Fig. 1. Several thin-film phenomena as simulated using our proposed method. Counting from left to right: (1) surface flow on a oscillating soap bubble, (2) pinched-off droplets between two circular rims pulling away from one another, (3) a soap bubble bursting into tiny droplets and filaments after being poked from the right, (4) the vibrant, opal-like color pattern caused by Rayleigh-Taylor instability on a large-deforming, wet thin-film surface.

We propose a particle-based method to simulate thin-film fluid that jointly facilitates aggressive surface deformation and vigorous tangential flows. We build our dynamics model from the surface tension driven Navier-Stokes equation with the dimensionality reduced using the asymptotic lubrication theory and customize a set of differential operators based on the weakly compressible Smoothed Particle Hydrodynamics (SPH) for evolving point-set surfaces. The key insight is that the compressible nature of SPH, which is unfavorable in its typical usage, is helpful in our application to co-evolve the thickness, calculate the surface tension, and enforce the fluid incompressibility on a thin film. In this way, we are able to two-way couple the surface deformation with the in-plane flows in a physically based manner. We can simulate complex vortical swirls, fingering effects due to Rayleigh-Taylor instability, capillary waves, Newton’s interference fringes, and the Marangoni effect on liberally deforming surfaces by presenting both realistic visual results and numerical validations. The particle-based nature of our system also enables it to conveniently handle topology changes and codimension transitions, allowing us to marry the thin-film simulation with a wide gamut of 3D phenomena, such as pinch-off of unstable catenoids, dripping under gravity, merging of droplets, as well as bubble rupture.

CCS Concepts:
• Computing methodologies → Modeling and simulation.

Additional Key Words and Phrases: sph method, thin-film dynamics, surface tension flow, point-based surface

1 INTRODUCTION
Thin films are fascinating fluid phenomena on two levels. On the macroscopic scale, their surface-tension-driven dynamics morph them into elegant, minimal-surface geometries such as catenoids in an energy-optimized manner, which are captivating feats both artistically and mathematically. When coupled with external forces of air pressure, wind, or gravity, these tendencies create the unique bounciness we see in soap bubbles that is satisfying to watch. On the microscopic scale, thin films carry vibrant and delicate color patterns that arise from the interference of light bouncing between the varying film thicknesses, while the turbulent flows precipitate the surface to constantly evolve and contort, thereby creating a smoothly flowing color palette as rich as that of oil paint.

Research in computer graphics has not ceased to strive and capture the beauty that thin films bring about. Seminal previous works [Da et al. 2015; Ishida et al. 2017; Kim et al. 2015; Saye and Sethian 2013; Zheng et al. 2009; Zhu et al. 2014] devise methods to obtain...
visually appealing simulations of surface tension driven thin-film phenomena, focusing on the metamorphosis of thin-film surfaces. On the other end, many works [Azencot et al. 2015a, 2014; Hill and Henderson 2016; Huang et al. 2020; Stam 2003; Vantzos et al. 2018] push towards the simulation of the surface tension-driven richness evolution on the deformed surface, which, when combined with the optical insight in thin-film interference [Belcour and Barla 2017; Glassner 2000; Iwasaki et al. 2004; Jaszkowski and Rzeszut 2003; Smits and Meyer 1992] can be used to recreate the distinct beauty of iridescent bubbles. Recently, Ishida et al. [2020] highlight the importance of integrating the two aspects in achieving enhanced richness and plausibility; and proposes a successful method to jointly simulate surface deformation and thickness evolution. Nevertheless, the endeavor is far from being a closed case, and it yet calls for the realization of surface flow with more liveliness and sophistication, surface deformation with higher frequency details, and the integration with more 3D surface tension phenomena, which together bring a significant challenge for a simulation’s robustness, efficiency, and versatility.

In this paper, we attempt to tackle this challenge with a particle-based method. We represent the thin film with a set of codimension-1 particles and devise a set of differential operators based on the Smoothed Particle Hydrodynamics on the evolving surface. In the full 3D simulation, the inevitable compressibility of SPH causes volume loss that results in visual implausibilities. However, because our simulation is principally performed on a codimension-1 surface, the in-plane compression can be compensated by an expansion in its codimension-direction, which grants each particle a varying thickness that together outlines a curved surface in the three-dimensional space. The surface such prescribed allows for the computation of surface-tension-related forces that the codimension-1 SPH particles will carry out. In this way, a simulation cycle is completed with the SPH playing a dual role: both as the initiator who forms the curved surface with the compressible nature and the executor who carries out the dynamics resulting from such a curved surface. Using this design, we can simulate thin-film-specific behaviors while inheriting the SPH’s virtue of simplicity, adding little additional cost. With the incorporation of a physically-based surface tension model derived from the lubrication assumption, our simulation algorithm effectively reproduces a wide range of phenomena such as the Newton interference patterns, the Marangoni effect, capillary waves, and the Rayleigh-Taylor Instability.

Our particle-based, codimensional simulation method for thin-film surfaces bridges the mature research literature of SPH with the point-set surface representation, a promising avenue that is not yet thoroughly studied. Compared to mesh-based paradigms [Da et al. 2015; Ishida et al. 2020, 2017; Zhu et al. 2014], our particle-based system is flexible for dealing with the topology changes and codimension transitions, which are particularly pertinent to this application, since thin films are delicate constructions that are marked for their tendency to disintegrate, yielding some of the most exciting spectacles such as the pinching-off of the film surface into countless filaments and droplets. To this end, our method conveniently implements particle sharing with an auxiliary, standard 3D SPH solver following previous works [Akinci et al. 2013; Müller et al. 2003]. Our particles can be directly copied from codimension-1 to codimension-0 and vise versa without any extra processing, enabling the realization of a variety of thin-film phenomena with 3D components.

In summary, our contribution includes:

- A thin-film SPH simulation framework that jointly simulates large and high-frequency surface deformation and lively in-surface flows
- A meshless simulation framework with a set of SPH-based differential operators to discretize the thin-film physics on a curved, point surface
- The leverage of the SPH compressibility to couple the in-surface thickness evolution and the surface metamorphosis via physically-based surface tension
- A versatile particle surface representation method that conveniently handles topology and codimension 0-1 transition to integrate surface and volumetric SPH simulation.

2 RELATED WORKS

**Mesh-Based Dynamic Thin-Film.** To precisely capture the dynamic flow and irregular geometry of a thin film, a myriad of previous work adopts the idea of representing a thin film using a triangle mesh. With the thickness of a thin film coupled into the fluid simulation, modifications can be applied to the projection method of incompressible fluid simulation to fit the requirement of simulating flow dynamics on thin films [Ishida et al. 2020; Zhu et al. 2014]. The vortex sheet model, which uses circulation as a primary variable, is also a feasible option [Da et al. 2015]. Another branch of work [Ishida et al. 2017] focuses on the surface area-minimizing effect of surface tension. Apart from fluid simulation based on Navier-Stokes equations, the continuum-based model can also be used for simulating highly-viscous fluid [Batty et al. 2012; Bergou et al. 2010], where surface tension is modeled as an area-minimizing term within the energy-optimization elastic model. These works not only propose some computational algorithms but also studied the physical model of dynamical thin films, thus set up a remarkable baseline for this topic. However, they have to carefully trade-off between the ability to drastically change the topology of thin-film and solve non-trivial tangential flow on it.

**Mesh-Based Static Thin-Film.** When constraining the thin film to a fixed shape, the complexity and challenges emerging from an arbitrary-shaped triangle mesh can be alleviated. When the fixed shape is a sphere, special treatments are required to perform discretization in spherical coordinates to solve the Navier-Stokes equations [Bridson 2015; Hill and Henderson 2016], or governing equations coupled with the evolution of thin-film thickness and surface tension [Huang et al. 2020]. Beyond bubble, Azencot et al. [2015b] further explores the flow dynamics of a thin layer of fluid sticking to an arbitrarily shaped object. These works show drastic flow convection on the surface of a fixed geometry. We seek to liberate the fixed-surface restriction with our particle method.

**Level-Set Bubble and Foam.** Triangle mesh is not the only data structure for solving thin-film dynamics. When taking the air within bubbles into consideration, the whole physical system can be easily viewed as a two-phase fluid system. In this case, the level set on grids
is a widely adopted tool. It can be used to model surface tension in different fluids such as ferrofluids [Ni et al. 2020]. The multi-phase fluid system, where bubbles are surrounded by large bulks of water, i.e., foams [Aanjjaneya et al. 2013; Albadawi et al. 2013; Hong and Kim 2003; Kang et al. 2008; Patkar et al. 2013], can also be conveniently simulated. Further, with a relatively symmetric treatment to fluid and water, simulate both foams and water drops [Hong and Kim 2005], or standalone bubbles in the air [Zheng et al. 2009] after taking delicate care respecting to the thin of soap film. Unfortunately, level-set immediately means that large bulks of water and air away from the interface have to be included in the computation, limiting the algorithm’s complexity and performance.

**SPH Methods.** It can be seen that all Eulerian methods will face the dilemma between surficial flow topology changes. Therefore, we turn to the Lagrangian formalism, where the fluid within a thin film is represented by particles. The most popular particle-based fluid simulation method, Smoothed Particle Hydrodynamics (SPH) [Brookshaw 1985; Gingold and Monaghan 1977; Koschier et al. 2019; Liu and Liu 2010; Monaghan 1992; Monaghan and Lattanzio 1985; Morris et al. 1997; Solenthaler and Pajarola 2009], where discretization formulation based on radial smoothing kernels is used to approximate differential operators. The simplicity and high parallelizability make SPH useful for interactive fluid simulation [Müller et al. 2003], and can readily operate in conjunction with other fluid simulation algorithms [Band et al. 2018; Cornelis et al. 2014; Ihmsen et al. 2013]. It can also be extended to different applications such as simulating multiphase flow, computing magnetohydrodynamics, and blue noise sampling [Jiang et al. 2015; Price 2012; Tartakovsky and Meakin 2005]. Recently, SPH has been used for snow [Gissler et al. 2020], elastic solids [Peer et al. 2018], ferrofluids [Huang et al. 2019] and viscous fluids [Bender and Koschier 2016; Peer et al. 2015]. On different geometries, simulation algorithms based on SPH are also developed [Omong et al. 2006; Tavakkol et al. 2017]. The SPH method can also be coupled with a grid-based simulation [Losasso et al. 2008] or rigid-body simulation [Gissler et al. 2019]. As a particle-based method, the treatment of multi-phases, including thin films and foams [Yang et al. 2017a,b], is straightforward in SPH. Some works modeled surface tension with the SPH method [Akinci et al. 2013; Becker and Teschner 2007; Yang et al. 2016]. The work of Ando and Tsuruno [2011] captures and preserves thin fluid films with particles. However, the main body of the algorithm is grid-based, and surface tension is not shown in the physical model.

**Point-Based Thin Film.** The modeling of thin-film needs geometry information, which can be implicitly given by the shape of the point cloud. A shallow water simulation facilitated by a particle-based height field [Solenthaler et al. 2011] inspired us to represent the film thickness similarly. On this basis, we still need some geometrical operators on the surface. Laplacian operator can be generalized to codimension manifolds [Schmidt 2014] and discretized in an SPH way [Petronetto et al. 2013]. Closest point method (CPM) [Cheung et al. 2015; Ruuth and Merriman 2008] provides a way to approximate functions on the Cartesian space defined on manifolds, and it thus can be used to solve PDEs on the surface, which includes the governing equations of surface flow [Auer et al. 2012; Auer and Westermann 2013; Kim et al. 2013; Morgenroth et al. 2020]. Other methods are also available, including graph Laplacian [Belkin and Niyogi 2008], local triangular mesh [Belkin et al. 2009; Lai et al. 2013] and moving least squares [Lancaster and Salkauskas 1981; Levin 1998; Nealen 2004; Saye 2014]. The last one is used by Wang et al. [2020] to solve surface tension flows, and is also adopted by our framework.

### 3 METHOD OVERVIEW

Overall, our thin-film SPH method is an enhanced version of a conventional SPH solver with geometric information. We represent the thin film by a set of codimension-1 (surface) particles with local coordinates carried by each of them. The thickness of the film is concurrently estimated both by a surface convolution method resembling the computation of density in classical SPH, and also by equations derived from the mass-conservation law. This point-cloud representation not only allows a Lagrangian in-surface flow simulation but also gives us the exact shape of the manifold thin film. Simulating the thin-film deformation then becomes possible as we can compute the mean curvature of the particle-surface using equations derived from the mass-conservation law. Finally, our particle-based data structure provides a way to naturally transit a codimension-1 thin film into a codimension-0 fluid bulk, with the latter handled by a traditional SPH fluid solver. In the following sections, we will provide details of this algorithm.

### 4 THIN-FILM GEOMETRY

We assume that a soap film is a three-layered, volumetric structure whose characteristic thickness is far less than its characteristic length. As shown in Figure 2, we denote the upper and lower interfaces as $S^+$ and $S^-$, and central surface $S_C$. We build a surface coordinate

**Table 1.** A list of symbols used in our thin-film SPH model

| Symbol | Meaning |
|--------|---------|
| $S_C$  | center surface of soap film |
| $S_i^+$ | upper interface of soap film |
| $S_i^-$ | lower interface of soap film |
| $h$    | distance between interfaces and center surface |
| $\rho$ | density of fluid in soap film |
| $\xi^1, \xi^2, \xi^3$ | local coordinate variables |
| $e_1, e_2, e_3$ | basis vectors of local frame |
| $u$    | velocity of fluid |
| $\gamma$ | surface tension coefficient |
| $\Gamma$ | surfactant concentration |
| $v$    | normal component of velocities on soap film |
| $w$    | tangential components of velocities on soap film |
| $k_c$  | the curvature of center surface |
| $k_h$  | the local curvature of interfaces |
| $\alpha_e$ | diffusion coefficient of surfactant |
| $\alpha_p, \alpha_k, \alpha_d$ | pressure coefficients |

ACM Trans. Graph., Vol. 40, No. 4, Article 110. Publication date: August 2021.
Fig. 2. The schematic diagram of the soap film’s internal structure. Central surface $S_C$ is enveloped by two interfaces $S^\pm_i$. For point $O$ on $S_C$, the coordinate system is established with normal axis $n$ and tangential axes $(e_1, e_2)$. The thickness of film is $2h$ along $n$.

system $(e_1, e_2)$ near point $O$ on the central surface $S_C$, with a corresponding normal vector $n$, making the coordinates $(\xi^1, \xi^2, z)$ near point $O$. Since we assume that the interfaces $S^\pm_i$ are symmetric about the central surface, we take the heights of the upper(yellow) and lower(red) surfaces in the local coordinate system as $z = \pm h(\xi^1, \xi^2)$ respectively. Thus the thickness of the thin film is $2h$.

For a codimension-1 surface $S_C$, we follow the work of Wang et al. [2020] to define the gradient of a scalar field $s$, the divergence of a velocity field $\mathbf{v}$ and the Laplacian of a scalar field $s$ as

$$
\nabla_s s = \sum_{k=1}^{2} \frac{\partial s}{\partial \xi^k} e_k\nabla, \quad \nabla_s \cdot \mathbf{v} = \frac{1}{\sqrt{g}} \sum_{k=1}^{2} \frac{\partial}{\partial \xi^k} \left( \sqrt{g} g^{kl} \frac{\partial s}{\partial \xi^l} \right),
$$

$$
\nabla^2_s s = \frac{1}{\sqrt{g}} \sum_{k=1}^{2} \sum_{l=1}^{2} \frac{\partial}{\partial \xi^k} \left( \sqrt{g} g^{kl} \frac{\partial s}{\partial \xi^l} \right),
$$

where

$$
g^{kl} = \frac{\partial x^i}{\partial \xi^k} \frac{\partial x^j}{\partial \xi^l}
$$

is the metric tensor on $S_C$, $x$ is the 3D Euclidean coordinates, and $g = \det(g^{kl})$.

For a soap film, the scale of the normal direction, which is typically in the order of $10^{-7}m$, is very small in comparison to that of the tangential direction. Hence, we will not consider small quantities of higher order with respect to $h$ in the numerical simulation. The normal direction, tangential direction, and curvature on the upper and lower surfaces $S^\pm_i$ can be approximated as

$$
\mathbf{n}^\pm = \frac{\pm \mathbf{n} - \nabla s h}{|\pm \mathbf{n} - \nabla s h|} = \pm \mathbf{n} - \nabla s h + O(h^2),
$$

$$
t^i_+ = \frac{e_1 \pm e_2 \times \nabla s h}{|e_1 \pm e_2 \times \nabla s h|} = e_1 \pm e_2 \times \nabla s h + O(h^2),
$$

$$
t^j_+ = \frac{e_2 \pm e_1 \times \nabla s h}{|e_2 \pm e_1 \times \nabla s h|} = e_2 \pm e_1 \times \nabla s h + O(h^2),
$$

$$
\kappa^\pm = -\frac{1}{2} \nabla \cdot \mathbf{n}^\pm = \kappa_C + \kappa_H + O(h^2),
$$

where $\nabla$ is the three dimensional gradient operator that satisfies $\nabla s = \nabla \cdot \mathbf{n} (\mathbf{n} \cdot \nabla)$. The curvature of the center surface is given by $\kappa_C = 0.5 \nabla^2 s C$, and the local curvature caused by the surface thickness is defined as $\kappa_H = 0.5 \nabla^2 h$. In addition, the gradient operator on the upper and lower surfaces $S^\pm_i$ can be approximated as

$$
\nabla s^\pm = \nabla - \mathbf{n}^\pm (\mathbf{n} \cdot \nabla) = \nabla s \pm \nabla s h (\mathbf{n} \cdot \nabla) \pm n(\nabla s h \cdot \nabla) + O(h^2).\quad (4)
$$

5 GOVERNING EQUATIONS

In this section, we derive the evolution equation of the central surface $S_C$. Considering the flow field near point $O$, we assume that the velocity field $\mathbf{v}$ within $-h < z < h$ satisfies 3D incompressible Navier–Stokes equations:

$$
\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + f, \quad -h < z < h.
$$

The term $f$ here denotes the external force. Following Chomaz [2001], we assume a force-balancing boundary condition at the interfaces $S^\pm_i$:

$$
(p - p_a + 2k^\pm)/n^\pm = -\nabla s^\pm + \mu[\nabla u + (\nabla u)^T] \cdot n^\pm, \quad z = \pm h,\quad (6)
$$

where $p_a$ is the air pressure.

By substituting (3) and (4) into (6) and employing Taylor expansion, we can obtain the dominant governing equations at the center surface $S_C$ where $z = 0$:

$$
\rho \frac{Du}{Dt} + O(h^3) = 2
\begin{pmatrix}
\nabla^2 \left( \kappa_C \nabla \mathbf{u} \cdot \mathbf{n} \right) + \nabla \mathbf{u} \cdot \mathbf{n} = 2\nabla^2 \left( \kappa_C \nabla \mathbf{u} \cdot \mathbf{n} \right) + \nabla \mathbf{u} \cdot \mathbf{n} + \frac{1}{h} \mathbf{n} \nabla \mathbf{u} + \mu \nabla^2 \mathbf{u} + f.
\end{pmatrix}
$$

Here $\mathbf{u}_s = \mathbf{u} - (\mathbf{n} \cdot \mathbf{u}) \mathbf{n}$ is the tangential velocity. We refer the readers to Appendix A for details. The mass conservation equation in (5) immediately yields the equation for the conservation of the height field [Solenthaler et al. 2011] as

$$
\frac{Dh}{Dt} = -(\nabla_s \cdot \mathbf{u}_s) h.
$$

The strength of surface tension of soap water is under the influence of multiple factors, including the water temperature and the concentration level of surfactant. We will only consider the latter under the assumption of always keeping a constant room temperature. Surfactants refer to chemical substances that can significantly reduce water’s surface tension when dissolved by them, like soap, oil, and kitchen detergents. The relationship between the concentration of a surfactant $\Gamma$ and the thin film’s surface tension coefficient $\gamma$ can be described by the following Langmuir equation of state:

$$
\gamma(\Gamma) = \gamma_0 + RT \Gamma \log \left( 1 - \frac{\Gamma}{\Gamma_m} \right) \quad [Muradoglu and Tryggvason 2008],
$$

where $\gamma_0$ is the surface tension coefficient of an untainted interface ($\Gamma = 0$), $\Gamma_m$ is the maximum packing concentration, $R$ and $T$ are ideal gas constant and temperature respectively. All these four quantities are considered as constants here. In our case, where $\Gamma \ll \Gamma_m$, the Langmuir equation of state can be approximated by $\gamma(\Gamma) = \gamma_0 - \gamma_d \Gamma$ with $\gamma_d = \gamma_0$ [Xu et al. 2006]. We further assume that the surfactant concentration $\Gamma$ is invariant with the normal coordinate and it satisfies the classical convection–diffusion equation with the diffusion coefficient $\sigma_c$ as:

$$
\sigma_c
$$
we assume symmetry about with a certain mass of fluid. In the regions where particles are

Given the SPH integration of a physical quantity \(A_i\) on the thin film \(A_i = \sum_j s_t^i A_{ij} W_{ij} = \sum_j W_{ij} A_{ij}\), where \(s_t^i\) represents the control area of particle \(j\). If we set the function \(A\) to equal \(h\), then we obtain the SPH integrated expression of the (half)-height field \(h\) on a thin film as:

\[
\begin{align*}
    D\hat{h} &= \sum_j V_{ij} \hat{w}_{ij}, \\
    \frac{D\Gamma}{Dt} &= \alpha_c \nabla_2 \Gamma.
\end{align*}
\]

which will be used to compute the interference color for rendering.

Now we compare the two forms of height computation. The advected height is obtained by temporally integrating the flux about a particle from the velocity field. The numerical height is obtained by first temporally integrating the particles’ positions using the same velocities and then estimating the volume density. The two forms are equivalent if there are no numerical errors. In practice, they deviate in their behavior, each with its advantages and drawbacks. The numerical height is free from error accumulation, as it is recomputed at each step, but it is inevitably smoothed due to the SPH kernel convolution. The advected height can suffer from numerical drift, but it can preserve higher-frequency details. In our system, we use both to their advantages. We use the numerical height to compute the dynamics, where smoothness is beneficial for robustness, and we use the advected height in the color computation for more appealing visual results. To reduce the numerical drift, we normalize the advected height at each timestep to conserve the total volume. As shown in Figure 3, the numerical height is much smoother and does not preserve the flow pattern introduced by convection. We provide further comparisons in the appendix.

\[\frac{D\hat{h}}{Dt} = - (\nabla \cdot \hat{u}) \hat{h}, \quad (12)\]

and the first equation of (10) describing the momentum evolution of fluid. It offers a smooth surface that prevents numerical instabilities, however “blurs” the height field with a mollifying kernel. To preserve the vibrant and turbulent flow patterns originating from the advection-diffusion evolution of the height field in (10), we introduce an additional advected height \(\hat{h}\), that actually follows the Lagrangian evolution equation (8).

\[\frac{D\hat{h}}{Dt} = - (\nabla \cdot \hat{u}) \hat{h}, \quad (12)\]

Fig. 3. The comparison between numerical height (left) and advected height (right).

Fig. 4. Bubble oscillation: A perfect sphere oscillates under perturbations and vortices (upper part), and an irregular bubble (lower part).

Fig. 4. Bubble oscillation: A perfect sphere oscillates under perturbations and vortices (upper part), and an irregular bubble (lower part).

6 SPH DISCRETIZATION

6.1 Particle Height

The key intuition underpinning our SPH particle height model in codimension-1 space is that each thin-film particle carries along with a certain mass of fluid. In the regions where particles are denser, the thin film is deemed thicker because more fluid is contained per unit area on the tangential plane. This intuition can be mathematically captured by the codimension-1 SPH framework. Given the SPH integration of a physical quantity \(A\) on the thin film \(A_i = \sum_j s_t^i A_{ij} W_{ij} = \sum_j W_{ij} A_{ij}\), where \(s_t^i\) represents the control area of particle \(j\). If we set the function \(A\) to equal \(h\), then we obtain the SPH integrated expression of the (half)-height field \(h\) on a thin film as:

\[
\begin{align*}
    D\hat{h} &= \sum_j V_{ij} \hat{w}_{ij}, \\
    \frac{D\Gamma}{Dt} &= \alpha_c \nabla_2 \Gamma.
\end{align*}
\]

In this way, the height of a particle is decided by (i) the particle distribution near its neighborhood and (ii) the volume that each neighboring particle carries. To view it another way, this translates the typical SPH mass density (how much mass per unit area) [Tartakovsky and Meakin 2005] to the volume density (how much volume per unit area), which naturally equates the concept of height on a thin film, an correlation previously explored by Solenthaler et al. [2011]. Note that \(h\) and \(V\) are one-sided quantities because we assume symmetry about \(S_C\), and the half volume \(V_i = m_i / \rho_i\) is assigned with every particle at the beginning and is conserved throughout the simulation.

The height \(h_i\) that is so computed will be termed the numerical height, and we use it for all computation relevant to the dynamics of soap bubbles, e.g., the discretization of all differential operators,
The velocity difference $\vec{u}_{ij}$ we use for divergence operator is different from the projection of $\vec{u}_j - \vec{u}_i$ to codimension-1 plane $\vec{e}^i$. The reason arises from the insight that a streamline on the surface, with a constant velocity rate $\alpha$ (i.e., $|\vec{u}_j| = |\vec{u}_i| = \alpha$, $\vec{u}_i \cdot \vec{n}_i = 0$), should yield a zero divergence. However, if we directly project $\vec{u}_j$ to the local plane at particle $i$, the deviation of normal directions $\vec{n}_j - \vec{n}_i \neq 0$ will incorrectly distort the project velocity rate as $\alpha_j - (\vec{u}_j \cdot \vec{n}_j) \neq \alpha_i$. To avoid this numerical artifact, we imitate the idea of the great-circle-advection algorithm used by Huang et al. [2020]; in particular, we pass the same coordinate values from the local frame at $j$ to that at $i$ instead of performing a projection, as shown in Figure 6.

First, we build temporary coordinate systems $(\vec{e}^i, \vec{n}_j)$ and $(\vec{e}^j, \vec{n}_j)$ at local frames of $i$ and $j$, with their normal directions unchanged, and one tangential axis, say, $\vec{e}_1$, along the direction of $\vec{r}_{ij}$:

$$
\vec{n}_i = \vec{n}_j, \quad \vec{n}_j = \vec{n}_j,
$$

$$
\vec{e}_1 = \frac{\vec{r}_{ij}}{|\vec{r}_{ij}|},
$$

$$
\vec{e}_2 = \vec{n}_i \times \vec{e}_1^i, \quad \vec{e}_2 = \vec{n}_j \times \vec{e}_1^j.
$$

Then, with the the decomposition of $\vec{u}$ at temporary coordinates $(\vec{e}, \vec{n})$ we have

$$
\vec{u}_i = \bar{u}_i \vec{e}_1^i + v_i \vec{e}_2^i + w_i \vec{n}_i,
$$

$$
\vec{u}_j = \bar{u}_j \vec{e}_1^j + v_j \vec{e}_2^j + w_j \vec{n}_j.
$$

Finally, as shown in Figure 6, we compute the actual velocity difference $\bar{u}_{ij}$ used for divergence as:

$$
\bar{u}_{ij} = (\bar{u}_j - \bar{u}_i) \vec{e}_1^i + (v_j - v_i) \vec{e}_2^i.
$$

Fig. 6. Definition of $\bar{u}_{ij}$ in (13). Here $\bar{n}$ represents normal axes, $\vec{e}_1$ is a a unit vector pointing from $i$ to $j$, and $\vec{e}_2$ is the cross product of $\bar{n}$ and $\vec{e}_1$.
We discretize the fluid forces in (10) using our surface SPH operators. With a linear formulation, the state equation which relates the state with the thin-film pressure we derived in the previous section to feedback. In our numerical algorithm, we combine this equation of state with the thin-film pressure to enhance the rotational motion on the bubble surface.

6.3 State Equation
With a linear formulation, the state equation which relates the compression of particles and the pressure of fluid in a classical SPH scheme can be written as

\[
p_i = \alpha_p \left( \frac{p_i}{\rho_0} - 1 \right),
\]

(18)

with \(\alpha_p\) as a constant parameter specifying the stiffness of the feedback. In our numerical algorithm, we combine this equation of state with the thin-film pressure derived in the previous section to obtain

\[
p = \kappa h y + \nabla_s \cdot u_s,
\]

(19)

which suggests that the fluid pressure inside the thin film is a function of the local curvature \(\kappa_h\) and velocity’s divergence on surface. With the analogy of density \(\rho\) in the classical SPH scheme and the numerical height \(h\) in ours, we acquire a final formulation of the particle pressure as

\[
p_i = \alpha_h \left( \frac{h_i}{h_0} - 1 \right) + \alpha_k y_i (\kappa h)_i + \alpha_d (\nabla_s \cdot u_s)_i.
\]

(20)

Here \(h_0\) is the rest thickness of thin films, and \(\alpha_h, \alpha_k, \alpha_d\) are three constant parameters controlling the tangential incompressibility of thin-film system. The first term administers the weakly compressibility to ensure that the particles are relatively evenly populated in the simulation domain to ensure numerical robustness, while the latter two terms achieve the surface-tension-driven behaviors of the thickness evolution. Specifically, the stiffness of our system is much lower than that of a standard SPH algorithm used for volumetric incompressible flow simulation. Therefore, the dynamics of particles are dominated by physical evolution instead of by the tendency to be evenly distributed.

6.4 Force Discretization
We discretize the fluid forces in (10) using our surface SPH operators (13). The mean curvature \(\kappa = 0.5 \nabla_s^2 S_C\), where \(S_C : z = z(x, y)\), is calculated using the surface Laplacian operator.

\[
2k_c = \frac{\sum V_j (r_{ij} \cdot n_j) 2|\nabla W_{ij}|}{|r_{ij}|^3}.
\]

(21)

We further introduce revise a vorticity confinement force [Selle et al. 2005; Yoon et al. 2009; Zhu et al. 2010] to enhance the vortical motion on the bubble surface. We carry the vorticity \(\zeta_i\) on every particle \(i\), which is subject to a similar convection-diffusion equation of the surfactant concentration \(n\). The vorticity confinement force \(f_i^z\) takes the form

\[
f_i^z = -\sum_j V_j \times \left( \frac{\zeta_i}{|r_{ij}|^3} \right).
\]

7 TIME INTEGRATION
Before computing the forces from the SPH discretization of (10), we first calculate the surface tension coefficient \(y_j\) and the fluid pressure \(p_i\) for each particle as shown in Algorithm 1. After that, we compute all forces in Algorithm 2. Following the position update, the local frames and metric tensors are reconstructed at every particle with the PCA-based method [Wang et al. 2020]. Finally, the numerical height \(h\) is updated with (11). We summarize our time integration scheme in Algorithm 3.

Algorithm 1 Compute Preliminary Variables for particle \(i\)

1. Update surface tension: \(y_j = y_0 - y_2 y_i^2\).
2. for each particle \(j\) in the neighborhood of \(i\) do
3. Update divergence: \((\nabla_s \cdot u_s)_i = \sum_j V_j \nabla_s u_{ij} \cdot \nabla_s W_{ij}\).
4. Update local curvature: \((\kappa h)_i = \sum_j V_j (h_j - h_i) 2|\nabla W_{ij}| |r_{ij}|^3\).
5. end for
6. Update pressure:
\[
p_i = \alpha_h \left( \frac{h_i}{h_0} - 1 \right) + \alpha_k y_i (\kappa h)_i + \alpha_d (\nabla_s \cdot u_s)_i.
\]

7.1 Codimension Transition
Under certain circumstances, the codimension-1 thin film will shrink or rupture into volumetric droplets. We capture this phenomenon with the criteria that \((\kappa h)_i > k_0\) or \(n_i < 2\). Here \(k_0\) is a constant depending on the setup of the scene, and \(n_i\) is the number of neighbors within the SPH kernel radius. The first criterion checks if the particle still lives on a relatively even surface, and the second checks
Algorithm 2 Compute Forces for Particle $i$

1. Compute external force: $f_i^e = m_i g + f_{ext}$.
2. for each particle $j$ in the neighborhood of $i$ do
3.  Compute vorticity confinement force: $f_i^v = -\sum_j \vec{r}_{ij} \times (\vec{u}_i \times \vec{n}_j)$.
4.  Compute pressure force with symmetric form: $f_i^p = 2V_i \sum_j h_i V_j \left( \frac{p_i}{h_i} + \frac{p_j}{h_j} \right) \nabla_s W_{ij}$.
5.  Compute Marangoni force with difference form: $f_i^m = \frac{V_i}{h_i} \sum_j V_j \left( \frac{1}{h_j} (\vec{r}_j - \vec{r}_i) \nabla_s W_{ij} \right)$.
6.  Compute capillary force: $f_i^c = \frac{V_i}{h_i} \sum_j V_j \left( -\vec{r}_{ij} \cdot \vec{n}_j \right) \frac{2|\nabla_s W_{ij}|}{|\vec{r}_{ij}|}$.
7.  Compute viscosity force: $u_{ij} = u_j - u_i$, $f_i^v = V_i \mu \sum_j V_j \left( u_{ij} - (u_i \cdot \vec{n}_j) \vec{n}_j \right) \frac{2|\nabla_s W_{ij}|}{|\vec{r}_{ij}|}$.
8. end for

Algorithm 3 Advance a Time Step

1. for each particle $i$ do
2.  Compute preliminary variables with Algorithm 1.
3.  Update advected height: $h_i + = -\Delta t (\nabla_s \cdot u_i)_i$.
4.  Diffuse surfactant concentration:
   \[ \Gamma_i + = \alpha_{\Delta t} \sum_j \frac{V_j}{h_j} (\Gamma_j - \Gamma_i) \frac{2|\nabla_s W_{ij}|}{|\vec{r}_{ij}|}. \]
5.  Compute forces with Algorithm 2.
6.  Update velocity: $u_i + = \frac{\Delta t}{m_i} \left( f_i^e + f_i^v + f_i^p + f_i^m + f_i^c + f_i^f \right)$.
7.  Update position: $r_i + = u_i \Delta t$.
8.  Update local frame $e_i$, normal $n_i$ and metric tensor $g_i$.
9.  Update numerical height: $h_i = \sum_j V_j W_{ij}$.
10. end for

Fig. 9. Rayleigh-Taylor instability simulated on a piece of dynamically deforming thin film. Upon initialization, the upper half of the thin film carries a larger density, and the lower half carries a larger volume. The interface evolves into finger-like patterns, which further disintegrates into numerous delicate, colorful threads, making up the rich and realistic palette as seen in the last two subfigures.

Fig. 10. An example of codimension transition. In the left subfigure, codimension-1 particles are colored with orange and codimension-0 particles with green. In the right subfigure, the surface consisted of codimension-1 particles is also displayed. One can see that at the tip of the cone formed by the thin film, some particles are transiting to codimension-0 because they passed the threshold.

if the particle is all alone. When both criteria are met, particle $i$ is deleted from our surface SPH solver and transferred to a volumetric SPH solver [Akinci et al. 2013; Yang et al. 2017b; Zhu et al. 2010]. Conversely, for a volumetric particle $i$, if there is a codimension-1 particle $j$ nearby, and $(r_j - r_i) \cdot n_j < \text{threshold}$, particle $j$ is deleted from 3D SPH solver, and transported to our codimension-1 surface SPH solver. An example of codimension-1-to-0 transition is illustrated in Figure 10. We can see that green-colored volumetric particles are detached from the orange-colored codimension-1 thin film, at where the thin film shows a sharp corner. Under surface tension, the bulk fluid tends to form a spherical drop.

7.2 Additional Air Pressure Difference

For a closed bubble, the air pressure differs between the interface, as suggested by the Young-Laplace equation, which tends to contract the bubble into a singularity. Thus an additional force is needed to
retain its shape. We calculate the pressure inside the bubble \( p_b \) with the ideal gas equation \( p_b V_b = n \rho V_b = k \), where \( n \rho \) is a constant number. The volume of closed bubble \( V_b \) is given by a sum with pyramid volume formula \( V_b = \sum_i \frac{1}{6} |\mathbf{n}_i \cdot \mathbf{x}_i| \), with \( \mathbf{x}_i \) the vector pointing from the gravity center of all particles on the bubble to particle \( i \). Afterwards, an external force \( f_{ext} = \frac{\rho_b - \rho}{2h} \mathbf{n} \) is added to all particles.

### 7.3 Particle Reseeding

During the evolution of soap films, the total area of films may be enlarged from the boundary. For example, a common way to produce bubbles is to blow air into a circle soap film on a ring soaked with soap water, and as the bubble forms, new portions of soap film are replenished from the ring.

Our algorithm includes a reseeding step for every iteration to implement this feature. At the end of every time step, every fixed boundary particle \( i \) is checked with its nearest non-boundary neighbor particle \( j \), and if \( r_{ij} > \text{threshold} \), a new particle \( k \) is reseeded at \( r_k = \frac{r_i + r_j}{2} + \mathbf{\sigma} \), with a random perturbation \( \mathbf{\sigma} \). The perturbation term alleviates artifacts emerging from regular patterns in the distribution of newly-generated particles. The mass of a new particle \( k \) inherits from \( m_k = m_j \), and \( \mathbf{u}_k = \frac{\mathbf{u}_i}{2} \). While adding particles to SPH solvers can undermine the physical consistency and cause visible artifacts if not treated carefully, we note that in our case, this reseeding step only takes place near the boundary and is viewed as the physical process of fluid flowing from the boundary, so no mass or momentum redistribution is performed.

### 7.4 Boundary Treatment

For our various examples that involve interaction with circular or square solid rims, it is necessary to devise appropriate boundary handling methods. The task is twofold: First, the particles should not penetrate the solid boundary. Second, a particle should behave consistently when its neighborhood is under-sampled near the boundary. For the first task, we correct the velocities of particles that are going to penetrate the solid boundary defined by an implicit boundary (e.g., the solid ring in the Newton ring example). For the second task, we adopt a geometric compensation method illustrated in Figure 12. For a particle \( P \) with its neighborhood \( C \) truncated by the boundary, we find the nearest neighboring particle \( B \) that is on the boundary, take its mirrored point with respect to particle \( P \) as \( B' \), and draw the secant \( S \) across \( B' \) perpendicular to \( B' - P \). All neighboring points sampled within the circular segment defined by \( C \) and \( S \) are counted twice. If the boundary is a straight line, this method would precisely “clone-stamp” a part of the sampled region to fill up the empty region. Otherwise, the cloned region would not fill the empty region perfectly, then we estimate the unsampled area and adjust the compensation accordingly.

We adopt this geometric compensation method in place of typical boundary particle method for two reasons: first, the relative positions of the boundary particles to the fluid particles must change as the surface deforms, which adds complication to the traditional ways with multiple layers of boundary particles. Secondly, we encourage the spatial variance of particle density with low stiffness, so having boundary particles with fixed volume or mass would not do justice. For example, if the particles congregate along the border, we would want the area near the border to rise in height/density, in which case this adaptive clone-stamp strategy is useful as it preserves the state of the fluid. This mirroring approach to enable large density variation is inspired by Keiser et al. [2006]; the dynamic and adaptive nature is also similar to the Ghost SPH approach [Schechter and Bridson 2012].

### 8 THIN-FILM VISUALIZATION

The color in all of our simulations is computed with the CIE Standard Illuminant D65, a standard illuminant that represents the average, open-air illumination under daylight. Given the discretized spectral power distribution, we compute the reflection, refraction, and
environmental lighting will be on top to conform it with the rest of the environment.

9 RESULTS

Thickness Profile. Here a piece of a planar thin film is confined in a vertically positioned rim, and the gravitational force pulls the fluid volume to amass at the bottom, resulting in a height field that decreases with the elevation. The vertically varying thickness would result in a set of horizontal color stripes known as the Newton interference pattern. The thickness profile $\eta = \eta(z)$ can be approximated by an exponential function $\eta(z) = \eta^* \exp\left((\rho g \eta^* z)/(2 (\sigma_0 - \sigma^*))\right)$ derived by Couder et al. [1989], where $\eta^*$ and $\sigma^*$ represents the surface's equilibrium thickness and surface tension coefficient when horizontally placed. Our experiments use $\eta^* = 387.875 \text{nm}$ and $\sigma^* = 71.996 \cdot 10^{-6} \text{N/m}$. Since our surface simulation is allowed to be compressible in nature, it handles the volume aggregation at the bottom automatically. As displayed in Figure 11, both the visual result of Newton interference patterns and the thickness profile matches the real-world data in trend.

Surface Flow on Oscillating Bubbles. As shown in the upper part of Figure 4, we initialize the simulation bubble as a perfect sphere. In the first 1.0 second, we apply an impulse towards the center at the north and south poles. Meanwhile, one hundred in-plane vortices are seeded randomly. The induced capillary wave by the initial perturbation propagates through the bubble, causing it to deform and oscillate. The height field is evolved by the surface deformation and vortical flows jointly, forming vivid color patterns. Our method is independent of the initialization method and can be easily extended to bubbles with custom shapes. The lower part of the same figure shows a simulation initialized on an irregular mesh without the initial impulse. Furthermore, as shown in Figure 7, the simulation can also be extended to a half bubble standing on a plane. The initial impulse in Figure 7 is applied at the top right region of the bubble instead of the north-pole.

Bubble Rupture. In the example shown in Figure 5, we burst a bubble into filaments and droplets. We first manually mark every particle in a small region as “punctured”, and give it an inward push, imitating the bubble to be poked by, for example, a human finger. Then, the surface near that region is bent inward by the momentum of the initial impulse, causing more and more codimension-1 particles to meet the criteria of codimension transition and hence rupture into volumetric fluid particles. These particles are then taken over by the 3D SPH solver, morphing them into numerous thin filaments.
we initialize the thin film as a half-sphere. Particles under gravity whose direction circulates periodically around the surface. On the volumetric SPH solver only. 1.5 times as large, reflected in the color difference. Under gravity, tend to fall down, and under surface tension tend to contract. When thin film.

As shown in Figure 14, Thin-film Dripping with Circular Rim. In Figure 8, a thin film is confined in a square rim, and we apply a large gravitational pull whose direction circulates periodically around the surface. On the surface, we seed 20 random vortices and perform a slight perturbation to the volume of particle $V_i$ as well as density $\rho_i$ with Perlin noise, which reflects in the initial color pattern. The rotating force causes the particles to perform in-plane and out-of-plane motions alternatively, forming appealing color patterns and surface morphology. In Figure 9, we use a circular rim to contain a thin film and recreates on it the R-T instability, a beautiful, classic flow pattern that can be captured by SPH method [Solenthaler and Pajarola 2008]. Unlike the oscillating bubble example, where we desire the bouncy feeling, here, we set the surface tension coefficient low to give the film a soft and mellow feeling, with plenty of high-frequency motions. In addition to the downward gravity, we apply a large external force $F_{\text{ext}} = \beta \left[ (\cos \alpha, \sin \alpha, 0) + (0, \cos 2\alpha, \sin 2\alpha) \right]$, in which $\beta$ is the scalar of force strength and $\alpha = 2\pi t/T$ where $t$ is the simulation time and $T$ is the period. We initialize the film so that the upper half has a density 4 times as large, and the lower half has a height 1.5 times as large, reflected in the color difference. Under gravity, the heavier top moves down, and the height difference tends the bottom to move up, counteracting each other and evolve into various finger-like patterns. Besides the flow on the surface, the aggressive deformation of the surface itself is also reflected by the color. For example, the stretched parts are visibly darker than the rest of the thin film.

**Thin-film Dripping with Circular Rim.** As shown in Figure 14, we initialize the thin film as a half-sphere. Particles under gravity tend to fall down, and under surface tension tend to contract. When the tip of the thin film contracts, some particles transform from codimension-1 to codimension-0, separating from others and drops down, forming droplets under the surface tension modeled by our auxiliary volumetric SPH solver.

**Catenoid.** Consider two rims connected by a thin soap film are pulled away from each other. The thin film will try to shrink under the area minimizing tendency, forming a catenoid, as have been validated numerically in Appendix B. Figure 13 shows the rendered result our simulation. After the separation between the two rims exceeds the Laplace limit, no more catenoids can sustain. Then, the membrane would further contract and collapse into a droplet in the center. As in the figure, our system can accurately simulate the surface-tension-driven deformation while considering the color changes due to the surface elongation.

**Droplet Marangoni Effect.** In the previous examples, we showcase the transition from surface particles to volumetric particles. As shown in Figure 15, we highlight the reverse process by releasing four drops of soap solution with high surfactant concentration $\Gamma$ to the thin film with low surfactant concentration. As the droplets impact the film, they are transformed as codimension-1 particles and blend into the surface, significantly reducing the local surface tension of the thin film. The response is twofold: first, the reduced surface tension can tolerate higher surface curvature; therefore, one can observe more high-frequency vibrations of the surface; secondly, a soapy droplet would significantly increase the soap concentration at the impact location, thereby creating a zone with low surface tension. Thus, a nonzero gradient of surface tension would occur at the perimeter of this zone, pointing outsides, pushing particles out, a phenomenon known as the Marangoni effect. As this happens, more particles will cluster at the outskirt of the low surface tension zone, resulting in an increase in film thickness, which is reflected in the color change.

**10 PERFORMANCE**

Our algorithm benefits from the intrinsic parallelizability of the particle method and thus is highly efficient. All parts in Algorithm 1, 2 and 3 are implemented in parallel with OpenMP for multi-core processors. However, the codimension transition takes a linear time complexity because it needs to move particles between the surface

ACM Trans. Graph., Vol. 40, No. 4, Article 110. Publication date: August 2021.
and volumetric solvers. The performance of all scenes is displayed in Table 2 in Appendix B.

11 DISCUSSION AND FUTURE WORKS

We propose a novel particle-based, thin-film simulation framework that jointly facilitates large surface deformation and lively in-surface flows. Our dynamic model is based on the surface-tension-driven 3D Navier-Stokes equations simplified under the lubrication assumption. We devise a set of differential operators on curved point-set surfaces that are discretized by SPH. Our key insight lies in that the compressible nature of SPH, which tends to create artifacts in its typical usage, can be leveraged to our benefit by defining an evolving height field $h$ that enables the incorporation of the surface tension model into the thin-film fluid. Due to the particle-based nature, our method easily handles codimension transitions and topology changes and conveniently integrates an auxiliary 3D SPH solver to simulate a wide gamut of visually appealing phenomena simultaneously featuring surface and volumetric characteristics, such as the pinch-off of catenoids, dripping from a thin film, the merging of droplets, as well as bubble rupture.

The main limitation of the proposed algorithm is its incapability of dealing with surfaces carrying non-manifold intersections. In the future, we seek to extend it to simulate the interaction of complex bubble clusters that form a network of plateau borders. Then, the dilemma between particle density variation required for interesting color patterns and relatively even distribution of particles needed for the SPH framework limits our system in recreating color fields with a steep gradient, which may be alleviated by adopting the adaptive kernels. We also look forward to exploring more dynamic interactions with solid boundaries, e.g. activated by boundary deformation, intrusion, or even inter-crossing.

The codimension transition is another challenging topic, for the criterion must be carefully chosen. If the transition happens too easily, the thin film can easily break because a hole of volumetric particles appears somewhere. If it’s too hard, the thin film will form unnatural sharp corners instead of turning to the volumetric SPH solver. Due to the lack of adaptivity, the simulation of pinch-off is affected by subjectively defined parameters more or less, which we think is a possible direction for future improvement. On the other hand, adding and deleting particles during the reseeding step has been a long-standing and difficult problem in conventional SPH simulation, which might cause physical quantity inconsistency. Our thin-film SPH model alleviated this problem thanks to its compressible nature and low stiffness coupling density and pressure on the particle thin film, although the volumetric particles in the simulation (in particular when the codimension transition occurs) could still suffer from such inconsistencies when transferring mass and momentum between the thin film.

On the animation side, the level of sophistication in the color pattern that our simulation generates is not quite at the level of real-world verismimitude. That is because our displayed particles are exactly the ones used for the simulation, which is inevitably limited in the number. In the future, we may further investigate the possibility of using our SPH system as a background simulation, on top of which a larger number of tracker particles are advected, to make for visualization of improved richness and fidelity. Finally, we hope to customize a set of rendering algorithms that can directly take advantage of the geometric representation and computation tools we have already developed for the simulation to allow more consistent and physically accurate visual results.

ACKNOWLEDGMENTS

We thank all the anonymous reviewers for their constructive comments. We acknowledge the funding support from NSF 1919647 and Dartmouth UGAR. We credit the Houdini Education licenses for the video rendering.

REFERENCES

Mengdi Wang, Yitong Deng, Xiangxin Kong, Aditya H. Prasad, Shiying Xiong, Bo Zhu. 2021. A monolithic mass tracking formulation for bubbles in incompressible flow. J. Comput. Phys. 247 (2013), 17–61.

Nadir Acik, Gizem Akinci, and Matthias Teschner. 2013. Versatile surface tension and adhesion for SPH fluids. ACM Transactions on Graphics (TOG) 32, 6 (2013), 1–8.

Albaidawi, DB Donoghue, AJ Robinson, DB Murray, and YMC Delauro. 2013. Influence of surface tension implementation in volume of fluid and coupled volume of fluid with level set methods for bubble growth and detachment. International Journal of Multiphase Flow 53 (2013), 11–28.

Ryosho Ando and Reiji Tsuruno. 2011. A particle-based method for preserving fluid sheets. In Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation. 7–16.

Stefan Auer, Colin B Macdonald, Marc Treib, Jens Schneider, and Rüdiger Westermann. 2012. Real-time fluid effects on surfaces using the closest point method. In Computer Graphics Forum, Vol. 31. Wiley Online Library, 1909–1923.

Stefan Auer and Rüdiger Westermann. 2013. A Semi-Lagrangian Closest Point Method for Deforming Surfaces. In Computer Graphics Forum, Vol. 32. Wiley Online Library, 505–514.

Omri Azencot, Orestis Vantzos, Max Wardetzky, Martin Rumpf, and Mirela Ben-Chen. 2015a. Functional thin films on surfaces. In Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 137–146.

Omri Azencot, Orestis Vantzos, Max Wardetzky, Martin Rumpf, and Mirela Ben-Chen. 2015b. Functional thin films on surfaces. In Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 137–146.

Omri Azencot, Steffen Weidmann, Maks Ovsjanikov, Max Wardetzky, and Mirela Ben-Chen. 2014. Functional fluids on surfaces. In Computer Graphics Forum, Vol. 33. Wiley Online Library, 237–246.

Stefan Band, Christoph Gissler, Markus Ihmsen, Jens Cornelis, Andreas Peer, and Matthias Teschner. 2018. Pressure-boundaries for implicit incompressible SPH. ACM Transactions on Graphics (TOG) 37, 2 (2018), 1–11.

Christopher Batty, Andres Uribe, Basile Audoly, and Eitan Grinspun. 2012. Discrete viscous sheets. ACM Transactions on Graphics (TOG) 31, 4 (2012), 1–7.

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. 209–217.

Laurent Belcour and Pascal Barla. 2017. A practical extension to microfacet theory for the modeling of varying iridescence. ACM Transactions on Graphics (TOG) 36, 4 (2017), 1–14.

Mikhail Belkin and Partha Niyogi. 2008. Towards a theoretical foundation for Laplacian-based manifold methods. J. Comput. System Sci. 74, 8 (2008), 1289–1308.

Mikhail Belkin, Jian Sun, and Yusu Wang. 2009. Constructing Laplace operator from point clouds in R d. In Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 1031–1040.

Jan Bender and Dan Koschier. 2016. Divergence-free SPH for incompressible and viscous fluids. IEEE Transactions on Visualization and Computer Graphics 23, 3 (2016), 1193–1206.

Miklos Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. ACM Transactions on graphics (TOG) 29, 4 (2010), 1–10.

Robert Bridson. 2015. Fluid simulation for computer graphics. CRC press.

L. Brokshaw. 1985. A Method of Calculating Radiative Heat Diffusion in Particle Simulations. Publications of the Astronomical Society of Australia 6, 2 (1985), 207–210.

K. Chun Cheung, Leevan Ling, and Steven J. Rauh. 2015. A localized meshless method for diffusion on folded surfaces. J. Comput. Phys. 297 (2015), 194–206.

Jean-Marc Chomaz. 2001. The dynamics of a viscous soap film with soluble surfactant. Journal of Fluid Mechanics 442 (2001), 387–409.

Jens Cornelis, Markus Ihmsen, Andreas Peer, and Matthias Teschner. 2014. JISFFLP for incompressible fluids. In Computer Graphics Forum, Vol. 33. Wiley Online Library, 255–262.
Y Coudur, JM Chemaz, and M Rahard. 1989. On the hydrodynamics of soap films. *Physica D: Nonlinear Phenomena* 37, 1-3 (1989), 384–405.

Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double bubbles sans toil and trouble. Discrete circulation-preserving vortex sheets for soap films and foams. *ACM Transactions on Graphics* (TOG) 34, 4 (2015), 1–9.

C Güalon, C Denez, T Combaz, P Marmottant, and F Elias. 2017. Sound and Vision: Visualization of music with a soap film, and the physics behind it. (2017).

Robert A Gingold and Joseph J Monaghan. 1977. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the royal astronomical society* 181, 3 (1977), 375–389.

Christoph Giseler, Andreas Henn, Stefan Band, Andreas Peer, and Matthias Teschner. 2020. An implicit compressible SPH solver for snow simulation. *ACM Transactions on Graphics* (TOG) 39, 4 (2020), 36–1.

Christoph Giseler, Andreas Peer, Stefan Band, Jan Bender, and Matthias Teschner. 2019. Interlinked SPH pressure solvers for strong fluid-rigid coupling. *ACM Transactions on Graphics* (TOG) 38, 1 (2019), 1–13.

Andrew Glassner. 2000. Soap bubbles: part 2. *IEEE Annals of the History of Computing* 20, 06 (2000), 99–109.

David J Hill and Ronald D Henderson. 2016. Efficient fluid simulation on the surface of a sphere. *ACM Transactions on Graphics* (TOG) 35, 2 (2016), 1–9.

Jeong-Mo Hong and Chang-Hun Kim. 2003. Animation of bubbles in liquid. In *Computer Graphics Forum*. Vol. 22. Wiley Online Library, 253–262.

Jeong-Mo Hong and Chang-Hun Kim. 2005. Discontinuous fluids. *ACM Transactions on Graphics* (TOG) 24, 3 (2005), 915–920.

Lih Huang, Torsten Håstad, and Dominik Michels. 2019. On the accurate large-scale simulation of ferrofluids. *ACM Transactions on Graphics* (TOG) 38, 4 (2019), 1–15.

Weizhen Huang, Julian Ierringhausen, Tom Kneiphof, Ziyin Qu, Chenfanzhu Jiang, and Matthias B. Hulin. 2020. Chemomechanical Simulation of Soap Film Flow on Spherical Bubbles. *ACM Transactions on Graphics* 39, 4 (2020). https://doi.org/10.1145/3386569.3392094

Markus Ihmten, Jens Cornelis, Barbara Solenthaler, Christoph Horvath, and Matthias Teschner. 2013. Implicit incompressible SPH. *IEEE transactions on visualization and computer graphics* 20, 3 (2013), 426–435.

Sadashige Ishida, Masafumi Yamamoto, Ryoichi Ando, and Toshiya Hachisuka. 2017. *Interference colours of soap bubbles*. Dariusz Jaszkowski and Janusz Rzeszut. 2003. Interference colours of soap bubbles. *Archives of computational methods in engineering* 10, 2 (2003), 159–185.

Robert I Saye and James A Sethian. 2013. Multiscale modeling of membrane rearrangement, drainage, and rupture in evolving foams. *J. Comput. Phys.* 290, 29–57.

Robert Saye. 2014. High-order methods for computing distances to implicitly defined surfaces. *Communications in Applied Mathematics and Computational Science* 9, 1 (2014), 107–141.

Robert Saye and James A Sethian. 2015. Multiscale modeling of membrane rearrangement, drainage, and rupture in evolving foams. *Science* 340, 6133 (2013), 720–724.

Hagit Schechter and Robert Bridson. 2012. *Ghost SPH for animating water*. ACM Transactions on Graphics (TOG) 31, 4 (2012), 1–8.

Saket Patkar, Mridul Aanjaneya, Dmitriy Karpman, and Ronald Fedkiw. 2013. A hybrid Lagrangian-Eulerian formulation for bubble generation and dynamics. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 105–114.

Andreas Peer, Christoph Giseler, Stefan Band, and Matthias Teschner. 2018. An implicit SPH formulation for incompressible linearly elastic solids. In *Computer Graphics Forum.* Vol. 37. Wiley Online Library, 135–148.

Andreas Peer, Markus Ihmten, Jens Cornelis, and Matthias Teschner. 2015. An implicit viscosity formulation for SPH fluids. *ACM Transactions on Graphics* (TOG) 34, 4 (2015), 1–10.

Fabiano Petronetto, Afonso Paiva, Elias S Helou, DE Stewart, and Luis Gustavo Nonato. 2013. Mesh-Free Discrete Laplace–Beltrami Operator. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 214–226.

Daniel J Price. 2013. Smoothed particle hydrodynamics and magnetohydrodynamics. *J. Comput. Phys.* 231, 3 (2012), 759–794.

Steven J Routh and Barry Merriman. 2008. A simple embedding method for solving partial differential equations on surfaces. *J. Comput. Phys.* 227, 3 (2008), 1943–1961.

Robert Saye. 2014. High-order methods for computing distances to implicitly defined surfaces. *Communications in Applied Mathematics and Computational Science* 9, 1 (2014), 107–141.

Robert Saye and James A Sethian. 2015. Multiscale modeling of membrane rearrangement, drainage, and rupture in evolving foams. *Science* 340, 6133 (2013), 720–724.

Hagit Schechter and Robert Bridson. 2012. Ghost SPH for animating water. *ACM Transactions on Graphics* (TOG) 31, 4 (2012), 1–8.

Frank Schmidt. 2014. The laplace-beltrami-operator on riemannian manifolds. In *Seminar Shape Analysis*. Andrew Selle, Nick Rasmussen, and Ronald Fedkiw. 2005. A vortex particle method for smoke, water and explosions. In *ACM SIGGRAPH 2005 Papers*. 910–914.

Brian E Smits and Gary W Meyer. 1992. *Newton’s colors: simulating interference phenomena in real image synthesis*. In *Photorealism in Computer Graphics*. Springer, 185–194.

Barbara Solenthaler, Peter Bucher, Nuttapong Chentanez, Matthias Müller, and Markus Gross. 2011. SPH-based shallow water simulation. (2011).

Barbara Solenthaler and Renato Pajarola. 2008. Density contrast SPH interfaces. (2008).

Barbara Solenthaler and Renato Pajarola. 2009. Predictive-corrective incompressible SPH. In *ACM SIGGRAPH 2009 papers*. 1–6.

Jos Stam. 2003. Flows on surfaces of arbitrary topology. *ACM Transactions On Graphics* (TOG) 22, 3 (2003), 724–731.

Alexandre M Tartakovsky and Paul Meakin. 2005. A smoothed particle hydrodynamics model for miscible flow in three-dimensional fractures and the two-dimensional Rayleigh–Taylor instability. *J. Comput. Phys.* 207, 2 (2005), 610–624.

Sasan Tavakkol, Amir Reza Zarrati, and Mahdiyar Khanpour. 2017. Curvilinear smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids* 84, 2 (2017), 115–131.

Orestis Vantzos, Saar Raz, and Mirela Ben-Chen. 2018. Real-time viscous thin films. *ACM Transactions on Graphics* (TOG) 37, 6 (2018), 1–10.

Hui Wang, Yongxu Jin, Anqi Luo, Xubo Yang, and Bo Zhu. 2020. Codimensional surface tension flow using moving least-squares particles. *ACM Transactions on Graphics* (TOG) 39, 4 (2020), 42–1.

Jian Jun Xu, Zhilin Li, John Lowengrub, and Hongkai Zhao. 2006. A model for soap film dynamics with evolving thickness. *J. Comput. Phys.* 227, 3 (2008), 1943–1961.

Andreas Peer, Christoph Giseler, Stefan Band, and Matthias Teschner. 2018. An implicit SPH formulation for incompressible linearly elastic solids. In *Computer Graphics Forum*. Vol. 37. Wiley Online Library, 135–148.

Andreas Peer, Markus Ihmten, Jens Cornelis, and Matthias Teschner. 2015. An implicit viscosity formulation for SPH fluids. *ACM Transactions on Graphics* (TOG) 34, 4 (2015), 1–10.

M Omang, Steinar Børve, and Jan Trulsen. 2006. SPH in spherical and cylindrical coordinates. *J. Comput. Phys.* 213, 1 (2006), 391–412.

Orestis Vantzos, Saar Raz, and Mirela Ben-Chen. 2018. Real-time viscous thin films. *ACM Transactions on Graphics* (TOG) 37, 6 (2018), 1–10.

Frank Losasso, Jerry Talton, Nipun Kwatra, and Ronald Fedkiw. 2008. Two-way coupled SPH and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (2008), 797–804.

Joe J Monaghan. 1992. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics* 30, 1 (1992), 543–574.

Joseph J Monaghan and John C Lattanzio. 1985. A refined particle method for astrophysical problems. *Astronomy and astrophysics* 149 (1985), 135–143.

Dieter Morgenroth, Stefan Reinhardt, Daniel Weiskopf, and Bernhard Eberhardt. 2020. Efficient photorealistic rendering of gradient fields. In *Computer Graphics Forum*. Vol. 39. Wiley Online Library, 27–38.

Joseph P Morris, Patrick J Fox, and Yi Zhu. 1997. Modeling low Reynolds number incompressible flows using SPH. *J. Comput. Phys.* 136, 1 (1997), 214–226.
The normal and tangential components of (6) are

\[
\begin{align*}
\rho - \rho_a + 2\kappa_z \gamma &= 2\mu \nabla \cdot (\nabla u) \cdot n^z + O(h^2), \\
\tau_x \cdot (\nabla s \cdot y) &= \mu \nabla \cdot (\nabla u + (\nabla u)^T) \cdot n^z + O(h^2), \\
\tau_y \cdot (\nabla s \cdot y) &= \mu \nabla \cdot (\nabla u + (\nabla u)^T) \cdot n^z + O(h^2),
\end{align*}
\]

respectively. Using (3), we obtain

\[
\begin{align*}
t_x \cdot (\nabla s \cdot y) &= (e_x + e_y \times \nabla s) \cdot (\nabla s \cdot y) = 0 + e_z \nabla s \cdot h \cdot \nabla y + O(h^2), \\
t_y \cdot (\nabla s \cdot y) &= (e_x + e_y \times \nabla s) \cdot (\nabla s \cdot y) = 0 + e_z \nabla s \cdot h \cdot \nabla y + O(h^2),
\end{align*}
\]

Substituting (3) and (23) into (22) yields

\[
\begin{align*}
\rho - \rho_a + 2(\pm \kappa_z + \kappa_h) \gamma &= 2\mu (e_x \times e_y) \cdot (\nabla u) \cdot (\pm e_z - \nabla s \cdot h) + O(h^2), \\
e_x \cdot (\nabla s \cdot y) &= \mu (e_x \times e_y) \cdot (\nabla u + (\nabla u)^T) \cdot (\pm e_z - \nabla s \cdot h) + O(h^2), \\
e_y \cdot (\nabla s \cdot y) &= \mu (e_x \times e_y) \cdot (\nabla u + (\nabla u)^T) \cdot (\pm e_z - \nabla s \cdot h) + O(h^2).
\end{align*}
\]

Further, we substitute the component-wise forms of \(\nabla s \cdot h\) and \(\nabla u\) into (24) as

\[
\begin{align*}
\rho - \rho_a + 2(\pm \kappa_z + \kappa_h) \gamma &= 2\mu \left( \frac{\partial \nu}{\partial z} + \frac{\partial \mu}{\partial x} + \frac{\partial \mu}{\partial y} \right) + O(h^2), \\
\frac{\partial \nu}{\partial x} &= \mu \left( \frac{\partial \nu}{\partial x} - 2 \frac{\partial \mu}{\partial x} \right) + \frac{\partial \mu}{\partial y} \pm \frac{\partial \mu}{\partial y} + O(h^2), \\
\frac{\partial \nu}{\partial y} &= \mu \left( \frac{\partial \nu}{\partial y} - 2 \frac{\partial \mu}{\partial y} \right) + \frac{\partial \mu}{\partial x} \pm \frac{\partial \mu}{\partial x} + O(h^2).
\end{align*}
\]

Suppose the tangential velocity is even along the local z-coordinate. Similar to the idea of asymptotic expansion [Chomaz 2001], we take the Taylor expansion of \(u_z\) as

\[
\begin{align*}
\nu = & \nu(z = 0) + \frac{\partial^2 \nu}{\partial z^2} \left( \frac{\rho u}{\partial z} \right) z^2 + O(h^2), \\
\nu = & \nu(z = 0) + \frac{\partial^2 \nu}{\partial z^2} \left( \frac{\rho u}{\partial z} \right) z^2 + O(h^2),
\end{align*}
\]

and with the solenoidal condition, we obtain

\[
\frac{\partial \nu}{\partial z} = -\nabla \cdot u_z = -\nabla \cdot u_z(z = 0) = O(h^2).
\]

Combining (26) and (27), as well as using the boundary conditions (25), we obtain

\[
\begin{align*}
\rho \frac{D u}{D t} + O(h^2) &= 2 \nabla \cdot (\kappa h \nabla s \cdot u_z) + 2 \frac{\gamma \kappa h}{h} n + \frac{1}{h} \nabla^2 s + \mu \nabla^2 u + f.
\end{align*}
\]

### B NUMERICAL VALIDATION AND PERFORMANCE TABLE

To evaluate the accuracy of our algorithm, we perform a set of numerical tests concerning the main key elements in our algorithm in both the geometric and dynamic computations. We will compare our simulation results with both the ones derived from analytic equations and real-life experiments.

**Mean Curvature.** The centerpiece of computing the surface tension behaviors is the estimation of mean curvature \(\kappa\), which relates to the pressure jump across the air-fluid interface via the Young-Laplace equation \(\Delta p = -\gamma k\).
Table 2. Performances of Different Scenes. A in table is a 128-core 3.1GHz AMD Ryzen Threadripper 3990X workstation, B in table is a 4-core 2.8GHz Intel(R) Core(TM) laptop, and C is a 6-core 2.6GHz Intel(R) Core(TM) laptop.

| Figure  | Description                          | Number of Particles   | Computational Resource                             | FPS† | Time / Frame (Avg.) |
|---------|--------------------------------------|-----------------------|----------------------------------------------------|------|---------------------|
| Figure 11 | Thickness Profile                    | 5,154–5,785†          | Laptop C with 6 Cores                              | 10   | 1.409s              |
| Figure 4 (Upper) | Oscillating Bubble (Sphere Bubble)         | 163,842               | Server A with 16 Cores                            | 50   | 10.65s              |
| Figure 4 (Lower) | Oscillating Bubble (Irregular Bubble)     | 24,578                | Server A with 128 Cores                           | 50   | 0.225s              |
| Figure 7  | Oscillating Bubble (Half Bubble)        | 86,013                | Server A with 32 Cores                            | 50   | 9.6s                |
| Figure 5  | Bubble Rupture                        | 163,842               | Server A with 128 Cores                           | 50   | 10.92s              |
| Figure 8  | Large-Deforming Thin Films (Square)      | 58,081                | Server A with 128 Cores                           | 5    | 15.48               |
| Figure 9  | Large-Deforming Thin Films (Circular)    | 126,282               | Server A with 128 Cores                           | 10   | 13.575s             |
| Figure 14 | Thin-Film Dripping                    | 5,185–5,501†          | Server A with 6 Cores                             | 50   | 9.74s               |
| Figure 13,17 | Catenoid                               | 4,968–5,542†          | Laptop C with 6 Cores                            | 300  | 0.883s              |
| Figure 15 | Droplet Marangoni Effect               | 8,173                 | Laptop C with 6 Cores                             | 50   | 2.01s               |
| Figure 16 | Mean Curvature Validation              | 10,000                | Laptop C with 6 Cores                             | 10   | 0.35s               |
| Figure 19 | Numerical Height Validation            | 5,154                 | Laptop C with 6 Cores                             | 10   | 0.774s              |
| Figure 18 | Capillary Wave Validation              | 1,000                 | Laptop B with 4 Cores                             | 50   | 0.02s               |

† The number of particles increases throughout the simulation due to particle reseeding, the two numbers listed in the table are the number of particles at the first and last frame.
† The number of frames per second for simulation. For example, FPS=50 means the time step of a frame Δt = 0.02s. We take CFL condition number C = 0.1, so a frame may consist of multiple time iterations depending on Δtmax, and the simulation time of a frame is subject to it.

with $c = 0.8, a = 0.3, u \in \{0, 2\pi\}, v \in \{0, 2\pi\}$, and a surface defined by $y = 0.1 \left(3 \sin(x) + 2 \cos(z) + 4 \sin(2x + z)\right)$ with $x, z \in \{0, 2\pi\}$.

To highlight the robustness of our algorithm, we initialize the sample particles from a uniform random distribution to rehearse the scenarios where particles are unevenly distributed in simulation. In Figure 16, we plot the numerical and analytical estimations of mean curvature on the left two columns and the error between them on the right column. We can see that overall, our algorithm does well in estimating the mean curvature for all kinds of surfaces. The error is mostly gathered on the edges due to insufficiently sampled neighborhoods, a shortcoming common to SPH methods [Koschier et al. 2019].

**Minimal Surface.** The dynamics of thin-film are marked by its tendency to form minimal surfaces under the surface tension, which constantly contracts the surface to minimize the area locally. When two rings are connected by a continuous soap film, and gravity is ignored, the soap film between them will form the catenoid —— a minimal surface formed by rotating a catenary. The analytical definition of a catenoid is given by

$$(x, y, z) = \left(c \cosh \left(\frac{u}{c}\right) \cos u, c \cosh \left(\frac{u}{c}\right) \sin u\right).$$

The way that soap-film catenoids are typically formed is by gradually pulling the parallel rims apart. If we let $D$ denote the diameter of the rims, and let $d$ denote the separation between them, then an analytical catenoid can be solved for when $\frac{D}{d} < 0.66274$.

The constant $0.66274$ is the approximation of the Laplace limit, which is the value $\frac{1}{\sinh u}$ for $u$ satisfying $u = \coth u, u > 0$. As a result, until this critical condition is met, we can obtain a corresponding analytical catenoid to compare and contrast for every point cloud in our simulation.
In the upper subfigures in Figure 17, we plot the result of our simulation along with the photographed real-world soap catenoid. On the rightmost one, we overlay our simulated results (orange) with the corresponding analytical catenoid(blue). As we can see, our simulation is only marginally different from the analytical solution, and the difference between our simulation and the photography is almost imperceptible. The bottom-left displays a close-up look of a curve from a cross-section of our simulated shape (orange) and an analytical catenary. The bottom-right displays two subplots that are time-varying: the upper one records the varying average mean curvature, which is supposed to be 0 as it is for all minimal surfaces, and the lower one is an average relative positional error, which represents the average distance between a particle and the analytical catenoid relative to the arc length of the catenary. As one can see, both plots illustrate a decreasing trend marked by periodic fluctuation, which corresponds to the oscillating motion of the simulated thin film; until the Laplace limit is violated, the simulated thin film does well in keeping close to the theoretical shape with the relative positional error less than 0.06.

**Numerical Height vs. Advectced Height.** We investigate qualitatively and quantitatively the behavioral discrepancy between the two forms of height computation. In the left subfigure of Figure 19, we exhibit the surface defined by the advected height (orange) and by the numerical height (blue) at frames [1, 3, 10, 22, 32, 100] of a simulation where a centripetal velocity is initially applied to move particles towards the middle, while the surface tension works to flatten out the curvature. The two surfaces are congruent in trend, and yet the surface formed by the advected height is bumpier than the other. In the top-right subfigure, the difference between the two surfaces is plotted alongside the surfaces to show that their commonality dominates their difference. In the bottom-right subfigure, we show that the change in the average percent difference over time is confined within 5%.

**Capillary Wave.** Figure 18 shows the capillary wave on a bubble. The scene setup is a simplification of that in Figure 4, where initially one downward impulse, instead of two, is applied on the north pole of the bubble. It immediately creates a perturbation at the top and then propagated to the bottom. This propagation is dominated by the capillary force \( \frac{2\pi\kappa}{\gamma} \), and thus a capillary wave. Note that \( \kappa \) is calculated by half of the Laplacian of surface, then the local dynamics along normal direction system subjects to wave equation

\[
\rho \frac{D^2 r_z}{Dt^2} + \frac{\gamma}{h} \frac{\partial^2 r_z}{\partial x^2} = 0
\]

with \( r_z \) as the local z-component of particle position. Thus the phase velocity of the wave is analytically given by \( v_p = \omega / k = \sqrt{\gamma / \rho} \).

The right part of Figure 18 shows the time-distance relationship traveled by the down-most wave knot, which represents the phase velocity of the capillary wave, comparing analytical solution and simulation results. It proves that our algorithm can accurately and convincingly reproduce the effect surface tension.