause significant particle deficiency with large parts of the compact support being empty. We believe that combining our method with existing codimensional [Wang et al. 2020] or thin-film simulation [Wang et al. 2021] approaches could be interesting for future work. Finally, we observed some of our simulation results to appear overly smoothed, most likely due to the dissipative nature of implicit integration schemes. Possible approaches to reintroduce details would be the simulation of capillary waves [Yang et al. 2016a], or the investigation of higher-order and less dissipative integration schemes [Löschner et al. 2020].

ACKNOWLEDGMENTS

The presented investigations were carried out at RWTH Aachen University within the framework of the Collaborative Research Centre SFB1120-236616214 "Bauteilpräzision durch Beherrschung von Schmelze und Erstarrung in Produktionsprozessen" and funded by the Deutsche Forschungsgemeinschaft e.V. (DFG, German Research Foundation). The sponsorship and support is gratefully acknowledged.

REFERENCES

Mridul Aanjaneya, Saket Patkar, and Ronald Fedkiw. 2013. A monolithic mass tracking formulation for bubbles in incompressible flow. *J. Comput. Phys.* 247 (aug 2013), 17–61. https://doi.org/10.1016/j.jcp.2013.03.048

Nadir Akinci, Gizem Akinci, and Matthias Teschner. 2013. Versatile surface tension and adhesion for SPH fluids. *ACM Transactions on Graphics* 32, 6 (2013), 1–8. 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 Transactions on Graphics* 31, 4 (July 2012), 1–8. https://doi.org/10.1145/2185520.2335413

A. Albadawi, D.B. Donoghue, A.J. Robinson, D.B. Murray, and Y.M.C. Delauré. 2013. Influence of surface tension implementation in Volume of Fluid and coupled Volume of Fluid with Level Set methods for bubble growth and detachment. *International Journal of Multiphase Flow* 53 (jul 2013), 11–28. https://doi.org/10.1016/j.ijmultiphaseflow.2013.01.005

Omri Azencot, Orestis Vantzos, Max Wardetzky, Martin Rumpf, and Mirela Ben-Chen. 2015. Functional Thin Films on Surfaces. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '15)*. ACM, New York, NY, USA, 137–146. https://doi.org/10.1145/2786784.2786793

Christopher Batty, Andres Uribe, Basile Audoly, and Eitan Grinspun. 2012. Discrete Viscous Sheets. *ACM Transactions on Graphics* 31, 4 (2012), 1–7. https://doi.org/10.1145/2185520.2185609

Markus Becker and Matthias Teschner. 2007. Weakly compressible SPH for free surface flows. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–8. http://portal.acm.org/citation.cfm?id=1272690.1272719

Jan Bender et al. 2022. SPlisHSPlasH Library. https://github.com/InteractiveComputerGraphics/SPlisHSPlasH.

Jan Bender and Dan Koschier. 2017. Divergence-Free SPH for Incompressible and Viscous Fluids. *IEEE Transactions on Visualization and Computer Graphics* 23, 3 (2017), 1193–1206. https://doi.org/10.1109/TVCG.2016.2578335

Jan Bender, Tassilo Kugelstadt, Marcel Weiler, and Dan Koschier. 2019. Volume Maps: An Implicit Boundary Representation for SPH. In *Motion, Interaction and Games*. ACM, 26.

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. *ACM Transactions on Graphics* 29, 4 (2010), 1. https://doi.org/10.1145/1833351.1778853

Fig. 11. Simulation of a thin-film catenoid connecting two tori (left), two colliding bubbles (middle) and 27 colliding bubbles (left) using our method. The catenoid simulation in particular demonstrates the ability of our method to robustly handle complex interactions of surface tension, adhesion and viscosity. The inside of the bubbles are filled with air particles, while the film is made up of different "soap" particles. Parameters: Δt = 0.002 s, particle radius = 0.004 m, density = 1000 kg m^{-3}, ν = 0.02 m^2 s^{-1}, σ = 107.52 N m^{-1}.

Landon Boyd and Robert Bridson. 2012. MultiFLIP for energetic two-phase fluid simulation. *ACM Transactions on Graphics* 31, 2 (2012), 16.

J.U Brackbill, D.B Kothe, and C Zemach. 1992. A continuum method for modeling surface tension. *J. Comput. Phys.* 100, 2 (jun 1992), 335–354. https://doi.org/10.1016/0021-9991(92)90240-y

John W. Cahn and John E. Hilliard. 1958. Free Energy of a Nonuniform System. I. Interfacial Free Energy. *The Journal of Chemical Physics* 28, 2 (feb 1958), 258–267. https://doi.org/10.1063/1.1744102

Jingyu Chen, Victoria Kala, Alan Marquez-Razon, Elias Gueidon, David A. B. Hyde, and Joseph Teran. 2021. A momentum-conserving implicit material point method for surface tension with contact angles and spatial gradients. *ACM Transactions on Graphics* 40, 4 (aug 2021), 1–16. https://doi.org/10.1145/3450626.3459874

Yi-Lu Chen, Jonathan Meier, Barbara Solenthaler, and Vinicius C. Azevedo. 2020. An extended cut-cell method for sub-grid liquids tracking with surface tension. *ACM Transactions on Graphics* 39, 6 (dec 2020), 1–13. https://doi.org/10.1145/3414685.3417859

Simon Clavet, Philippe Beaudoin, and Pierre Poulin. 2005. Particle-based Viscoelastic Fluid Simulation. In *Symposium on Computer Animation*, D. Terzopoulos, V. Zordan, K. Anjyo, and P. Faloutsos (Eds.). The Eurographics Association. https://doi.org/10.2312/SCA/SCA05/219-228

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 Transactions on Graphics* 34, 4, Article 149 (July 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 *Eurographics*. Springer Vienna, 61–76. https://doi.org/10.1007/978-3-7091-7486-9_5

Amirsaman Farrokhpanah, Javad Mostaghimi, and Markus Bussmann. 2021. Nonlinear Enthalpy Transformation for Transient Convective Phase Change in Smoothed Particle Hydrodynamics (SPH). *Numerical Heat Transfer, Part B: Fundamentals* 79, 5-6 (June 2021), 255–277. https://doi.org/10.1080/10407790.2021.1929295

R. a. Gingold and J.J. Monaghan. 1977. Smoothed Particle Hydrodynamics: Theory and Application to Non-Spherical Stars. *Monthly Notices of the Royal Astronomical Society* 181 (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 Transactions on Graphics* 39, 4 (Aug. 2020), 1–16.

Christoph Gissler, Andreas Peer, Stefan Band, Jan Bender, and Matthias Teschner. 2019. Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling. *ACM Transactions on Graphics* 38, 1 (2019).

Xiaowei He, Huamin Wang, Fengjun Zhang, Hongan Wang, Guoping Wang, and Kun Zhou. 2014. Robust Simulation of Sparsely Sampled Thin Features in SPH-Based Free Surface Flows. *ACM Transactions on Graphics* 34, 1 (2014), 7:1–7:9. https://doi.org/10.1145/2682630

Jeong-Mo Hong and Chang-Hun Kim. 2003. Animation of Bubbles in Liquid. *Computer Graphics Forum* 22, 3 (sep 2003), 253–262. https://doi.org/10.1111/1467-8659.00672

Jeong-Mo Hong and Chang-Hun Kim. 2005. Discontinuous fluids. *ACM Transactions on Graphics* 24, 3 (2005), 915. https://doi.org/10.1145/1073204.1073283

Jeong-Mo Hong, Ho-Young Lee, Jong-Chul Yoon, and Chang-Hun Kim. 2008. Bubbles alive. *ACM Transactions on Graphics* 27, 3 (Aug. 2008), 1. https://doi.org/10.1145/1360612.1360647

Manuel Hopp-Hirschler and Ulrich Nieken. 2019. Fully implicit time integration in truly incompressible SPH. *The European Physical Journal Special Topics* 227, 14 (mar 2019), 1501–1514. https://doi.org/10.1140/epjst/e2019-800152-6

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

Markus Huber, Stefan Reinhardt, Daniel Weiskopf, and Bernhard Eberhardt. 2015. Evaluation of Surface Tension Models for SPH-Based Fluid Animations Using a Benchmark Test. https://doi.org/10.2312/VRIPHYS.20151333

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 Transactions on Graphics* 39, 4 (2020). 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. *Eurographics (State of the Art Reports)* (2014), 21–42. https://doi.org/10.2312/egst.20141034

Sadashige Ishida, Peter Synak, Fumiya Narita, Toshiya Hachisuka, and Chris Wojtan. 2020. A model for soap film dynamics with evolving thickness. *ACM Transactions on Graphics* 39, 4 (aug 2020). https://doi.org/10.1145/3386569.3392405

Sadashige Ishida, Masafumi Yamamoto, Ryoichi Ando, and Toshiya Hachisuka. 2017. A Hyperbolic Geometric Flow for Evolving Films and Foams. *ACM Transactions on Graphics* 36, 6, Article 199 (Nov. 2017), 11 pages. https://doi.org/10.1145/3130800.3130835

Stefan Rhys Jeske, Marek Sebastian Simon, Oleksii Semenov, Jan Kruska, Oleg Mokrov, Rahul Sharma, Uwe Reisgen, and Jan Bender. 2022. Quantitative evaluation of SPH in TIG spot welding. *Computational Particle Mechanics* (apr 2022). https://doi.org/10.1007/s40571-022-00465-x

Myungjoo Kang, Barry Merriman, and Stanley Osher. 2008. Numerical simulations for the motion of soap bubbles using level set methods. *Computers & Fluids* 37, 5 (jun 2008), 524–535. https://doi.org/10.1016/j.compfluid.2007.07.002

Hisaya Komen, Manabu Tanaka, and Hidenori Terasaki. 2020. Three-dimensional Simulation of Gas Metal Arc Welding Process Using Particle-grid Hybrid Method. *QUARTERLY JOURNAL OF THE JAPAN WELDING SOCIETY* 38, 2 (2020), 25s–29s. https://doi.org/10.2207/qjjws.38.25s

Dan Koschier and Jan Bender. 2017. Density Maps for Improved SPH Boundary Handling. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–10.

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

Tassilo Kugelstadt, Jan Bender, José Antonio Fernández-Fernández, Stefan Rhys Jeske, Fabian Löschner, and Andreas Longva. 2021. Fast Corotated Elastic SPH Solids with Implicit Zero-Energy Mode Control. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 4, 3 (sep 2021), 1–21. https://doi.org/10.1145/3480142

L. D. Landau and E. M. Lifshitz. 2013. *Fluid Mechanics*. Elsevier Science & Techn.

J E Lennard-Jones. 1931. Cohesion. *Proceedings of the Physical Society* 43, 5 (sep 1931), 461–482. https://doi.org/10.1088/0959-5309/43/5/301

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 (sep 2022), 3168–3179. https://doi.org/10.1109/tvcg.2021.3055789

Fabian Löschner, Andreas Longva, Stefan Jeske, Tassilo Kugelstadt, and Jan Bender. 2020. Higher-Order Time Integration for Deformable Solids. *Computer Graphics Forum* 39, 8 (2020).

L. B. Lucy. 1977. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal* 82 (12 1977), 1013–1024. https://doi.org/10.1086/112164

Marek Krzysztof Misztal, Kenny Erleben, Adam Bargteil, Jens Fursund, Brian Bunch Christensen, Jakob Andreas Baerentzen, and Robert Bridson. 2014. Multiphase Flow of Immiscible Fluids on Unstructured Moving Meshes. *IEEE Transactions on Visualization and Computer Graphics* 20, 1 (jan 2014), 4–16. https://doi.org/10.1109/tvcg.2013.97

Marek Krzysztof Misztal, Kenny Erleben, Adam Bargteil, Jens Fursund, Brian Bunch Christensen, J. Andreas Bærentzen, and Robert Bridson. 2012. Multiphase Flow of Immiscible Fluids on Unstructured Moving Meshes. https://doi.org/10.2312/SCA/SCA12/097-106

J.J. Monaghan. 1989. On the problem of penetration in particle methods. *J. Comput. Phys.* 82, 1 (may 1989), 1–15. https://doi.org/10.1016/0021-9991(89)90032-6

JJ Monaghan. 1992. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics* 30, 1 (1992), 543–574. https://doi.org/10.1146/annurev.astro.30.1.543 arXiv:arXiv:1007.1245v2

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

Matthias Müller, David Charypar, and Markus Gross. 2003. Particle-Based Fluid Simulation for Interactive Applications. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 154–159. https://doi.org/citation.cfm?id=846298

Michael Nelkon. 1969. *Mechanics and properties of matter*. Heinemann Educational. 235 pages.

Xingyu Ni, Bo Zhu, Bin Wang, and Baoquan Chen. 2020. A level-set method for magnetic substance simulation. *ACM Transactions on Graphics* 39, 4 (aug 2020). https://doi.org/10.1145/3386569.3392445

Saket Patkar, Mridul Aanjaneya, Dmitriy Karpman, and Ronald Fedkiw. 2013. A hybrid Lagrangian-Eulerian formulation for bubble generation and dynamics. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM Press. https://doi.org/10.1145/2485895.2485912

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/10.1111/cgf.13317

Stéphane Popinet. 2018. Numerical Models of Surface Tension. *Annual Review of Fluid Mechanics* 50, 1 (jan 2018), 49–75. https://doi.org/10.1146/annurev-fluid-122316-045034

Liangwang Ruan, Jinyuan Liu, Bo Zhu, Shinjiro Sueda, Bin Wang, and Baoquan Chen. 2021. Solid-fluid interaction with surface-tension-dominant contact. *ACM Transactions on Graphics* 40, 4 (aug 2021), 1–12. https://doi.org/10.1145/3450626.3459862

Craig Schroeder, Wen Zheng, and Ronald Fedkiw. 2012. Semi-implicit surface tension formulation with a Lagrangian surface mesh on an Eulerian simulation grid. *J. Comput. Phys.* 231, 4 (feb 2012), 2092–2115. https://doi.org/10.1016/j.jcp.2011.11.021

Barbara Solenthaler and Renato Pajarola. 2008. Density Contrast SPH Interfaces. In *Eurographics/SIGGRAPH Symposium on Computer Animation*, Markus Gross and Doug James (Eds.). The Eurographics Association. https://doi.org/10.2312/SCA/SCA08/211-218

Alexandre Tartakovsky and Paul Meakin. 2005. Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Physical Review E* 72, 2 (aug 2005), 026301. https://doi.org/10.1103/physreve.72.026301

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

Mengdi Wang, Yitong Deng, Xiangxin Kong, Aditya H. Prasad, Shiying Xiong, and Bo Zhu. 2021. Thin-Film Smoothed Particle Hydrodynamics Fluid. *ACM Transactions on Graphics* 40, 4 (2021), 1–16.

Xiao-Kun Wang, Xiao-Juan Ban, Ya-Lan Zhang, 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, 6 (nov 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* 37, 2 (2018).

Jingrui Xing, Liangwang Ruan, Bin Wang, Bo Zhu, and Baoquan Chen. 2022. Position-Based Surface Tension Flow. *ACM Transactions on Graphics* 41, 6 (nov 2022), 1–12. 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. In *Volume 6: 15th International Conference on Multibody Systems, Nonlinear Dynamics, and Control*. American Society of Mechanical Engineers. https://doi.org/10.1115/detc2019-98124

Sheng Yang, Xiaowei He, Huamin Wang, Sheng Li, Guoping Wang, Enhua Wu, and Kun Zhou. 2016a. Enriching SPH simulation by approximate capillary waves.. In *Symposium on Computer Animation*. 29–36.

Tao Yang, Ming Lin, Ralph Robert Martin, Jian Chang, and Shimin Hu. 2016b. Versatile interactions at interfaces for SPH-based simulations. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2016)*. Association for Computing Machinery, 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 (oct 2017), 2235–2247. https://doi.org/10.1109/tvcg.2017.2706289

Mingyu Zhang. 2010. Simulation of surface tension in 2D and 3D with smoothed particle hydrodynamics method. *J. Comput. Phys.* 229, 19 (sep 2010), 7238–7259. https://doi.org/10.1016/j.jcp.2010.06.010

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 (aug 2012), 1281–1289. https://doi.org/10.1109/tvcg.2011.141

Wen Zheng, Jun-Hai Yong, and Jean-Claude Paul. 2009. Simulation of bubbles. *Graphical Models* 71, 6 (nov 2009), 229–239. https://doi.org/10.1016/j.gmod.2009.08.001

Wen Zheng, Bo Zhu, Byungmoon Kim, and Ronald Fedkiw. 2015. A new incompressibility discretization for a hybrid particle MAC grid representation with surface tension. *J. Comput. Phys.* 280 (jan 2015), 96–142. https://doi.org/10.1016/j.jcp.2014.08.051

Bo Zhu, Minjae Lee, Ed Quigley, and Ronald Fedkiw. 2015. Codimensional Non-Newtonian Fluids. *ACM Transactions on Graphics* 34, 4 (2015), 1–9.

Bo Zhu, Ed Quigley, Matthew Cong, Justin Solomon, and Ronald Fedkiw. 2014. Codimensional surface tension flow on simplicial complexes. *ACM Transactions on Graphics* 33, 4 (July 2014), 1–11. 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 (mar 2020), 23. https://doi.org/10.3390/computers9020023