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

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.


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

ACM Reference Format:

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
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 modeled 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 IIISPH 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 [Aanjanyea 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 [Azenot et al. 2013] 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...
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 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

\[ f_{st}(x) = -\sigma \int_{\Omega} \frac{x-y}{||x-y||} W(x-y; \tilde{h}) dy, \] (1)

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

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

Intuitively speaking, this integral accumulates force contributions of all points \( y \) in the domain in the direction of evaluation point \( x \) 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 \( W_{\text{cubic}} \) to be the cubic spline kernel [Mongaghan 1992] often used in SPH simulations, with a small modification resulting in

\[ W(x - y; \tilde{h}) = ||x - y|| W_{\text{cubic}}(x - y; \tilde{h}), \] (2)

\[ W_{\text{cubic}}(x - y; \tilde{h}) = \begin{cases} \frac{W_{\text{cubic}}(x - y; \tilde{h})}{||x - y|| d; \tilde{h}}, & \text{for } ||x - y|| > d \\ 0, & \text{otherwise}. \end{cases} \] (3)

Here \( d \) denotes the particle diameter, such that the value of \( W_{\text{cubic}} \), 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 \( \theta = \frac{12}{||x - y||} \) (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 \( ||x - y|| \) 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

\[ f_{st}(x_i) = -\sigma \sum_{j \in N_i} \frac{m_j}{\rho_j} (x_i - x_j) W_{\text{cubic}}(x_i - x_j; \tilde{h}), \] (4)

where \( m_j \) is the mass and \( \rho_j \) is the density of discrete particle \( j \) in the neighborhood \( N_i \) around position \( x_i \). Note that due to the dependency on the density \( \rho_j \), the term also depends on all positions \( x_k \) in the neighborhood \( k \in N_j \), since

\[ \rho_j = \sum_{k \in N_j} m_k W_{\text{cubic}}(x_j - x_k; \tilde{h}). \] (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 \( W_{ij} = W_{\text{cubic}}(x_i - x_j; \tilde{h}) \) and \( x_{ij} = x_i - x_j \). In order to symmetrize Eq. (4), we introduce \( \tilde{\rho}_{ij} = \frac{1}{2}(\rho_i + \rho_j) \) and \( \tilde{m}_{ij} = \frac{1}{2}(m_i + m_j) \) as the averaged densities...
and masses. This yields the symmetric force
\[ \mathbf{f}_{ij}(\mathbf{x}_i) = -\sum_j m_{ij} \frac{\mathbf{x}_{ij}}{p_{ij}} W_{ij}. \] (6)

Finally, the acceleration on particle \( i \) resulting from the discretized surface tension force can be written as
\[ a_{i,\text{st}} = -\frac{1}{m_i} \sum_j m_{ij} \mathbf{f}_{ij}. \] (7)

Discretizing this equation in time using the forward Euler method yields
\[ a_{i,\text{st}} = \frac{\mathbf{v}_{i}^{t+1} - \mathbf{v}_{i}^{t}}{\Delta t} = -\frac{\sigma}{m_i} \sum_j \frac{m_{ij}}{p_{ij}} \mathbf{x}_{ij} W_{ij}^{t}. \] (8)
resulting in a closed expression for \( \mathbf{v}_{i}^{t+1} \). Discretizing with the backward Euler method instead, this equation becomes
\[ a_{i,\text{st}} = \frac{\mathbf{v}_{i}^{t+1} - \mathbf{v}_{i}^{t}}{\Delta t} = -\frac{\sigma}{m_i} \sum_j \frac{m_{ij}}{p_{ij}} \mathbf{x}_{ij} W_{ij}^{t+1}, \] (9)
which results in a non-linear system for the velocity \( \mathbf{v}_{i}^{t+1} \), since the (non-linear) kernel function and the density both depend on \( \mathbf{x}_{i}^{t+1} = \mathbf{x}_{i}^{t} + \Delta t \mathbf{v}_{i}^{t+1} \).

### 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 \( \mathbf{v}_{i}^{t+1} \), while the implicit version in Eq. (9) requires solving a non-linear system.

The usage of a fully 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
\[ a_{i,\text{st}}(\mathbf{x}_{i}^{t+1}(\mathbf{v}_{i}^{t+1})) = a_{i,\text{st}}(\mathbf{x}_{i}^{t+1}(\mathbf{v}_{i}^{t})) + \nabla_{\mathbf{v}} a_{i,\text{st}}(\mathbf{x}_{i}^{t+1}(\mathbf{v}_{i}^{t}))|_{\mathbf{v}_{i}^{t}} (\mathbf{v}_{i}^{t+1} - \mathbf{v}_{i}^{t}) + O(\Delta t^2). \] (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 as 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 \( h \)
\[ \frac{\mathbf{v}_{i}^{t+1} - \mathbf{v}_{i}^{t}}{\Delta t} = -\frac{\sigma}{m_i} \sum_j \frac{m_{ij}}{p_{ij}} W_{ij}^{t+1} \mathbf{x}_{ij}^{t+1}, \] (11)
and linearize them independently with respect to \( t \). Realizing that the term \( h \) only depends linearly on particle positions \( \mathbf{x}_i \) and \( \mathbf{x}_j \), it is linearized using backward Euler time integration. This results in
\[ h_{ij}^{t+1} = \left( \mathbf{x}_i^{t+1} + \Delta t \mathbf{v}_i^{t+1} - \mathbf{x}_j^{t} - \Delta t \mathbf{v}_j^{t+1} \right), \] (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 \( t \)
\[ g_{ij}^{t+1} \approx g_{ij}^{t+1} = g_{ij}^{t} + \Delta t \frac{\partial g_{ij}}{\partial \mathbf{v}}|_{\mathbf{v}_{i}^{t}} = g_{ij}^{t} + \Delta t \left( \frac{\partial g_{ij}}{\partial \mathbf{v}} \cdot \mathbf{v}_{i}^{t} \right)|_{t}. \] (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
\[ g_{ij}^{t} = \frac{W_{ij}^{t}}{p_{ij}^{t}}, \] (14)
and
\[ \left( \frac{\partial g_{ij}}{\partial \mathbf{v}} \right)|_{\mathbf{v}_{i}^{t}} = \frac{\nabla_{\mathbf{v}} W_{ij}^{t} \cdot \mathbf{v}_{i}^{t} + \nabla_{\mathbf{v}} W_{ij}^{t} \cdot \mathbf{v}_{j}^{t}}{p_{ij}^{t}} \]
\[ = -\frac{W_{ij}^{t}}{2} \left( \sum_{k \in \mathcal{N}_i} m_{ik} (\mathbf{v}_{i}^{t} - \mathbf{v}_{k}^{t}) \cdot \nabla_{\mathbf{v}} W_{ik}^{t} \right) + \frac{\sum_{l \in \mathcal{N}_i} m_{ij} (\mathbf{v}_{j}^{t} - \mathbf{v}_{l}^{t}) \cdot \nabla_{\mathbf{v}} W_{il}^{t}}{p_{ij}^{t}^2}. \] (15)

The first line in this equation corresponds to the time derivative of the kernel function \( W_{ij}^{t} \), 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 \( \nabla_{\mathbf{v}} \) and \( \nabla \) denote derivatives wrt. particle positions \( \mathbf{x}_i \) and \( \mathbf{x}_j \). The second and third line correspond to the time derivative of the reciprocal averaged density \( p_{ij}^{-1} \), which has been expanded analogously. These terms

ACM Trans. Graph., Vol. 43, No. 1, Article 13. Publication date: November 2023.
can be recognized from the continuity equation as the negative divergence of the velocity field \( \nabla \cdot \mathbf{v} \) times density \( \rho \):

\[
-(\rho \mathbf{v} \cdot \mathbf{v})_i = \sum_{j \in N_i} m_j (v_i - v_j) \cdot \nabla_i W_{ij}.
\]

We precompute and store this scalar value per particle and thus can assemble the final linearized system

\[
\frac{\sigma^{t+1} - \sigma^t}{\Delta t} = -\frac{\sigma}{m_i} \sum_{j \in N_i} \tilde{m}_{ij} \left( x_{ij}^t + \Delta t \sigma^{t+1}_{ij} \right) \tilde{g}_{ij}^{t+1},
\]

\[
\Rightarrow \sigma^{t+1} + \Delta t^2 \frac{\sigma}{m_i} \sum_{j \in N_i} \tilde{m}_{ij} x_{ij}^t \tilde{g}_{ij}^t = \sigma^t - \Delta t \frac{\sigma}{m_i} \sum_{j \in N_i} \tilde{m}_{ij} x_{ij}^t \tilde{g}_{ij}^t.
\]

This can also be written as

\[
(I + \Delta t A) \sigma^{t+1} = \sigma^t + b,
\]

where

\[
A_{ii} = I_3 \frac{\sigma}{m_i} \Delta t \sum_{j \in N_i} \tilde{m}_{ij} \tilde{g}_{ij}^t,
\]

\[
A_{ij} = -I_3 \frac{\sigma}{m_i} \Delta t \tilde{m}_{ij} \tilde{g}_{ij}^t \text{ for } i \neq j,
\]

\[
b_i = -\Delta t \frac{\sigma}{m_i} \sum_{j \in N_i} \tilde{m}_{ij} x_{ij}^t \tilde{g}_{ij}^t.
\]

Here, \( A_{ij} \) denotes the matrix block corresponding to the entry of particles \( i \) and \( j \), while \( A_{ii} \) denotes the diagonal matrix element and \( I_3 \) 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

\[
v_{\text{pred}} = \mathbf{v}^t + (\mathbf{v}^t - \mathbf{v}^{t-1}).
\]

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 \( \sigma^b \), 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 \( V^b_t \) occupied within the compact support of particle \( i \), the closest point on the boundary \( x^b_i \), and the velocity of the boundary \( \mathbf{v}^b_i \). 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

\[
A_{ii} = I_3 \frac{\Delta t}{m_i} \sum_{b \in N^b_i} \tilde{m}_i b \cdot \tilde{g}_{i}^{b, t+1} + \sigma \sum_{j \in N_i} \tilde{m}_{ij} \tilde{g}_{ij}^t,
\]

\[
b_i = -\Delta t \frac{\sigma}{m_i} \sum_{j \in N_i} \tilde{m}_{ij} x_{ij}^t \tilde{g}_{ij}^t + \sigma \sum_{b \in N^b_i} \tilde{m}_i b \cdot \tilde{g}_{i}^{b, t+1}.
\]

where \( b \in N^b_i \) denotes all neighboring boundaries of particle \( i \), and \( \tilde{m}^b_i = \frac{1}{2} (m_i + \rho_0 V^b_i) \) denotes the averaged mass. Here \( V^b_i \) 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 \( A_{ij} \) remains unchanged and that the linear system remains symmetric for uniform particle masses. The term \( \tilde{g}_{i}^{b, t+1} \) looks slightly different,

\[
\tilde{g}_{i}^{b, t+1} = \frac{W^b_i}{2 (\rho_i^t + \rho_0)} \mathbf{v}^b_i + \frac{W^b_i}{2 (\rho_i^t + \rho_0)} \mathbf{v}^b_i \cdot \nabla_i W^b_i + \frac{W^b_i}{2 (\rho_i^t + \rho_0)} \frac{\sum_{k \in N_i} m_k (v_i^t - v_k^t) \cdot \nabla_i W_{ik}^b}{\left( \frac{3}{2} \left( \rho_i^t + \rho_0 \right) \right)^2}.
\]

Here the term corresponding to the boundary density 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 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 better than the implicit model. Using an explicit viscosity model is problematic for convergence, we expect the strong coupling 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).

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

$$\frac{\mathbf{v}_i^{t+1} - \mathbf{v}_i^t}{\Delta t} = f^{t+1}_\text{st} + f^{t+1}_\text{visc}. \quad (23)$$

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

$$(\mathbf{I} + \Delta t (\mathbf{A}_\text{st} + \mathbf{A}_\text{visc} )) \mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \mathbf{b}_\text{st}, \quad (24)$$

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

$$\mathbf{A}_{\text{visc},ij} = 2(\delta + 2) \mu \frac{m_{ij}}{r_i^2 r_j^2} \nabla W_{ij} x_i x_j^T + \mathbf{A}_{\text{visc},ii} = - \sum_{j \in N_i} \mathbf{A}_{\text{visc},ij},$$

where $\mu$ is the dynamic viscosity coefficient, $\delta$ the number of spatial dimensions and $\delta$ 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.

**Algorithm 1 Implicit Surface Tension with Viscosity**

1. **procedure** COMPUTE_SURFACE_TENSION_WITH_VISCOSITY
2. for all particles $i$ do
3. $$(\rho \nabla \cdot \mathbf{v})_i^t \leftarrow \text{PRECOMPUTE_DIVERGENCE} \quad \triangleright \text{Eq. (16)}$$
4. $\mathbf{h}_i^t \leftarrow \text{COMPUTE_COHESION_ADHESION_RHS} \quad \triangleright \text{Eq. (21)}$
5. $\mathbf{v}_i^t \leftarrow \text{COMPUTE_WARMSTART} \quad \triangleright \text{Eq. (20)}$
6. $\mathbf{v}_i^{t+1} \leftarrow \text{SOLVE_CG}(\mathbf{h}_i^t, \mathbf{v}_i^t, \mathbf{v}_i^{t\text{pred}}) \quad \triangleright \text{Eq. (24)}$
7. for all particles $i$ do
8. $\mathbf{a}_i \leftarrow \mathbf{a}_i + \mathbf{v}_i^{t+1} - \mathbf{v}_i^t \Delta t \quad \triangleright \text{Add acceleration}$

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 $\sigma = [1625, 3250, 6500] \text{N m}^{-1}$, while the rows from top to bottom show increasing adhesion coefficients $\sigma^h = [0\sigma, 0.7\sigma, 0.9\sigma] \text{N m}^{-1}$ as a fraction of the surface tension coefficient.

relative residual error tolerance $\text{tol} = 0.001$ and a maximum number of iterations $\text{iter}_\text{max} = 100$.

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 2 Simulation Step

1: procedure SIMSTEP
2: for all particles \( i \) do
3: \( N_i \leftarrow \text{FindNeighboringParticles}(i) \)
4: for all particles \( i \) do
5: \( \rho_i \leftarrow \text{ComputeDensity}(i) \)
6: \( v^*_i \leftarrow v_i + \Delta t g \) \hspace{1em} \( \triangleright \) Add gravity accel.
7: \( v_i^{**} \leftarrow v_i^* + \Delta t a_i^{**} \) \hspace{1em} \( \triangleright \) Add explicit non-pressure forces
8: while \( \text{iter} < \text{iter}_{\text{max}} \) do
9: \( \sigma^* \leftarrow \text{DFSPHDivergenceFree}(\sigma^*) \)
10: \( \sigma^* \leftarrow \text{Algorithm 1} \) \hspace{1em} \( \triangleright \) Implicit surface tension
11: \( v^* \leftarrow v^* + \Delta t a_i^{**} \)
12: \( v^* \leftarrow \text{DFSPHConstantDensity}(\sigma^*) \)
13: for all particles \( i \) do
14: \( v_i \leftarrow v_i^* \) \hspace{1em} \( \triangleright \) Update velocity
15: \( x_i \leftarrow x_i + \Delta t v_i \) \hspace{1em} \( \triangleright \) Update positions

The algorithm is shown in Algorithm 2. Weakly coupled operator splitting can be recovered when setting \( \text{iter}_{\text{max}} = 1 \). Results comparing 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].

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 \( \text{iter}_{\text{max}} = 5 \) 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].

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 five 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.

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 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.

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.

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].

<table>
<thead>
<tr>
<th></th>
<th>Double Droplet</th>
<th>Droplet Crown</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Δt Avg. Runtime</td>
<td>Δt Avg. Runtime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours (sc)</td>
<td>2.0 1.90</td>
<td>2.0 22.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours (wc)</td>
<td>2.0 3.06</td>
<td>2.0 36.57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Akinci et al</td>
<td>1.0 2.91</td>
<td>2.0 32.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BT</td>
<td>0.5 1.51</td>
<td>2.0 21.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>He et al.</td>
<td>0.5 1.92</td>
<td>2.0 32.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zorilla et al.</td>
<td>0.8 24.21</td>
<td>2.0 436.10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

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 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.
This results in the formation of a stable bell-like shape, which can visualize the interior. Parameters: vertical emitters meet. The particle view is cut open by a clipping plane to visualize the interior. Parameters: \( \Delta t = 0.001 \text{s}, \) particle radius = 0.015 m, density = 1000 kg m\(^{-3}\), \( \nu = 0.2 \text{m}^2 \text{s}^{-1}, \sigma = 7560 \text{N} \text{m}^{-1}. \)

around the bottom of the sphere and a less uniform stream dripping down. In addition, the particle coloring, indicating the velocity 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.

**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.

**Dripping Faucet.** 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: \( \Delta t = 0.001 \text{s}, \) particle radius = 0.025 m, density = 1000 kg m\(^{-3}\), \( \nu = 0.002 \text{m}^2 \text{s}^{-1}, \sigma = 8400 \text{N} \text{m}^{-1}. \)

**Droplet Toy.** In Fig. 1 we show 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 $= 5 \times 10^{-5}$ m, density $= 1000$ kg m$^{-3}$, $\nu = 5 \times 10^{-6}$ m$^2$ s$^{-1}$, $\sigma = 0.02$ N m$^{-1}$, $\sigma$ = 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 Sølenthaler 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 smooth, 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 [Loschechner 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


The inside of the bubbles are filled with air particles, while the film is made up of different soap particles. Parameters: $\Delta_t = 0.002$ s, particle radius = 0.004 m, density $= 1000 \text{ kg m}^{-3}$, $\nu = 0.02 \text{ m}^2 \text{s}^{-1}$, $\sigma = 107.52 \text{ N m}^{-1}$.

https://doi.org/10.101145/1833551.1778853


