Position-Based Surface Tension Flow

JINGRUI XING*, SIST & KLMP (MOE), Peking University, China
LIANGWANG RUAN*, SCS & KLMP (MOE), Peking University, China
BIN WANG†, Beijing Institute for General Artificial Intelligence, China
BO ZHU, Dartmouth College, United States of America
BAOQUAN CHEN†, SIST & KLMP (MOE), Peking University, China

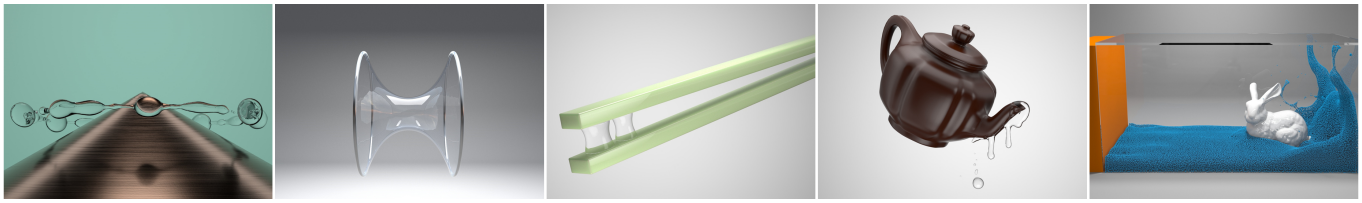


Fig. 1. Our computational framework can realize various small-scale surface tension phenomena, including thin films, droplets, and intricate contact dynamics, by enhancing a position-based fluid framework. From left to right, the images present: dynamics of a droplet impacting on a cone, fluid catenoid, tweezers for droplets, teapot effect, and dam breaking with high surface tension.

This paper presents a novel approach to simulating surface tension flow within a position-based dynamics (PBD) framework. We enhance the conventional PBD fluid method in terms of its surface representation and constraint enforcement to furnish support for the simulation of interfacial phenomena driven by strong surface tension and contact dynamics. The key component of our framework is an on-the-fly local meshing algorithm to build the local geometry around each surface particle. Based on this local mesh structure, we devise novel surface constraints that can be integrated seamlessly into a PBD framework to model strong surface tension effects. We demonstrate the efficacy of our approach by simulating a multitude of surface tension flow examples exhibiting intricate interfacial dynamics of films and drops, which were all infeasible for a traditional PBD method.

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

Additional Key Words and Phrases: fluid simulation, surface tension, position-based dynamics

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

*Joint first authors
†Corresponding authors

Authors' addresses: Jingrui Xing, SIST & KLMP (MOE), Peking University, Beijing, China, xjr01@hotmail.com; Liangwang Ruan, SCS & KLMP (MOE), Peking University, Beijing, China, ruanliangwang@pku.edu.cn; Bin Wang, Beijing Institute for General Artificial Intelligence, Beijing, China, binwangbuaa@gmail.com; Bo Zhu, Dartmouth College, Hanover,NH, United States of America, bo.zhu@dartmouth.edu; Baoquan Chen, SIST & KLMP (MOE), Peking University, Beijing, China, baoquan@pku.edu.cn.

0730-0301/2022/12-ART244 $15.00
https://doi.org/10.1145/3550454.3555476

1 INTRODUCTION

The position-based fluid simulation, after its invention by Macklin and Müller [2013], has been enjoying its multifaceted advantages regarding the implementation easiness, computational performance, and coupling friendliness in many computer graphics applications. The cornerstone of the framework is a particle-based discretization with a density constraint on each particle to control the weighted average of the number density in its vicinity. These particle density constraints are further enforced via projection iterations, which support robust, large-timestep simulations of various constraint physical systems such as soft bodies [Müller et al. 2007], cloth [Kim et al. 2012], hair [Müller et al. 2012], rods [Umetani et al. 2015], and fluid [Macklin et al. 2014].

Despite the advent of PBD on different fronts of physical simulation, its potential for simulating interfacial flow phenomena remains largely unexplored. Many visually appealing fluid systems driven by strong interfacial forces, ranging from films to bubbles and from small drops on impact to liquid bridges in a tweezer, stay on the periphery of the current PBD research and therefore endure their insulation from the elongating list of the multi-physics simulations supported by PBD. In the first position-based fluid framework [Macklin and Müller 2013], the fluid's surface tension effects were realized with an artificial pressure term to cohere the boundary particles inward. This numerical treatment enjoys its easiness for implementation yet suffers from the limited range of parameters and, therefore, scope of phenomena being supported. On the one hand, a conventional PBD method cannot simulate thin fluid phenomena such as films or bubbles because of the undistinguished treatment of surface and interior particles. On the other hand, because this mechanic was implemented jointly with the artificial pressure term, it can only support small or moderate surface tension simulation without suffering from any numerical instabilities. Adding the surface tension as an explicit force term like SPH [Akinci et al. 2013; Wang et al. 2021] or MPM [Chen et al. 2021] does not fit PBD's constraint-based nature and will not alleviate such instabilities.

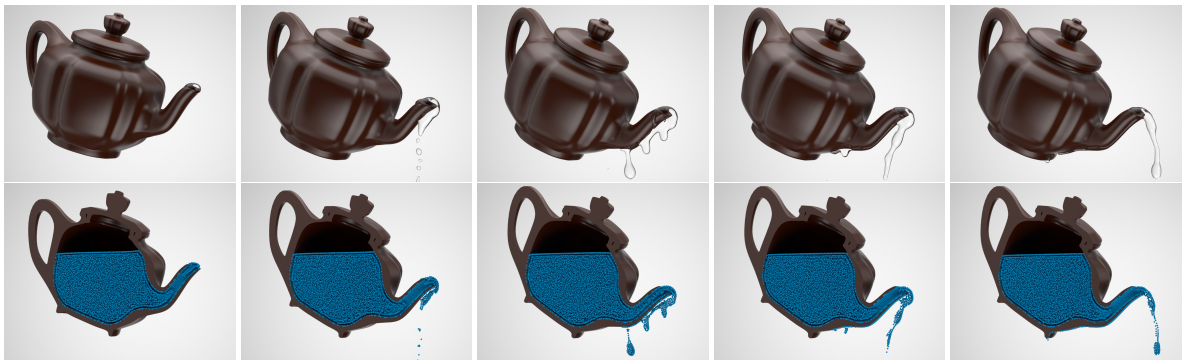


Fig. 2. Teapot effect: from left to right, as the teapot gradually tilts, the liquid in it flows out with increasing speed. When the flow rate is low, the liquid tends to dribble down the outside of the spout. Top: rendered by a reconstructed mesh; bottom: rendered by original particles.

To address these challenges in terms of interface representation and constraint formulation, we proposed a novel approach to extending the current PBD framework to support strong surface tension flow simulation. We tackle the problem on two fronts. First, we augment the current particle-based surface representation with a *local* (and imperfect) mesh data structure, which boosts the model's geometric expressiveness regarding the liquid surface and film capturing and accuracy regarding the surface tension solving, with a marginal computational overhead. Second, we build an area-minimization constraint for each surface particle based on this local mesh representation and incorporate these constraints into a standard PBD framework. With these two extensions, we enhance the geometric complexities and enlarge the parameter range of a traditional position-based fluid solver, which leads to the support of many surface tension phenomena that were previously impractical due to either the interface's geometric complexities or computational cost. Our local mesh scheme was motivated by a series of recent works in solving numerical PDEs on a point-set discretization [Lai et al. 2013], the hybridization of meshes and particles in simulating moving particle semi-implicit (MPS) fluid [Matsunaga et al. 2020], as well as the many mesh-based front-tracking and simulation methods in computer graphics [Clausen et al. 2013; Da et al. 2015, 2016; Wojtan et al. 2011; Yu et al. 2012]. To the best of our knowledge, our approach is the first to leverage local mesh connectivities to facilitate the particle fluid's interfacial representation, which combines the computational merits of both mesh-based and particle-based methods and extends the traditional PBD fluid to support rich surface tension effects.

We summarize our main contributions as follows:

- The first hybrid particle and local mesh representation for modeling Lagrangian interfacial fluid;
- A novel constraint formulation based on local area minimization for the PBD framework;
- The first position-based fluid simulation approach that supports small-scale surface tension phenomena.

2 RELATED WORK

Position-based Dynamics. Müller et al. [2007] first introduce the position-based dynamics (PBD) framework to enable real-time soft-body simulation in computer graphics. By directly manipulating each particle's position, PBD can achieve robust simulations under large timesteps, which further enables the simulation of a broad spectrum of materials including deformable solids [Bender et al. 2014], cloth [Kim et al. 2012], curly hair [Müller et al. 2012], rods [Umetani et al. 2015], sands[Macklin et al. 2014], fluids [Macklin and Müller 2013], and various coupling effects [Frâncu and Moldoveanu 2017; Robinson-Mosher et al. 2008; Rumman et al. 2020]. We refer readers to [Bender et al. 2017] for a comprehensive survey on PBD simulations. Among these pieces of work, Macklin and Müller [2013] first use PBD to simulate fluid, in which a density constraint augmented with an artificial pressure term for penalizing abnormal particle gathering is added to each particle to enforce incompressibility and mimic surface tension. Alduán et al. [2017] further extend the PBD fluid method with the density contrast formulation to accommodate fluids with different densities and resolutions. A traditional position-based fluid method cannot realize authentic surface tension effects by naively increasing the artificial pressure or the number of iterations, which limits its ability in modeling many small-scale fluid scenes such as realistic films, bubbles, and drops. To the best of our knowledge, there exists no effective constraint in the PBD framework for surface tension, by neither minimizing surface area nor calculating curvature.

Lagrangian front tracking. A Lagrangian front tracking algorithm maintains a moving Lagrangian data structure, traditionally with simplicial meshes [Da et al. 2016; Wojtan et al. 2010; Zhu et al. 2014] and recently with point sets [Wang et al. 2020, 2021], to capture the evolution of a dynamic interface and solve physics on it. Compared with grid-based interface tracking methods such as level sets [Foster and Fedkiw 2001; Osher and Sethian 1988; Sethian and Smereka 2003] or volume of fluid (VOF) [Hirt and Nichols 1981], mesh-based approaches manifest their unique merits regarding their highly detailed surface geometry and flow dynamics. For example, Thürey et al. [2010] combine a high-resolution triangle mesh with a low-resolution grid to obtain a detailed simulation of capillary effects on liquid surface. Brochu et al. [2010] use a surface mesh to guide the placement of additional pressure samples in a Voronoi discretization to capture thin fluid structures such as films and splashes. Yu et al. [2012] use a temporally coherent surface mesh, advected with particle velocity field, for both the fluid interface and the fluid properties

tracking. Zhu et al. [2015, 2014] use a non-manifold simplicial mesh to track and solve fluid volumes with codimensional features. Da et al. [2016] simulate volumetric fluids with only the degrees of freedom on the surface mesh, which is later extended to simulate ferrofluid [Huang and Michels 2020] and large-scale water surfaces [Huang et al. 2021]. Most of these methods rely on a manifold mesh to track the dynamic fluid surface and calculate its surface tension force. The simulation quality highly relies on the mesh quality. Any mesh connectivity inconsistencies, which will probably lead to holes on the surface, will quickly expand and break the fluid film due to the contracting surface tension force on the rim. To maintain such a good-quality mesh is costly, requiring mesh repair operations [Brochu et al. 2010] or global remeshing schemes [Zheng et al. 2015] in each timestep. This meshing overhead can be alleviated by dynamic point-set alternatives [Wang et al. 2020, 2021], at the expense of implementing a new set of codimensional differential operators discretized on moving points.

Lagrangian surface tension. Surface tension is ubiquitous in fluid phenomena on small scales. The existing Lagrangian methods can be mainly categorized into three groups according to their different ways of calculating the surface tension force. The first category simulates surface tension based on inter-particle cohesive forces [Becker and Teschner 2007; Clavet et al. 2005; Kim et al. 2021; Tartakovsky and Meakin 2005]. These methods avoid computing surface curvature and only use pairwise forces between particles, thus ensuring momentum conservation. The accuracy of the cohesion-based method largely depends on a full ring of neighboring particles in the vicinity, which limits its ability to model thin flow features such as films and membranes. The second category approximates surface tension with the local normal and curvature [Akinci et al. 2013; Hyde et al. 2020; Müller et al. 2003; Wang et al. 2020, 2021; Zhang 2010]. These methods rely on a robust boundary detection algorithm and an accurate reconstruction scheme (e.g., moving least squares) to approximate the local geometry [Hyde et al. 2020; Wang et al. 2020, 2021; Zhang 2010]. The third category is to calculate the surface tension force on a surface mesh [Da et al. 2016; Ruan et al. 2021; Yu et al. 2012; Zhang et al. 2011], allowing an accurate approximation of the local surface and, therefore, the surface tension calculation. Another benefit of using an explicit mesh is its easiness for thin feature representation. Thus, mesh-based surface tension models support an ensemble of interesting bubble and film simulations (e.g., [Batty et al. 2012; Da et al. 2015; Ishida et al. 2017; Zhang et al. 2012; Zhu et al. 2015, 2014]). The main bottleneck of a mesh-based method is the computational cost of generating a high-quality mesh, whose implementation typically cannot be parallelized.

3 BACKGROUND

We briefly introduce the position-based fluid framework [Macklin et al. 2014] for completeness. In a PBD fluid solver, the fluid volume is discretized as a set of particles. Each particle carries the same mass m and position $\mathbf{p}_i$. For particle i, we compute its density using the standard SPH density estimator:

$$\rho_i = \sum_{j \in N(i)} m W(\mathbf{p}_i - \mathbf{p}_j, h), \tag{1}$$

where W is the kernel function, h is the kernel radius, $N(i) = \{j_1, \cdots, j_{k_i}, i\}$ is the set of neighboring particles within the kernel radius and i itself. We use the cubic spline kernel [Koschier et al. 2019] and set h to be 6 times particle radius in our simulation.

The fluid incompressibility is enforced by a density constraint C_i^ρ applied on each particle as

$$C_i^\rho(\mathbf{p}_i, \mathbf{p}_{j_1}, \cdots, \mathbf{p}_{j_{k_i}}) = \frac{\rho_i}{\rho_0} - 1, \tag{2}$$

where ρ_0 is the density at the initial configuration. The density constraints are solved using the Newton-Raphson method. For each constraint C_i^ρ, a position correction along the constraint gradient direction $\Delta\mathbf{p} = \lambda_i \nabla C_i^\rho$ is applied to make $C_i^\rho(\mathbf{p} + \Delta\mathbf{p}) = 0$. By a first-order Taylor expansion of C_i^ρ, we solve λ_i as:

$$\lambda_i = -\frac{C_i^\rho(\mathbf{p})}{\sum_{k \in N(i)} \|\nabla_k C_i^\rho(\mathbf{p})\|^2 + \varepsilon}. \tag{3}$$

The ε in the denominator is a user-specified relaxation parameter (also called constraint force mixing (CFM) [Smith 2004]) which is tuned to obtain different simulation effects.

Surface tension treatment. The surface tension effect is realized by an artificial pressure term. Summing up the contributions from the density constraint of each neighboring particle as well, the total position correction $\Delta\mathbf{p}_i$ for particle i is:

$$\Delta\mathbf{p}_i = \frac{m}{\rho_0} \sum_{j \in N(i)} \left(\frac{s_{\text{corr}}}{m} + \lambda_i + \lambda_j \right) \nabla W(\mathbf{p}_i - \mathbf{p}_j, h), \tag{4}$$

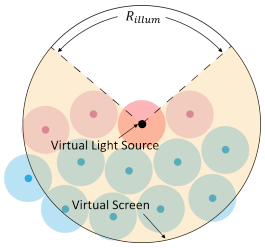
in which $s_{\text{corr}} = -k(W(\mathbf{p}_i - \mathbf{p}_j, h)/W(\Delta\mathbf{q}, h))^n$ with $\|\Delta\mathbf{q}\|$ as a fixed distance smaller than h is the so-called artificial pressure term. As neighboring particles i and j get closer to each other, s_{corr} increases sharply and behaves as a repulsive force to prevent abnormal particle clustering. Consequently, boundary particles pull their neighbors inwards, which produces the surface-tension-like surface minimization effect, with its strength controlled by the value of k.

4 ALGORITHM

The main body of our algorithm consists of four components: surface particle detection, local mesh construction, surface tension constraints, and time integration. The implementation of these four components can be based on a standard position-based fluid framework. We introduce each of the components in the following sections.

4.1 Surface Particle Detection

The first step of our approach is to identify the surface particles and calculate their surface normals. Our algorithm is based on the light shading method proposed in [Shibata et al. 2015]. As shown in the inset figure, we place a point light source in the center of each particle casting light rays to a spherical virtual screen surrounding it. An interior particle is detected if the particle's screen is



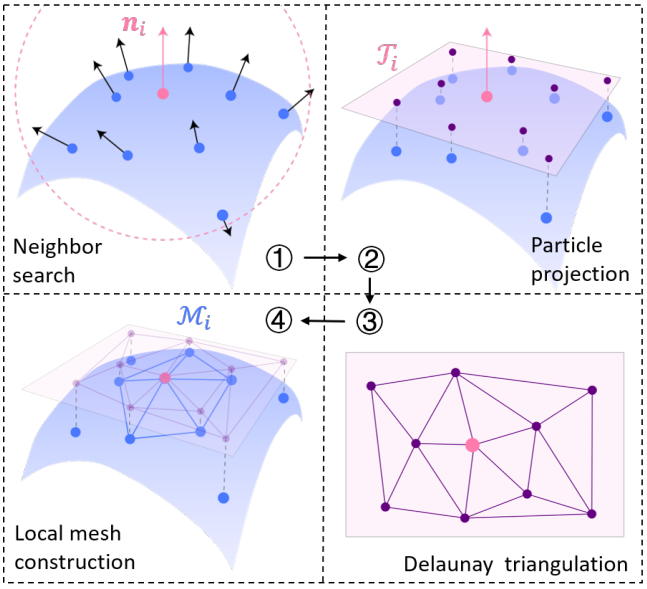


Fig. 3. Our local mesh construction algorithm with four sub-steps.

all shadowed by its neighbors. And a boundary particle is detected if at least one part of the particle's screen is not shadowed.

Mathematically, this process is modeled by a scalar R_{illum} that denotes the fraction of each source particle's overall illuminated area. We use the threshold $\beta = 1/9$ to classify surface and interior particles. We refer the readers to Supplementary 1 for implementation details.

Normal calculation. After detecting all the surface particles, we further estimate their normal vectors by calculating the color gradient on each particle. This step follows the SPH normal calculation method proposed in [Müller et al. 2003]. We define the color field in position $\mathbf{p}$ as:

$$c(\mathbf{p}) = \sum_{j \in N(\mathbf{p})} \frac{m}{\rho_j} W(\mathbf{p} - \mathbf{p}_j, h), \tag{5}$$

where $N(\mathbf{p})$ is the set of neighboring particles around $\mathbf{p}$ within the kernel radius. This function indicates 1 inside the fluid, 0 outside the fluid, and manifests a smooth transition near the boundary. We compute the normal vector as the color gradient:

$$\mathbf{n}_i^* = -\nabla c(\mathbf{p}_i) = -\sum_{j \in N(i)} \frac{m}{\rho_j} \nabla W(\mathbf{p}_i - \mathbf{p}_j, h), \ \mathbf{n}_i = \frac{\mathbf{n}_i^*}{\|\mathbf{n}_i^*\|}. \tag{6}$$

It is worth noting that the color gradient was used in [Müller et al. 2003] also for surface particle detection. We did not choose to do so because we notice a gradient-based approach tends to detect a narrow band of particles rather than a sharp layer, which does not fit the requirement of our local meshing algorithm.

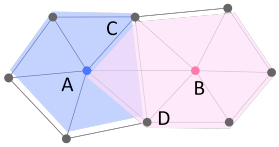
4.2 Local Mesh Construction

After labeling all the surface particles, we will represent the local geometry of each particle (i.e., their local area, shape, and curvature) by building a local mesh structure based on its neighboring particles. In particular, we build a full ring of triangles centered around the given particle as the *local* mesh representation, without considering the triangle rings in other particles, to feature the geometry in the particle's vicinity on the surface. It is worth mentioning that the local mesh construction is carried out for every surface particle in parallel, and each particle will maintain its incident triangles separately without checking the triangle redundancy or inconsistency across different particles. This local mesh is further used to establish the surface tension constraints and their sensitivities for each particle.

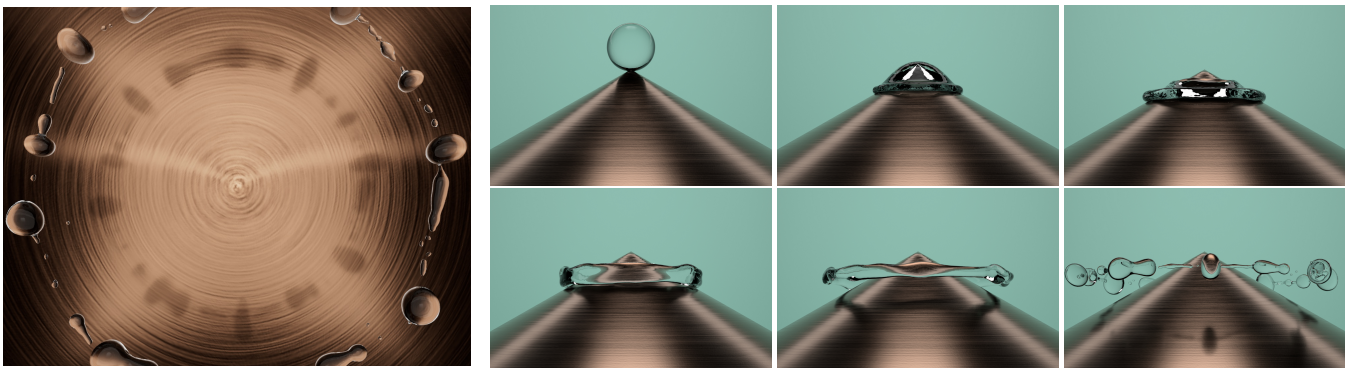
As shown in Figure 3, our local mesh construction algorithm consists of four steps, including neighbor search, particle projection, planar Delaunay triangulation, and mesh construction.

(1) **Neighbor search**: For each surface particle i, we find all the neighboring surface particles within a kernel radius through the nearest neighbor search, which is denoted as $\mathcal{K}_i$.
(2) **Particle projection**: $\mathcal{K}_i$ and $\mathbf{p}_i$ are projected onto a tangential plane $\mathcal{T}_i$ which passes through $\mathbf{p}_i$ and is perpendicular to $\mathbf{n}_i$.
(3) **Delaunay triangulation**: We carry out a 2D Delaunay triangulation [Delaunay et al. 1934] on $\mathcal{T}_i$, with particle i and $\mathcal{K}_i$'s projected coordinates as the input of the Delaunay algorithm. We use the Delaunator library https://github.com/delfrrr/delaunator-cpp for implementation.
(4) **Local mesh construction**: We build the local mesh $\mathcal{M}_i$ of $\mathbf{p}_i$ by taking its 1-ring mesh on $\mathcal{T}_i$ and replacing the vertex part with the corresponding particles' 3D position.

Triangle non-manifoldness. We note that the local mesh representation has non-manifold triangles across different particles. Ideally, on a manifold mesh, a triangle should belong to the local mesh of each of its three vertices. However, on a local mesh, this might not be the case. For instance, as shown in the inset figure, after separate Delaunay triangulations on the two particles A and B, B has ACB and ADB in its local mesh, while A's local mesh does not include ACB and ADB, but has ACD instead. In a traditional mesh-based simulator, such non-manifold triangles will cause problems because they introduce holes onto the surface discretization, which will be inevitably enlarged due to the -induced contraction forces on their rims.



Our position-based framework naturally avoids this issue because the role of each particle's local mesh is to establish a surface area constraint in the projection system, and no connectivity assumption is needed to enforce these constraints. As long as each particle's local mesh has a full ring of triangles and can properly reflect its local geometry (i.e., area and curvature), we do not need to check or ensure any connectivity manifoldness across particles. As such, reminiscent of the role of local differentiable operators in [Wang et al. 2020, 2021], the local mesh in our setting is used to approximate a local shape (in order to minimize the local surface area), rather than to construct a globally manifold mesh. Therefore, it can safely enjoy its parallel nature without worrying about any element manifoldness issues (see our validation in Section 6.1). In this sense,



(a) Top view (b) Front view

Fig. 4. Impaled droplet: a water drop impacting a hydrophobic cone with an aperture of 120°. The water drop is pierced and bursts into a string of pearls.

we regard our local mesh scheme to be naturally compliant with PBD for its surface representation.

4.3 Surface Tension Constraint

Based on each particle's local mesh, we devise two types of constraints, including the surface area constraint and the unilateral edge distance constraint, to model the surface area minimization process and meanwhile maintain a uniform distribution of particles on the evolving surface. This constraint-enabled tension mechanics further interacts with the incompressible constraints to produce surface tension effects. We discuss each constraint type as follows.

Area constraint. We build an area-minimization constraint to minimize the area sum of the triangles on each particle's local mesh. For particle i and its neighboring triangle set $T(i) = \{t_1, \cdots, t_{k_i}\}$, the area constraint C_i is defined as

$$C_i^A(\mathbf{p}) = \sum_{t \in T(i)} \frac{1}{2} \|(\mathbf{p}_{t^2} - \mathbf{p}_{t^1}) \times (\mathbf{p}_{t^3} - \mathbf{p}_{t^1})\|, \tag{7}$$

in which $t^{1,2,3}$ represents the three particle indices of triangle t. The gradients of the area constraint function with respect to the particle positions $\mathbf{p}_{t^1}$, $\mathbf{p}_{t^2}$ and $\mathbf{p}_{t^3}$ are calculated as

$$\begin{aligned}
\nabla_{t^1} C_t^A(\mathbf{p}) &= \frac{1}{2} \frac{(\mathbf{p}_{t^2} - \mathbf{p}_{t^1}) \times (\mathbf{p}_{t^3} - \mathbf{p}_{t^1})}{\|(\mathbf{p}_{t^2} - \mathbf{p}_{t^1}) \times (\mathbf{p}_{t^3} - \mathbf{p}_{t^1})\|} \times (\mathbf{p}_{t^3} - \mathbf{p}_{t^2}), \\
\nabla_{t^2} C_t^A(\mathbf{p}) &= \frac{1}{2} \frac{(\mathbf{p}_{t^3} - \mathbf{p}_{t^2}) \times (\mathbf{p}_{t^1} - \mathbf{p}_{t^2})}{\|(\mathbf{p}_{t^3} - \mathbf{p}_{t^2}) \times (\mathbf{p}_{t^1} - \mathbf{p}_{t^2})\|} \times (\mathbf{p}_{t^1} - \mathbf{p}_{t^3}), \\
\nabla_{t^3} C_t^A(\mathbf{p}) &= \frac{1}{2} \frac{(\mathbf{p}_{t^1} - \mathbf{p}_{t^3}) \times (\mathbf{p}_{t^2} - \mathbf{p}_{t^3})}{\|(\mathbf{p}_{t^1} - \mathbf{p}_{t^3}) \times (\mathbf{p}_{t^2} - \mathbf{p}_{t^3})\|} \times (\mathbf{p}_{t^2} - \mathbf{p}_{t^1}).
\end{aligned} \tag{8}$$

Distance constraint. The surface tension force tends to minimize the mesh area around each particle. During this process, particles may gather and redistribute unevenly on the surface. We devise an additional unilateral distance constraint to solve the particles' uneven distribution problem. The constraint is defined as

$$C_{ij}^D(\mathbf{p}) = \min\{0, \|\mathbf{p}_i - \mathbf{p}_j\| - d_0\}, \tag{9}$$

in which $\mathbf{p}_i$ and $\mathbf{p}_j$ are neighboring particles with identical type labels (surface or inner), and d_0 is set to be 2 times of particle radius.

The gradients of the constraint function with respect to particle positions $\mathbf{p}_i$ and $\mathbf{p}_j$ are

$$\begin{aligned}
\nabla_i C_{ij}^D(\mathbf{p}) &= \begin{cases} 0, & \|\mathbf{p}_i - \mathbf{p}_j\| > d_0, \\ \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|}, & \text{others.} \end{cases} \\
\nabla_j C_{ij}^D(\mathbf{p}) &= \begin{cases} 0, & \|\mathbf{p}_i - \mathbf{p}_j\| > d_0, \\ \frac{\mathbf{p}_j - \mathbf{p}_i}{\|\mathbf{p}_i - \mathbf{p}_j\|}, & \text{others.} \end{cases}
\end{aligned} \tag{10}$$

We note that the distance constraint can be applied to both surface particles and volume particles without causing any jittering artifacts. In our experiments, we observe our distance constraints achieve similar effects as the artificial repulsive force in PBD fluid when they are applied to volume particles.

4.4 Time Integration

Last, we integrate our surface particle detection, local mesh construction, and surface tension constraints into a traditional position-based fluid framework. The entire algorithm is demonstrated in 1 (the new steps are in violet). We rebuild our constraints for each resetIter iteration for potential topology changes during the constraint projection. We use resetIter = 10 for all our simulations.

Algorithm 1 Simulation Loop

```
for each particle i do v_i ← v_i + (Δt/m) f_i^ext
for each particle i do p_i^* ← p_i + Δt v_i
for k ← 0, ··· , nIter do
    if k%resetIter = 0 then
        S ← DetectSurfaceParticles(p^*)
        M ← BuildLocalMesh(p^*, S)
        C^ρ, C^A, C^D ← BuildConstraints(p^*, M)
    p^* ← ProjectConstraints(p^*, C^ρ, C^A, C^D)
for each particle i do v_i ← (p_i^* − p_i)/Δt
v ← ApplyViscosity(v, p^*)
for each particle i do p_i ← p_i + Δt v_i
```

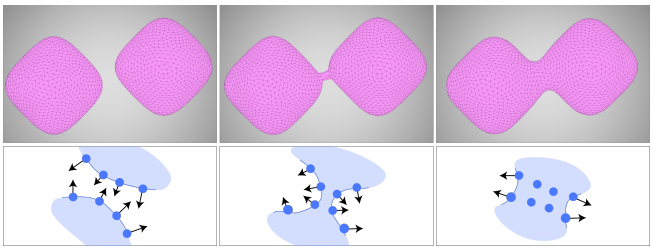


Fig. 5. The merge of two square droplets. The blue dots represent fluid particle, and the black arrows are the surface normal at that point.

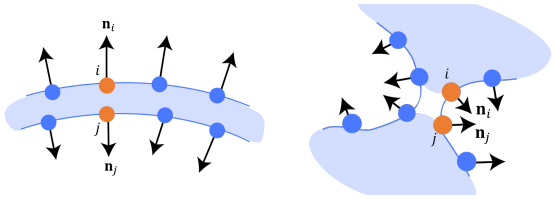
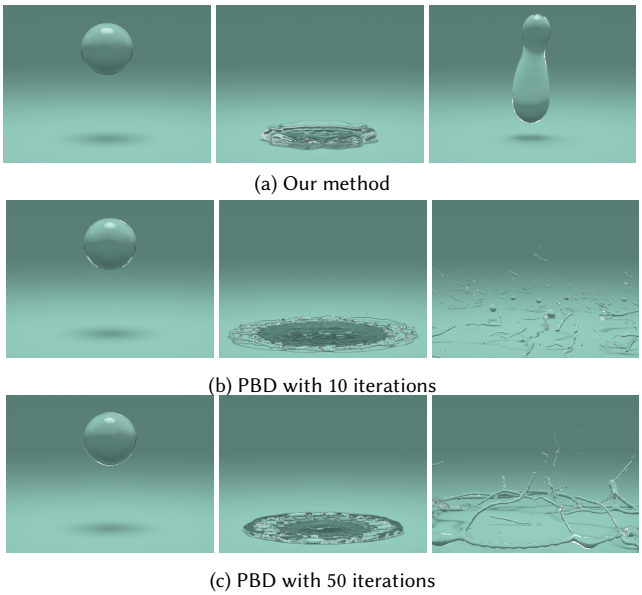


Fig. 6. Two typical configurations when nearby particles i and j possess obviously different normal directions. (a) particles are on the different sides of the membrane; (b) particles are on two approaching surfaces.

5 TOPOLOGICAL/CODIMENSIONAL TRANSITIONS

Topological transitions. Surface tension flow manifests complicated topological changes, ranging from fluid volume merging and splitting to sliding on a solid surface. Handling these topological changes remains substantially challenging for a mesh-based method (e.g., [Da et al. 2016; Huang and Michels 2020; Yu et al. 2012], which requires implementing an ensemble of meshing operations in a highly dynamic setting. Our method does not rely on local meshing operations to track topological transitions. Instead, topological evolution, such as fluid volume merging and splitting, can happen naturally with particles. In each time step, we detect surface particles using the algorithm in Section 4.1 and build a local mesh for each particle, which we found can sufficiently reconstruct the local topological features. For example, as shown in Figure 5, when two fluid volumes get close, the local mesh algorithm can automatically distinguish the topology before and after a collision. The splitting process is the inverse of the merging process shown in Figure 5 and the topological change is also processed automatically, as the local mesh connectivities between two particles will vanish automatically when one particle moves out from the neighbor range of the other.

Codimensional transitions. Our framework handles two types of codimensional transitions, between fluid volumes and films, and between fluid volumes and filaments. First, we handle the codimensional transition from a volume to a film with an additional normal check. We represent a thin film using two layers of surface particles. The side of each particle is identified by its normal vector based on the color field (see Equation 6). For the situation shown in Figure 6a, it is sufficient to remove particle j from particle i's neighbors based on the large discrepancy of their normal directions (i.e. $\arccos(\mathbf{n}_i \cdot \mathbf{n}_j) > \theta$ holds, where $\theta = \frac{\pi}{4}$). However, nearby particles with obviously different normal directions may also come from two approaching volumes, as shown in Figure 6b. In order to allow them to merge, in practice, we only remove particle j from i's neighbors if $\arccos(\mathbf{n}_i \cdot \mathbf{n}_j) > \theta$ and $(\mathbf{n}_i - \mathbf{n}_j) \cdot (\mathbf{p}_i - \mathbf{p}_j) > 0$ hold simultaneously.



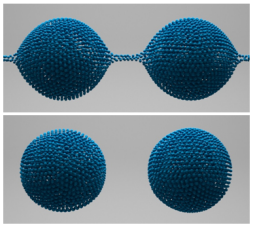
(a) Our method

(b) PBD with 10 iterations

(c) PBD with 50 iterations

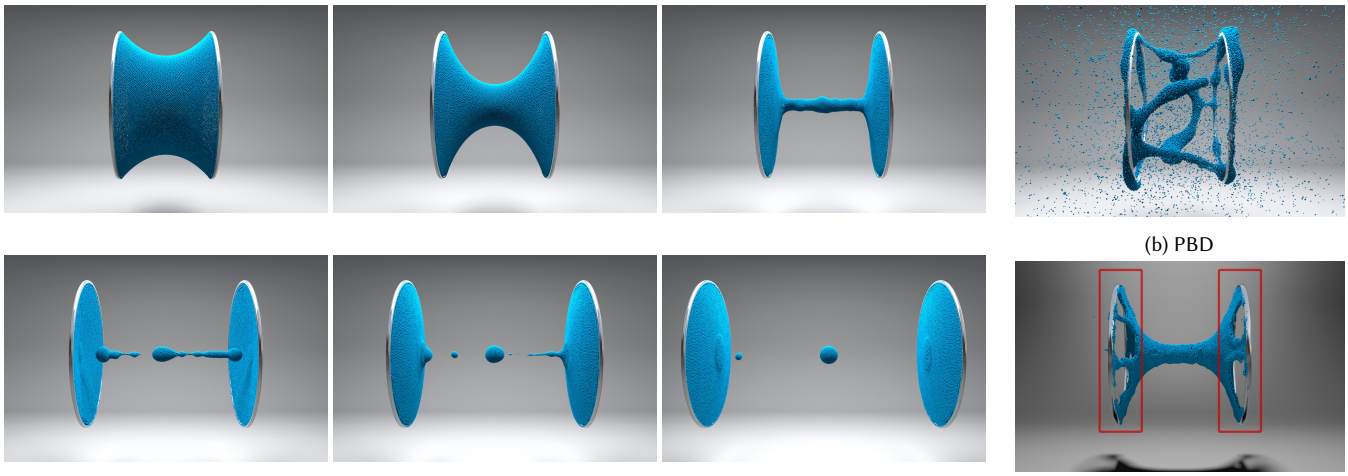
Fig. 7. Droplet hitting the ground: a free-falling droplet can bounce off the ground after squeezing into a thin disk. From top to bottom are results from: our method, PBD fluids with 10 iterations and 50 iterations, respectively.

The codimensional transition from a film to a volume can naturally happen when the film gets thickened with incoming volumes (see examples in Figure 11). Second, we handle the transitions between fluid volumes and filaments. A fluid volume can transition to a filament in a way similar to the volume-film transition. For filament-to-volume transition, we lower the density constraint to avoid any particles being "locked" on the filament due to the complicated interactions among density, distance, and area constraints. Specifically, we detect the filament particles by applying Principal Component Analysis (PCA) to the neighborhood particle positions (see Supplementary 2) and turn off their density constraints. The inset figure compares the simulations with the density constraints turned on, which exhibit non-contractable filament artifacts, and turned off, in which fluid filaments naturally become droplets under surface tension. We employ a simple mechanism for the isolated particles: a particle becomes an isolated droplet when it splits from the bulk; an isolated particle merges into the bulk if they are close.



6 RESULTS

In this section, we first describe experiments that help to validate the correctness of our algorithm. Then, we demonstrate the efficacy



(a) Our method (b) PBD (c) Our method W/O C^D

Fig. 8. Film catenoid: A fluid membrane is attached to two parallel circular rings. When the rings separate from each other, under surface tension, the membrane will contract in the center to minimize its surface, and finally split into two membranes and splash out some small droplets in the middle after the ring separation surpasses the Laplace limit. (a) simulation result using our method, which can reproduce the entire process; (b) simulation snapshot using traditional position-based fluid solver, which failed right away; (c) simulation snapshot using our method without distance constraint, in which particles clustered and non-physical holes on the membrane can be observed.

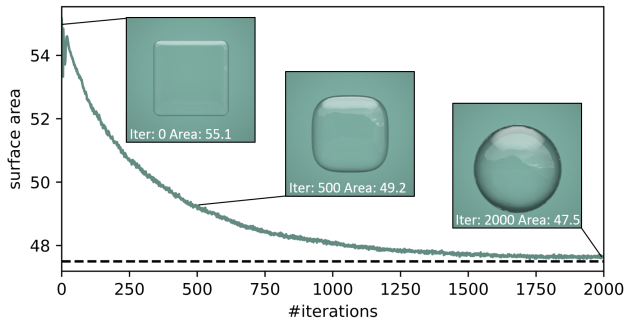


Fig. 9. Parameter study of iteration number. As the iteration number increase, the area constraint converges smoothly and saturates when the cubic droplet turns into a sphere.

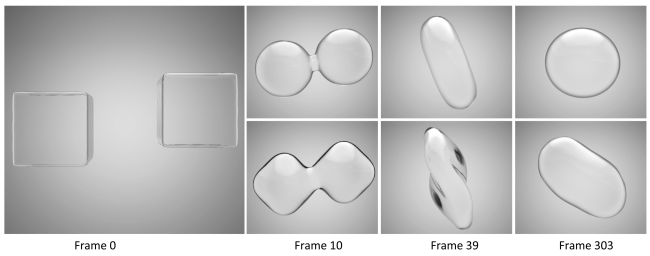


Fig. 10. Cubic droplets colliding under different surface tension intensities. Top and bottom show simulation snapshots at the same time instance with $\varepsilon^A = 5.0$ and $\varepsilon^A = 1.0$ respectively.

of our method through an array of examples dominated by strong surface tension.

6.1 Validation

Comparison to PBD fluid. We first compare our approach to traditional PBD fluid [Macklin et al. 2014] in two examples to showcase the efficacy of our surface tension constraints and local mesh representation. When a droplet falls onto a solid surface, the liquid will be flattened to a thin film and then pulled back by its surface tension. Our method successfully reproduces this phenomenon (see Figure 7a). However, PBD simulations with strong (50 iterations per frame, Figure 7b) and weak (10 iterations per frame, Figure 7c) surface tension both blow up at the moment when the droplet squeezes into a thin disk. In the film catenoid example, a fluid membrane is initially attached to two solid rings moving apart from each other. The membrane contracts and splits into two, with small droplets pinched off in the middle. Our method can reproduce the entire process (see Figure 8). In comparison, the traditional position-based fluid solver fails right away due to the incorrect surface tension model on a thin film.

Ablation study on distance constraints. We conduct an ablation study of the distance constraints in the film catenoid example as shown in Figure 8. We compare the simulations with and without the distance constraints under the same parameter setting. The results show that the distance constraints can distribute the particles uniformly on a dynamic surface, yet the simulation with only density constraints will manifest non-physical holes on the membrane due to the particle clustering.

Parameter study of ε^A. We show a parameter study of surface tension effects by tuning the value of ε^A of area constraints. Figure 10 shows the simulation results of two cubic droplets colliding with $\varepsilon^A = 5.0$ and 1.0 respectively. Large surface tension (top row) turns the cubic droplets into a sphere quickly, while the area minimization process is much slower for small surface tension (bottom row).

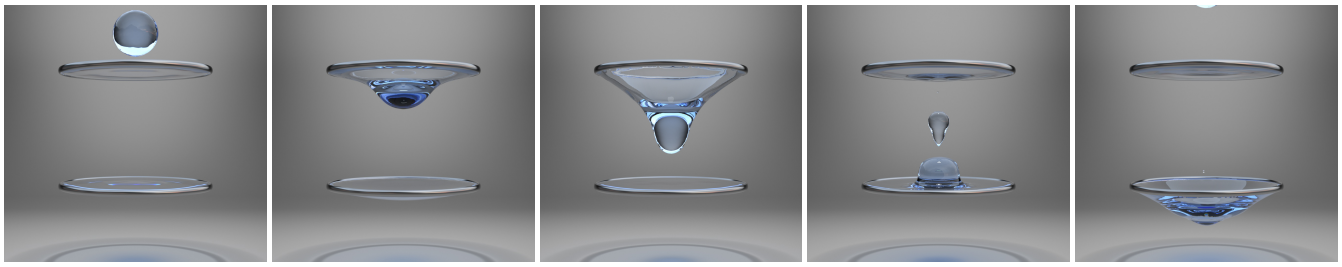


Fig. 11. Liquid membrane filter: relies on surface tension force, free-falling droplets carrying big enough momentum can pass through the first membrane, but are trapped by the second membrane after it is slowed down.

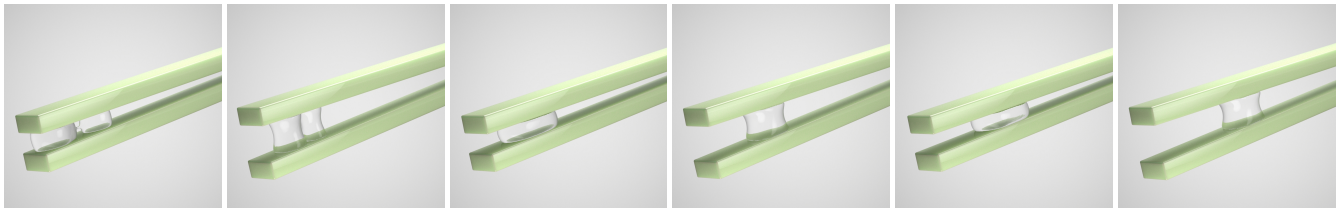


Fig. 12. Tweezers for droplets: the periodic open and close motion of the chopstick "tweezers" can drive the trapped droplets to move toward the narrow end.

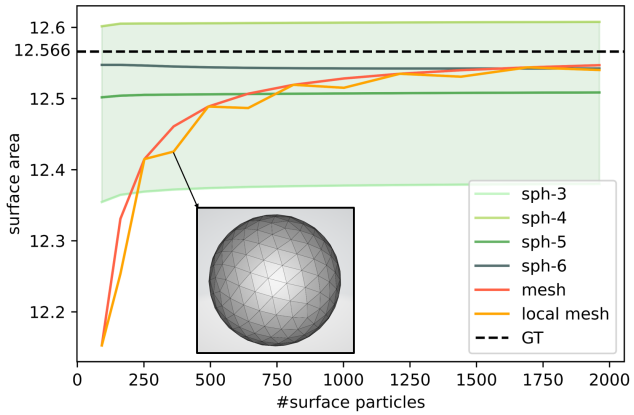
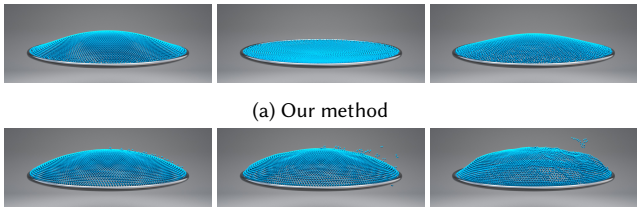


Fig. 13. Area calculation on a uniformly sampled sphere with different methods. With different kernel sizes, both our local mesh and the manifold mesh converge to the true solution, while the SPH method could not converge.



(a) Our method

(b) [Akinci et al. 2013]'s method

Fig. 14. A fluid film with its rim fixed on a ring oscillates with zero gravity. Our constraint-based formulation (a) can deliver more stable simulation results than the explicit force-based method in [Akinci et al. 2013] (b).

Parameter study of iteration number. We study the influence of the number of iterations on enforcing the surface tension constraints. We set up a quasistatic simulation to turn a cubic droplet into a sphere within a single time step. The convergence plot in Figure 9 shows that the droplet's surface area stably converges to its minimum as the droplet's shape turns from a cube to a sphere. We visualize three snapshots of fluid shapes during the iterations.

Non-manifold triangle statistics. We measure the triangles' non-manifoldness in the particles' local meshes by counting the duplication of each triangle. Figure 16 plots the ratio between the number of non-manifold triangles over the number of all triangles in the droplet collision example. Except for the initialization in the first few frames, the ratio is consistently below 2.5%, and for most of the frames, this ratio is below 0.5%. At the same time, we visualize the non-manifold triangles with a different color, as shown in the two screenshots in Figure 16. We can observe that these triangles are randomly distributed over the entire surface. No artifacts are observed in the simulation around these non-manifold triangles.

Comparison to explicit force based methods. It is always possible to add a force term explicitly to a PBD system, reminiscent of how gravity is treated. To justify this, we implemented the SPH force in [Akinci et al. 2013] in a PBD framework. As shown in Figure 17, we observed that an explicit force model does not deliver a robust simulation under large surface tension. The same SPH force model does not work for thin-film simulation, as shown in Figure 14.

Comparison on surface-area calculation. We performed a simple test for surface area calculation using the local mesh, manifold mesh, and the SPH method (using the codimension-1 cubic spline kernel). We sample points on a unit sphere. As shown in Figure 13, as the particle number increases, both local and manifold meshes converge to the ground truth. However, the SPH method manifests a large variation with different kernel sizes ($3-6$ times the particle radius).

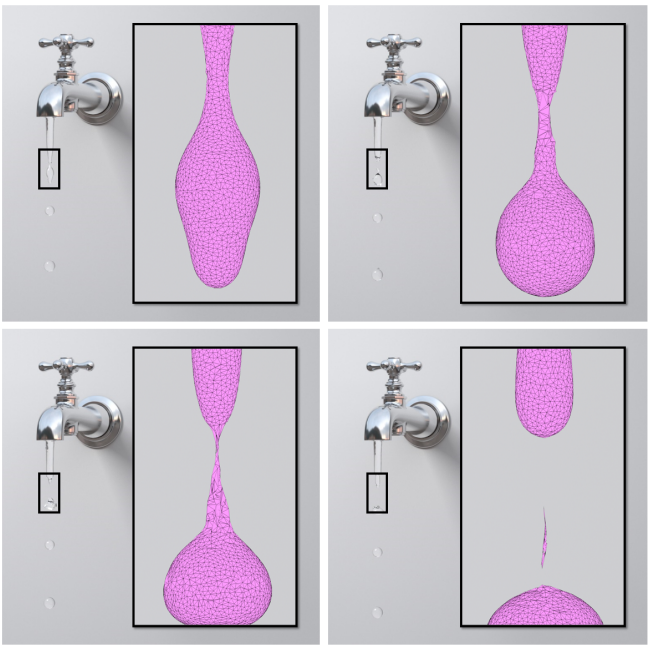


Fig. 15. Faucet dripping: image sequence of a drop of water dripping from a faucet, including the formation of the liquid neck due to surface tension, and its subsequent thinning until breakup. The pink meshes in each inset figure are constructed using our proposed method.

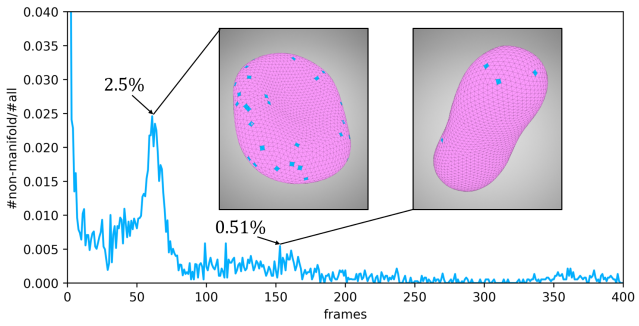
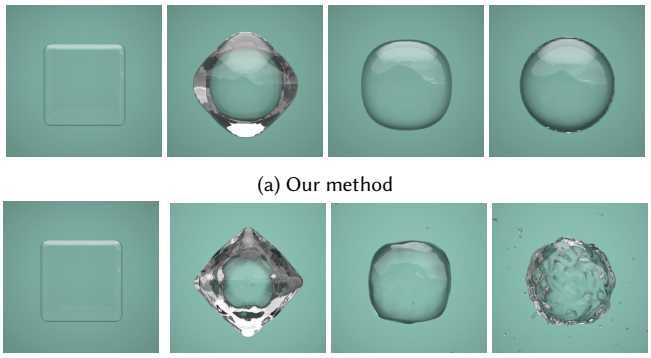


Fig. 16. Statistics of non-manifold ratio for droplets colliding example. In the inset figures, triangles with pink color are consistent ones, while triangles with blue color are non-manifold ones.

6.2 Examples

Impaled droplet. Figure 4 shows a droplet impaling on a hydrophobic solid cone [Durey et al. 2020] with an aperture of 120°. The water drop is pierced by the tip of the cone and repelled into a ring shape by the slope. Then, the liquid ring expands and breaks into smaller drops under surface tension. This example demonstrates a series of topological changes of the droplets under surface tension.

Dripping faucet. In the dripping faucet example, as shown in Figure 15, fluid flows out from a faucet and forms a neck due to the surface tension force, and then it pinches off into a series of small droplets. This phenomenon is also called Rayleigh-Plateau instability [Plateau 1873]. We also visualize the local mesh in the



(a) Our method

(b) PBD with explicit surface tension force in [Akinci et al. 2013]

Fig. 17. A cubic droplet gradually turns into a sphere driven by surface tension force. (a) simulation result using our method, which can reproduce the entire process; (b) simulation snapshots using traditional PBD fluid solver, with surface tension treated as an explicit force term using the model in [Akinci et al. 2013].

animation to show its evolution. In addition, we demonstrate the comparison with the water drips from a pipe with different surface tension coefficients (see the supplementary video).

Drop rail. We simulate two examples with water droplets interacting with hydrophilic chopsticks. The adhesion between droplets and solids is modeled as adhesion constraints [Bender et al. 2017] in all our examples. In Figure 19, water droplets roll off a pair of chopsticks under the influence of gravity, surface tension, and adhesion forces. We can observe rich topological changes in this simulation, including droplets merging, pinching off, wetting, and dribbling.

Liquid tweezer. We show a liquid tweezer in Figure 12. The water droplets are sandwiched in a chopstick 'tweezer'. As the tweezer opens and closes periodically, the interaction between surface tension and adhesion force drives the droplets to move towards to the narrow end of the chopstick. This simulation reproduced a simplified version of a real bird drinking water with its long beaks, motivated by [Prakash et al. 2008].

Double-membrane filter. The liquid membrane filter relies on surface tension force for object separation. Only fluid volumes with a strong enough momentum can pass through the membrane. As shown in Figure 11, the droplet gains enough momentum during free falling and successfully passes through the first membrane. However, due to this interaction, the droplet is caught by the second membrane. Complicated topological and codimension transitions happen during this process, including the droplet merging into and pinching off from a wet liquid film.

Surface tension splash. We reproduce the classic bunny example in [Macklin et al. 2014] enhanced with high surface tension effects. As shown in Figure 18, the water from the breaking dam flushes onto a bunny, creating high splashes and pinched-off drops due to strong surface tension.

Teapot effect. We simulate the teapot effect [Duez et al. 2010] to showcase our algorithm's ability to handle solid-fluid interactions. The teapot effect shows the tendency for liquid to dribble down

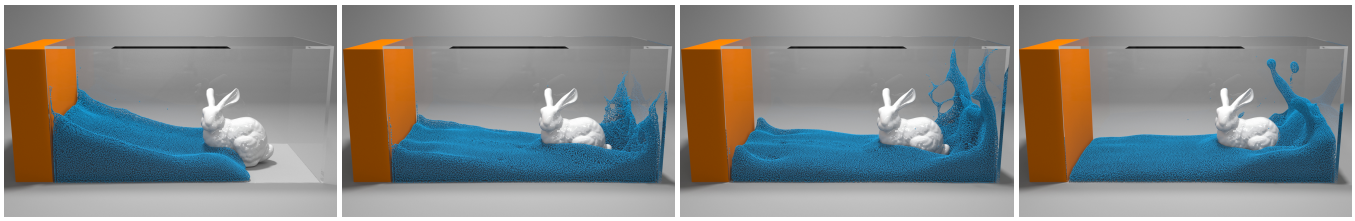


Fig. 18. Bunny splash: the water from the breaking dam flushes onto a rabbit, creating a high splash and spherical drops.

Fig. 19. Water droplets roll off chopsticks under the influence of gravity, surface tension, and adhesion force and exhibit rich dynamic effects, such as merging, pinching off, wetting, and dribbling.

Table 1. Simulation parameters for the examples.

Example[§]	# Particles	ε^A[†]	g	# Iterations	Time/Frame			
					Surface Detection	Local Mesh	Project Constraints	Total (s)
Droplets Colliding	9262	5&1	0	100	7.1%	5.3%	64.7%	1.7
Bouncing Droplet	11494	1	0	100	6.7%	5.4%	70.8%	2.4
Impaled droplet	47369	0.8	0	15	8.3%	6.1%	50.0%	1.8
Faucet dripping	28336	0.21	1	100	6.5%	5.4%	66.7%	7.8
Drop rail	4280	5	1	100	4.7%	6.5%	64.5%	1.1
Liquid tweezer	14346	0.1	0	100	5.2%	3.2%	80.0%	6.0
Double-membrane filter	23957	1.0	0.5	40	14%	25%	10.2%	5.2
Film catenoid	53620	0.08	0	100	7.6%	12.6%	60.7%	7.9
Surface tension splash	441143	1	9.8	10	14%	5%	31.2%	6.4
Teapot effect	121308	3	4	20	3.7(s)	1.8(s)	3.0(s)	21.3[‡]

[§] The time step size is 1/30s for all examples.
[†] ε^A is the CFM of surface constraints, we use $\varepsilon^D = 40$ for distance constraints, $\varepsilon^\rho = 180$ for density constraints in surface tension splash and teapot effect, $\varepsilon^\rho = 600$ for the rest of examples.
[‡] Most of the time is consumed by detecting whether two neighboring particles are inside/outside the teapot at the same time by computing the closest normal on the mesh, which prevents artifacts caused by the thin wall of the teapot.

the outside of the spout when pouring with a low flow rate. This phenomenon is driven by the interaction between adhesion and surface tension force. In the simulation shown in Figure 2, a teapot filled with water is gradually tilted. As the teapot angle gets inclined, the flow rate increases, the flow adhesion effects are weakened, and splashes start to flow.

6.3 Performance

We parallelize most of the steps in our algorithm using OpenMP [Dagum and Menon 1998] on the CPU. All experiments were performed on an Intel(R) Xeon(R) 28-core processor. The parameters and timing statistics are summarized in Table 1. From Table 1, we can see that the main bottleneck still comes from the constraint projection step as conventional PBD. For bulk liquid simulations, our method only adds a small computation overhead to the origin PBD fluid solver. Moreover, most of the extra computational loads are concentrated on the surface particle detection step, especially for large-scale volumetric examples. For thin film examples, the overall extra computational cost is reasonably low. Since the portion of surface particles is much higher in these scenarios, the surface construction step becomes the major time consumer. The total time can be further accelerated if we implement our algorithm on GPU.

7 CONCLUSION

We proposed a novel PBD method to simulate surface tension flow. Our approach enhanced the traditional PBD framework with a local mesh representation, which effectively characterizes the local geometric features and enables accurate surface tension calculation. This geometric representation, as well as its constraint formula, are parallelizable and easy to integrate into a standard PBD fluid

framework. We demonstrate the efficacy of our approach by simulating several surface tension flow examples, such as liquid tweezer, membrane filter, and teapot effects, which were all impractical for a traditional PBD method.

8 LIMITATION AND FUTURE WORK

We identify three limitations of our method. First, our local mesh scheme relies on accurate surface particle detection. If an interior particle is incorrectly detected as a surface particle, it will contribute surface tension to the system in a non-physical way. Therefore, a more versatile local mesh scheme that is not sensitive to particle distribution can improve the method's geometric robustness. Second, though our approach can significantly improve the surface tension effects of the PBD fluid, it does not provide a physically accurate model to simulate these interfacial phenomena from the first principles. The ε values used in our example do not correspond to the surface tension coefficients in real-world settings. Therefore, its current application scope is within fluid animation. Third, we insert the new constraints into the default PBD projection scheme without investigating any acceleration strategies. We anticipate that well-designed preconditioners and optimization schemes can lead to better convergence rates and performance, which will benefit interactive or even real-time simulations.

For future work, we plan to continue our exploration in these three directions to explore more robust local mesh schemes, more accurate physical models, and effective preconditioners for the PBD projection scheme. In particular, we are interested in extending the current local mesh method to enhance other particle-based fluid solvers such as SPH and PIC/FLIP. Incorporating more complex physical models such as viscoelasticity and magnetic-mechanical coupling is another interesting direction to explore.

ACKNOWLEDGMENTS

Jingrui Xing, Liangwang Ruan, and Baoquan Chen acknowledge the funding support from China-Israel (NSFC-ISF) International Cooperation and Exchanges (62161146002). Bo Zhu acknowledges Burke Research Initiation Award and Toyota Central R&D Labs. We credit the Houdini Education licenses for the video generations.

REFERENCES

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

Iván Alduán, Angel Tena, and Miguel A. Otaduy. 2017. DYVERSO: A Versatile Multi-Phase Position-Based Fluids Solution for VFX. *Computer Graphics Forum* 36, 8 (2017), 32–44. https://doi.org/10.1111/cgf.12992

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

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

Mikhail Belkin, Jian Sun, and Yusu Wang. 2009. *Constructing Laplace Operator from Point Clouds in Rd.* Society for Industrial and Applied Mathematics, USA, 1031–1040.

Jan Bender, Dan Koschier, Patrick Charrier, and Daniel Weber. 2014. Position-based simulation of continuous materials. *Computers & Graphics* 44 (2014), 1–10. https://doi.org/10.1016/j.cag.2014.07.004

Jan Bender, Matthias Müller, and Miles Macklin. 2017. A Survey on Position Based Dynamics, 2017. In *Proceedings of the European Association for Computer Graphics: Tutorials* (Lyon, France) *(EG '17)*. Eurographics Association, Goslar, DEU, Article 6, 31 pages. https://doi.org/10.2312/egt.20171034

Kenneth Bodin, Claude Lacoursière, and Martin Servin. 2012. Constraint Fluids. *IEEE Transactions on Visualization and Computer Graphics* 18 (2012), 516–526.

Tyson Brochu, Christopher Batty, and Robert Bridson. 2010. Matching Fluid Simulation Elements to Surface Geometry and Topology. *ACM Trans. Graph.* 29, 4, Article 47 (jul 2010), 9 pages. https://doi.org/10.1145/1778765.1778784

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 Trans. Graph.* 40, 4, Article 111 (jul 2021), 16 pages. https://doi.org/10.1145/3450626.3459874

Pascal Clausen, Martin Wicke, Jonathan R. Shewchuk, and James F. O'Brien. 2013. Simulating Liquids and Solid-Liquid Interactions with Lagrangian Meshes. *ACM Trans. Graph.* 32, 2, Article 17 (apr 2013), 15 pages. https://doi.org/10.1145/2451236.2451243

Simon Clavet, Philippe Beaudoin, and Pierre Poulin. 2005. Particle-Based Viscoelastic Fluid Simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '05)*. Association for Computing Machinery, New York, NY, USA, 219–228. https://doi.org/10.1145/1073368.1073400

Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double Bubbles sans Toil and Trouble: Discrete Circulation-Preserving Vortex Sheets for Soap Films and Foams. *ACM Trans. Graph.* 34, 4, Article 149 (jul 2015), 9 pages. https://doi.org/10.1145/2767003

Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-Only Liquids. *ACM Trans. Graph.* 35, 4, Article 78 (jul 2016), 12 pages. https://doi.org/10.1145/2897824.2925899

Leonardo Dagum and Ramesh Menon. 1998. OpenMP: an industry standard API for shared-memory programming. *IEEE computational science and engineering* 5, 1 (1998), 46–55.

Boris Delaunay et al. 1934. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7, 793-800 (1934), 1–2.

R. Diziol, J. Bender, and D. Bayer. 2011. Robust Real-Time Deformation of Incompressible Surface Meshes. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Vancouver, British Columbia, Canada) *(SCA '11)*. Association for Computing Machinery, New York, NY, USA, 237–246. https://doi.org/10.1145/2019406.2019438

Cyril Duez, Christophe Ybert, Christophe Clanet, and Lydéric Bocquet. 2010. Wetting Controls Separation of Inertial Flows from Solid Surfaces. *Physical Review Letters* 104, 8 (feb 2010). https://doi.org/10.1103/physrevlett.104.084503

Guillaume Durey, Quentin Magdelaine, Mathias Casiulis, Hoon Kwon, Julien Mazet, Pierre Chantelot, Anaïs Gauthier, Christophe Clanet, and David Quéré. 2020. Droplets impaling on a cone. *Phys. Rev. Fluids* 5 (Nov 2020), 110507. Issue 11. https://doi.org/10.1103/PhysRevFluids.5.110507

Nick Foster and Ronald Fedkiw. 2001. Practical Animation of Liquids. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 23–30. https://doi.org/10.1145/383259.383261

M. Frâncu and F. Moldoveanu. 2017. Unified Simulation of Rigid and Flexible Bodies Using Position Based Dynamics. In *Proceedings of the 13th Workshop on Virtual Reality Interactions and Physical Simulations* (Lyon, France) *(VRIPHYS '17)*. Eurographics Association, Goslar, DEU, 49–58. https://doi.org/10.2312/vriphys.20171083

C.W Hirt and B.D Nichols. 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* 39, 1 (1981), 201–225. https://doi.org/10.1016/0021-9991(81)90145-5

Libo Huang and Dominik L. Michels. 2020. Surface-Only Ferrofluids. *ACM Trans. Graph.* 39, 6, Article 174 (nov 2020), 17 pages. https://doi.org/10.1145/3414685.3417799

Libo Huang, Ziyin Qu, Xun Tan, Xinxin Zhang, Dominik L. Michels, and Chenfanfu Jiang. 2021. Ships, Splashes, and Waves on a Vast Ocean. *ACM Trans. Graph.* 40, 6, Article 203 (dec 2021), 15 pages. https://doi.org/10.1145/3478513.3480495

Markus Huber, Stefan Reinhardt, Daniel Weiskopf, and Bernhard Eberhardt. 2015. Evaluation of Surface Tension Models for SPH-Based Fluid Animations Using a Benchmark Test. In *Workshop on Virtual Reality Interaction and Physical Simulation*, Fabrice Jaillet, Florence Zara, and Gabriel Zachmann (Eds.). The Eurographics Association, USA, 41–50. 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 Trans. Graph.* 39, 6, Article 183 (nov 2020), 13 pages. https://doi.org/10.1145/3414685.3417845

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

Sadashige Ishida, Peter Synak, Fumiya Narita, Toshiya Hachisuka, and Chris Wojtan. 2020. A Model for Soap Film Dynamics with Evolving Thickness. *ACM Trans. Graph.* 39, 4, Article 31 (jul 2020), 11 pages. 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 Trans. Graph.* 36, 6, Article 199 (nov 2017), 11 pages. https://doi.org/10.1145/3130800.3130835

Ki-Hoon Kim, Jung Lee, Chang-Hun Kim, and Jong-Hyun Kim. 2021. Particle-Based Dynamic Water Drops with High Surface Tension in Real Time. *Symmetry* 13, 7 (2021), 1265. https://doi.org/10.3390/sym13071265

Tae-Yong Kim, Nuttapong Chentanez, and Matthias Müller-Fischer. 2012. Long Range Attachments - a Method to Simulate Inextensible Clothing in Computer Games. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Lausanne, Switzerland) *(SCA '12)*. Eurographics Association, Goslar, DEU, 305–310.

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2019. Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids. In *Eurographics 2019 - Tutorials*. The Eurographics Association;, New York, NY, USA, 1–41. https://doi.org/10.2312/egt.20191035

S. Koshizuka and Y. Oka. 1996. Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid. *Nuclear Science and Engineering* 123, 3 (1996), 421–434. https://doi.org/10.13182/NSE96-A24205 arXiv:https://doi.org/10.13182/NSE96-A24205

Rongjie Lai, Jian Liang, and Hongkai Zhao. 2013. A local mesh method for solving PDEs on point clouds. *Inverse Problems and Imaging* 7 (2013), 737–755.

Miles Macklin and Matthias Müller. 2013. Position based fluids. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. 2014. Unified Particle Physics for Real-Time Applications. *ACM Trans. Graph.* 33, 4, Article 153 (jul 2014), 12 pages. https://doi.org/10.1145/2601097.2601152

Takuya Matsunaga, Seiichi Koshizuka, Tomoyuki Hosaka, and Eiji Ishii. 2020. Moving surface mesh-incorporated particle method for numerical simulation of a liquid droplet. *J. Comput. Phys.* 409 (2020), 109349. https://doi.org/10.1016/j.jcp.2020.109349

Joseph Morris. 2000. Simulating surface tension with Smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids* 33 (06 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 *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03)*. Eurographics Association, Goslar, DEU, 154–159.

Matthias Müller, Nuttapong Chentanez, Tae-Yong Kim, and Miles Macklin. 2015. Air Meshes for Robust Collision Handling. *ACM Trans. Graph.* 34, 4, Article 133 (jul 2015), 9 pages. https://doi.org/10.1145/2766907

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and J. W. Ratcliff. 2007. Position based dynamics. *J. Vis. Commun. Image Represent.* 18 (2007), 109–118.

Matthias Müller, Barbara Solenthaler, Richard Keiser, and Markus Gross. 2005. Particle-Based Fluid-Fluid Interaction. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '05)*. Association for Computing Machinery, New York, NY, USA, 237–244. https://doi.org/10.1145/1073368.1073402

Matthias Müller, Tae-Yong Kim, and Nuttapong Chentanez. 2012. Fast Simulation of Inextensible Hair and Fur. In *Workshop on Virtual Reality Interaction and Physical Simulation*, Jan Bender, Arjan Kuijper, Dieter W. Fellner, and Eric Guerin (Eds.). The Eurographics Association, USA, 39–44. https://doi.org/10.2312/PE/vriphys/vriphys12/039-044

Stanley Osher and James A Sethian. 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* 79, 1 (1988), 12–49. https://doi.org/10.1016/0021-9991(88)90002-2

Joseph Antoine Ferdinand Plateau. 1873. *Statique expérimentale et théorique des liquides soumis aux seules forces moléculaires*. Vol. 2. Gauthier-Villars.

Manu Prakash, David Quéré, and John WM Bush. 2008. Surface tension transport of prey by feeding shorebirds: the capillary ratchet. *science* 320, 5878 (2008), 931–934.

Alec R. Rivers and Doug L. James. 2007. FastLSM: Fast Lattice Shape Matching for Robust Real-Time Deformation. *ACM Trans. Graph.* 26, 3 (jul 2007), 82–es. https://doi.org/10.1145/1276377.1276480

Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. 2008. Two-Way Coupling of Fluids to Rigid and Deformable Solids and Shells. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–9. https://doi.org/10.1145/1360612.1360645

Liangwang Ruan, Jinyuan Liu, Bo Zhu, Shinjiro Sueda, Bin Wang, and Baoquan Chen. 2021. Solid-Fluid Interaction with Surface-Tension-Dominant Contact. *ACM Trans. Graph.* 40, 4, Article 120 (jul 2021), 12 pages. https://doi.org/10.1145/3450626.3459862

Nadine Abu Rumman, Prapanch Nair, Patric Müller, Loic Barthe, and David Vanderhaeghe. 2020. ISPH–PBD: Coupled Simulation of Incompressible Fluids and Deformable Bodies. *Vis. Comput.* 36, 5 (may 2020), 893–910. https://doi.org/10.1007/s00371-019-01700-y

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 (2012), 2092–2115. https://doi.org/10.1016/j.jcp.2011.11.021

James A Sethian and Peter Smereka. 2003. Level set methods for fluid interfaces. *Annual review of fluid mechanics* 35, 1 (2003), 341–372.

Kazuya Shibata, Issei Masaie, Masahiro Kondo, Kohei Murotani, and Seiichi Koshizuka. 2015. Improved pressure calculation for the moving particle semi-implicit method. *Computational particle mechanics* 2, 1 (2015), 91–108.

Russell Smith. 2004. Open dynamics engine v0. 5 user guide. *http://ode. org/* (2004).

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

Jos Stam. 1999. Stable Fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 121–128. https://doi.org/10.1145/311535.311548

Jos Stam and Eugene Fiume. 1995. Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. Association for Computing Machinery, New York, NY, USA, 129–136. https://doi.org/10.1145/218380.218430

Alexandre Tartakovsky and Paul Meakin. 2005. Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Physical review. E, Statistical, nonlinear, and soft matter physics* 72 (09 2005), 026301. https://doi.org/10.1103/PhysRevE.72.026301

Nils Thürey, Chris Wojtan, Markus Gross, and Greg Turk. 2010. A Multiscale Approach to Mesh-Based Surface Tension Flows. *ACM Trans. Graph.* 29, 4, Article 48 (jul 2010), 10 pages. https://doi.org/10.1145/1778765.1778785

Nobuyuki Umetani, Ryan Schmidt, and Jos Stam. 2015. Position-Based Elastic Rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Copenhagen, Denmark) *(SCA '14)*. Eurographics Association, Goslar, DEU, 21–30.

Hui Wang, Yongxu Jin, Anqi Luo, Xubo Yang, and Bo Zhu. 2020. Codimensional Surface Tension Flow Using Moving-Least-Squares Particles. *ACM Trans. Graph.* 39, 4, Article 42 (jul 2020), 16 pages. https://doi.org/10.1145/3386569.3392487

Huamin Wang, Peter J. Mucha, and Greg Turk. 2005. Water Drops on Surfaces. *ACM Trans. Graph.* 24, 3 (jul 2005), 921–929. https://doi.org/10.1145/1073204.1073284

Mengdi Wang, Yitong Deng, Xiangxin Kong, Aditya H. Prasad, Shiying Xiong, and Bo Zhu. 2021. Thin-Film Smoothed Particle Hydrodynamics Fluid. *ACM Trans. Graph.* 40, 4, Article 110 (jul 2021), 16 pages. https://doi.org/10.1145/3450626.3459864

Chris Wojtan, Matthias Müller-Fischer, and Tyson Brochu. 2011. Liquid Simulation with Mesh-Based Surface Tracking. In *ACM SIGGRAPH 2011 Courses* (Vancouver, British Columbia, Canada) *(SIGGRAPH '11)*. Association for Computing Machinery, New York, NY, USA, Article 8, 84 pages. https://doi.org/10.1145/2037636.2037644

Chris Wojtan, Nils Thürey, Markus Gross, and Greg Turk. 2010. Physics-Inspired Topology Changes for Thin Fluid Features. *ACM Trans. Graph.* 29, 4, Article 50 (jul 2010), 8 pages. https://doi.org/10.1145/1778765.1778787

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

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

Yizhong Zhang, Huamin Wang, Shuai Wang, Yiying Tong, and Kun Zhou. 2011. A deformable surface model for real-time water drop animation. *IEEE Transactions on Visualization and Computer Graphics* 18, 8 (2011), 1281–1289.

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

Wen Zheng, Jun-Hai Yong, and Jean-Claude Paul. 2006. Simulation of Bubbles. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Vienna, Austria) *(SCA '06)*. Eurographics Association, Goslar, DEU, 325–333.

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, C (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 Trans. Graph.* 34, 4, Article 115 (jul 2015), 9 pages. https://doi.org/10.1145/2766981

Bo Zhu, Ed Quigley, Matthew Cong, Justin Solomon, and Ronald Fedkiw. 2014. Codimensional Surface Tension Flow on Simplicial Complexes. *ACM Trans. Graph.* 33, 4, Article 111 (jul 2014), 11 pages. https://doi.org/10.1145/2601097.2601201

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