Thin-Film Smoothed Particle Hydrodynamics Fluid

MENGDI WANG, Dartmouth College
YITONG DENG, Dartmouth College
XIANGXIN KONG, Dartmouth College
ADITYA H. PRASAD, Dartmouth College
SHIYING XIONG, Dartmouth College
BO ZHU, Dartmouth College

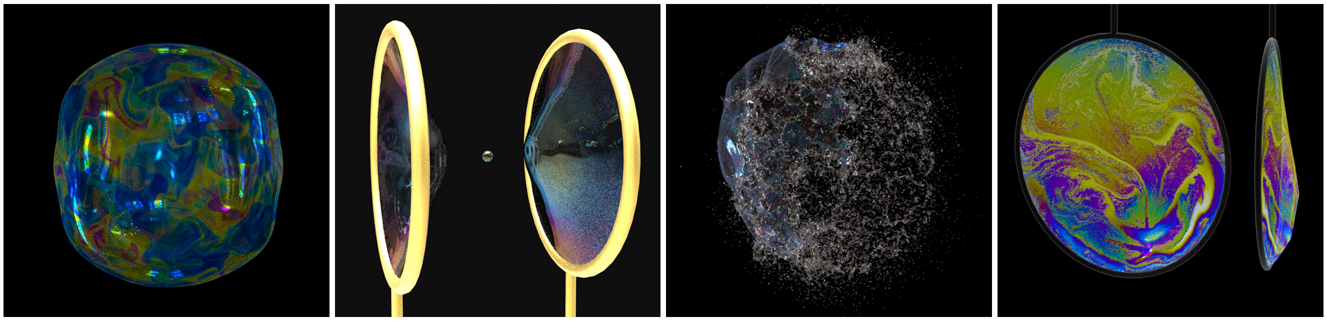

Fig. 1. Several thin-film phenomena as simulated using our proposed method. Counting from left to right: (1) surface flow on a oscillating soap bubble, (2) pinched-off droplets between two circular rims pulling away from one another, (3) a soap bubble bursting into tiny droplets and filaments after being 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

ACM Reference Format:
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 (August 2021), 16 pages. https://doi.org/10.1145/3450626.3459864

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

Authors' addresses: Mengdi Wang, Dartmouth College, mengdi.wang.gr@dartmouth.edu; Yitong Deng, Dartmouth College, yitong.deng.gr@dartmouth.edu; Xiangxin Kong, Dartmouth College, xiangxin.kong.gr@dartmouth.edu; Aditya H. Prasad, Dartmouth College, aditya.h.prasad.21@dartmouth.edu; Shiying Xiong, Dartmouth College, shiying.xiong@dartmouth.edu; Bo Zhu, Dartmouth College, bo.zhu@dartmouth.edu.

0730-0301/2021/8-ART110 $15.00
https://doi.org/10.1145/3450626.3459864

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 thickness 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-dimension 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 multiphase fluid system, where bubbles are surrounded by large bulks of water, *i.e.*, foams [Aanjaneya 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 and 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 [Omang 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.

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
ρ	density of fluid in soap film
ξ^1, ξ^2, z	local coordinate variables
$\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{n}$	basis vectors of local frame
$\boldsymbol{u}$	velocity of fluid
γ	surface tension coefficient
Γ	surfactant concentration
u, v	tangential components of velocities on soap film
w	normal component of velocities on soap film
κ_c	the curvature of center surface
κ_h	the local curvature of interfaces
α_c	diffusion coefficient of surfactant
$\alpha_h, \alpha_k, \alpha_d$	pressure coefficients

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

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 classic 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 our codimension-1 differential operators. 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 $S_I^\pm$, and central surface S_C. We build a surface coordinate

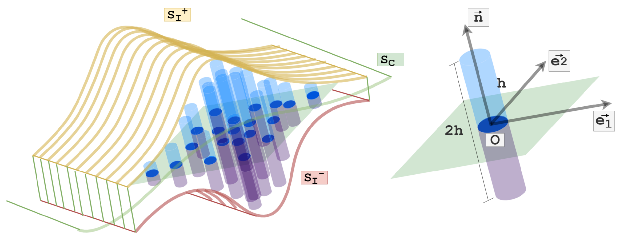

Fig. 2. The schematic diagram of the soap film's internal structure. Central surface S_C is enveloped by two interfaces $S_I^{\pm}$. For point O on S_C, the coordinate system is established with normal axis $\boldsymbol{n}$ and tangential axes $(\boldsymbol{e}_1, \boldsymbol{e}_2)$. The thickness of film is $2h$ along $\boldsymbol{n}$.

system $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ near point O on the central surface S_C, with a corresponding normal vector $\boldsymbol{n}$, making the coordinates (ξ^1, ξ^2, z) near point O. Since we assume that the interfaces $S_I^{\pm}$ 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 $\boldsymbol{v} = v^1\boldsymbol{e}_1 + v^2\boldsymbol{e}_2$ and the Laplacian of a scalar field s as

$$\begin{cases} \boldsymbol{\nabla}_s s = \displaystyle\sum_{k=1}^{2}\sum_{l=1}^{2} g^{kl}\frac{\partial s}{\partial \xi^l}\boldsymbol{e}_k, \\ \boldsymbol{\nabla}_s \cdot \boldsymbol{v} = \dfrac{1}{\sqrt{g}}\displaystyle\sum_{k=1}^{2}\frac{\partial}{\partial \xi^k}\left(\sqrt{g}v^k\right), \\ \nabla_s^2 s = \dfrac{1}{\sqrt{g}}\displaystyle\sum_{k=1}^{2}\sum_{l=1}^{2}\frac{\partial}{\partial \xi^k}\left(\sqrt{g}g^{kl}\frac{\partial s}{\partial \xi^l}\right), \end{cases} \tag{1}$$

where

$$g^{kl} = \frac{\partial \boldsymbol{x}}{\partial \xi^k}\frac{\partial \boldsymbol{x}}{\partial \xi^l} \tag{2}$$

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

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 $S_I^{\pm}$ can be approximated as

$$\begin{cases} \boldsymbol{n}^{\pm} = \dfrac{\pm\boldsymbol{n} - \boldsymbol{\nabla}_s h}{|\pm\boldsymbol{n} - \boldsymbol{\nabla}_s h|} = \pm\boldsymbol{n} - \boldsymbol{\nabla}_s h + O(h^2), \\ \boldsymbol{t}_1^{\pm} = \dfrac{\boldsymbol{e}_1 \mp \boldsymbol{e}_2 \times \boldsymbol{\nabla}_s h}{|\boldsymbol{e}_1 \mp \boldsymbol{e}_2 \times \boldsymbol{\nabla}_s h|} = \boldsymbol{e}_1 \mp \boldsymbol{e}_2 \times \boldsymbol{\nabla}_s h + (h^2), \\ \boldsymbol{t}_2^{\pm} = \dfrac{\boldsymbol{e}_2 \pm \boldsymbol{e}_1 \times \boldsymbol{\nabla}_s h}{|\boldsymbol{e}_2 \pm \boldsymbol{e}_1 \times \boldsymbol{\nabla}_s h|} = \boldsymbol{e}_2 \pm \boldsymbol{e}_1 \times \boldsymbol{\nabla}_s h + O(h^2), \\ \kappa^{\pm} = -\dfrac{1}{2}\boldsymbol{\nabla} \cdot \boldsymbol{n}^{\pm} = \pm\kappa_c + \kappa_h + O(h^2), \end{cases} \tag{3}$$

where $\boldsymbol{\nabla}$ is the three dimensional gradient operator that satisfies $\boldsymbol{\nabla}_s = \boldsymbol{\nabla} - \boldsymbol{n}(\boldsymbol{n} \cdot \boldsymbol{\nabla})$. The curvature of the center surface is given by $\kappa_c = 0.5\boldsymbol{\nabla}_s^2 S_C$, and the local curvature caused by the surface thickness is defined as $\kappa_h = 0.5\nabla_s^2 h$. In addition, the gradient operator on the upper and lower surfaces $S_I^{\pm}$ can be approximated as

$$\boldsymbol{\nabla}_{s^{\pm}} = \boldsymbol{\nabla} - \boldsymbol{n}^{\pm}(\boldsymbol{n}^{\pm} \cdot \boldsymbol{\nabla}) = \boldsymbol{\nabla}_s \pm \boldsymbol{\nabla}_s h(\boldsymbol{n} \cdot \boldsymbol{\nabla}) \pm \boldsymbol{n}(\boldsymbol{\nabla}_s h \cdot \boldsymbol{\nabla}) + O(h^2). \tag{4}$$

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 $\boldsymbol{u}$ within $-h < z < h$ satisfies 3D incompressible Navier–Stokes equations:

$$\begin{cases} \rho\left(\dfrac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla}\boldsymbol{u}\right) = -\boldsymbol{\nabla}p + \mu\nabla^2\boldsymbol{u} + \boldsymbol{f}, \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \end{cases} \quad -h < z < h. \tag{5}$$

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

$$(p - p_a + 2\kappa^{\pm}\gamma)\boldsymbol{n}^{\pm} = -\boldsymbol{\nabla}_{s^{\pm}}\gamma + \mu[\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^T] \cdot \boldsymbol{n}^{\pm}, \ z = \pm h, \tag{6}$$

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 $z = 0$ as:

$$\begin{aligned} &\rho\frac{D\boldsymbol{u}}{Dt} + O(h^2) = \\ &\quad 2\boldsymbol{\nabla}_s(\kappa_h\gamma + \boldsymbol{\nabla}_s \cdot \boldsymbol{u}_s) + \frac{2\gamma}{h}\kappa_c\boldsymbol{n} + \frac{1}{h}\boldsymbol{\nabla}_s\gamma + \mu\nabla_s^2\boldsymbol{u} + \boldsymbol{f}. \end{aligned} \tag{7}$$

Here $\boldsymbol{u}_s = \boldsymbol{u} - (\boldsymbol{n} \cdot \boldsymbol{u})\boldsymbol{n}$ 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

$$\frac{Dh}{Dt} = -(\boldsymbol{\nabla}_s \cdot \boldsymbol{u}_s)h. \tag{8}$$

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 Γ and the thin film's surface tension coefficient γ can be described by the following *Langmuir equation of state*: $\gamma(\Gamma) = \gamma_0 + RT\Gamma_\infty \log\left(1 - \frac{\Gamma}{\Gamma_\infty}\right)$ [Muradoglu and Tryggvason 2008], where γ_0 is the surface tension coefficient of an untainted interface ($\Gamma = 0$), Γ_∞ 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 $\Gamma \ll \Gamma_0$, The *Langmuir equation of state* can be approximated by $\gamma(\Gamma) = \gamma_0 - \gamma_a\Gamma$ with $\gamma_a = RT$ [Xu et al. 2006]. We further assume that the surfactant concentration Γ is invariant with the normal coordinate and it satisfies the classical convection–diffusion equation with the diffusion coefficient α_c as:

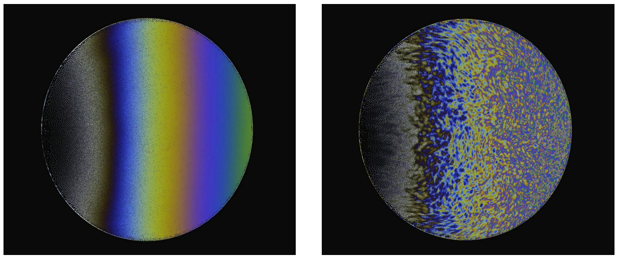

Fig. 3. The comparison between numerical height (left) and advected height (right).

$$\frac{\mathrm{D}\Gamma}{\mathrm{D}t} = \alpha_c \nabla_s^2 \Gamma. \tag{9}$$

Discarding the quantities of the magnitude $O(h^2)$ and combining (7), (8), and (9), we arrive at the our functional dynamics model:

$$\begin{cases} \rho \dfrac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = 2\boldsymbol{\nabla}_s(\kappa_h \gamma + \boldsymbol{\nabla}_s \cdot \boldsymbol{u}_s) + \dfrac{2\gamma}{h}\kappa_c \boldsymbol{n} + \dfrac{1}{h}\boldsymbol{\nabla}_s \gamma + \mu \nabla_s^2 \boldsymbol{u} + \boldsymbol{f}, \\ \dfrac{Dh}{Dt} = -(\boldsymbol{\nabla}_s \cdot \boldsymbol{u}_s)h, \\ \dfrac{\mathrm{D}\Gamma}{\mathrm{D}t} = \alpha_c \nabla_s^2 \Gamma. \end{cases} \tag{10}$$

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 $A_i = \sum_j s_j A_j W_{ij} = \sum_j \frac{V_j}{h_j} A_j W_{ij}$, where s_j 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:

$$h_i = \sum_j V_j W_{ij}. \tag{11}$$

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, 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),

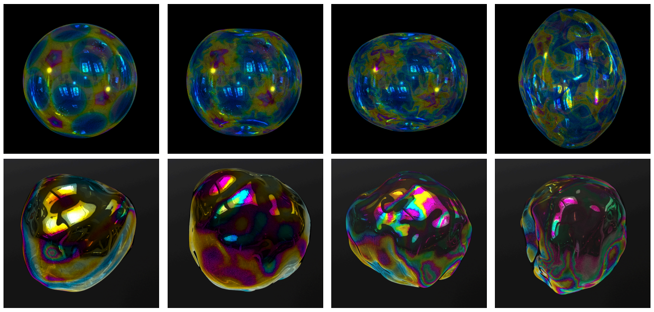

Fig. 4. Bubble oscillation: A perfect sphere ocillates under perturbations and vortices (upper part), and an irregular bubble (lower part).

$$\frac{\mathrm{D}\hat{h}}{\mathrm{D}t} = -\left(\boldsymbol{\nabla}_s \cdot \boldsymbol{u}_s\right)\hat{h}, \tag{12}$$

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.

6.2 Differential Operators

In our scheme, the discretization of surface differential operators, *i.e.*, $\boldsymbol{\nabla}_s$,$(\boldsymbol{\nabla}_s \cdot)$ and ∇_s^2 at particle i are obtained by projecting all of its neighbor particles to the plane defined by its local frame $\boldsymbol{e}^i$, and performing traditional SPH operators on that codimension-1 plane, as shown in (13):

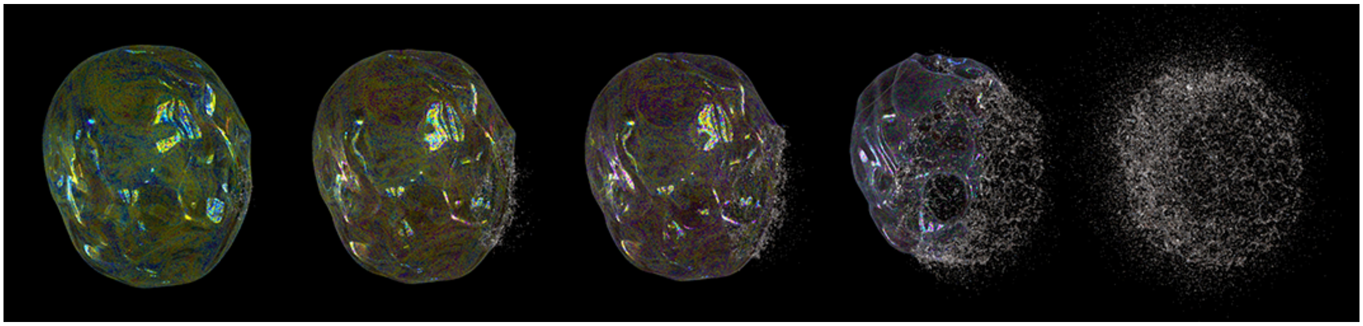

Fig. 5. Bubble rupture: The bubble bursts into filaments and droplets after being "punctured" in a small region on the right. The color change on the bubble comes as a result of the surface contraction, which increases the thickness of the thin film.

$$\begin{cases} (\nabla_s f)_i = \sum_j h_i V_j \left(\frac{f_i}{h_i^2} + \frac{f_j}{h_j^2} \right) \nabla_s W_{ij}, (\textit{symmetric form}) \\ (\nabla_s f)_i = \sum_j \frac{V_j}{h_j} (f_j - f_i) \nabla_s W_{ij}, (\textit{difference form}) \\ (\nabla_s \cdot \boldsymbol{u}_s)_i = \sum_j \frac{V_j}{h_j} \widetilde{\boldsymbol{u}}_{ij} \cdot \nabla_s W_{ij}, \\ \left(\nabla_s^2 f \right)_i = \sum_j \frac{V_j}{h_j} (f_j - f_i) \frac{2|\nabla_s W_{ij}|}{|\widetilde{\boldsymbol{r}}_{ij}|}. \end{cases} \tag{13}$$

Distinguishing between the *symmetric* and *difference* forms of the surface gradient operator is a common numerical technique employed in SPH fluid simulations [Koschier et al. 2019; Müller et al. 2003]. In our discretization, we primarily use the symmetric form to take advantage of its momentum conserving nature, with two exceptions. One is the Marangoni force, where we want to highlight the directional quality of the motion. The other is when treating particles near the boundary to negate their tendencies to be pulled towards the center due to the lack of particles on the other side.

The surface vector $\widetilde{\boldsymbol{r}}_{ij}$ in (13) is obtained by subtracting the local normal component $\boldsymbol{n}_i$ from the vector $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$ pointing from particle j to i as $\widetilde{\boldsymbol{r}}_{ij} = \boldsymbol{r}_{ij} - (\boldsymbol{r}_{ij} \cdot \boldsymbol{n}_i)\, \boldsymbol{n}_i$.

We further define the surface gradient $\nabla_s W_{ij}$ of a codimension-1 kernel scalar kernel function $W = W(\widetilde{r})$ as

$$\nabla_s W_{ij} = g_i^{-1} \frac{\mathrm{d}W}{\mathrm{d}\widetilde{r}} \frac{\widetilde{\boldsymbol{r}}_{ij}}{|\widetilde{\boldsymbol{r}}_{ij}|}. \tag{14}$$

Here g_i is the metric tensor at particle i and $\widetilde{r} = |\widetilde{\boldsymbol{r}}_{ij}|$. We use the fourth-order spline function [Tartakovsky and Meakin 2005] to compute the numerical height h_i and the Spiky kernel [Müller et al. 2003] to compute all other terms to avoid clustering caused by inappropriate repulsive forces. Mathematically, these kernel functions are all in 2D forms because they're operating on a codimension-1 plane.

The velocity difference $\widetilde{\boldsymbol{u}}_{ij}$ we used for divergence operator is different from the projection of $\boldsymbol{u}_j - \boldsymbol{u}_i$ to codimension-1 plane $\boldsymbol{e}^i$. The reason arises from the insight that a streamline on the surface, with a constant velocity rate α (*i.e.*, $|\boldsymbol{u}_i| = |\boldsymbol{u}_j| = \alpha, \quad \boldsymbol{u}_i \cdot \boldsymbol{n}_i = \boldsymbol{u}_j \cdot \boldsymbol{n}_j = 0$), should yield a zero divergence. However, if we directly project $\boldsymbol{u}_j$ to the local plane at particle i, the deviation of normal directions $\boldsymbol{n}_i - \boldsymbol{n}_j \neq 0$ will incorrectly distort the project velocity rate as $|\boldsymbol{u}_j - (\boldsymbol{u}_j \cdot \boldsymbol{n}_i)\, \boldsymbol{n}_i| \neq \alpha$. 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 $(\widetilde{\boldsymbol{e}}^i, \widetilde{\boldsymbol{n}}_i)$ and $(\widetilde{\boldsymbol{e}}^j, \widetilde{\boldsymbol{n}}_j)$ at local frames of i and j, with their normal directions unchanged, and one tangential axis, say, $\boldsymbol{e}_1$, along the direction of $\boldsymbol{r}_{ij}$:

$$\begin{aligned} &\widetilde{\boldsymbol{n}}_i = \boldsymbol{n}_j, \quad \widetilde{\boldsymbol{n}}_j = \boldsymbol{n}_j, \\ &\widetilde{\boldsymbol{e}}_1^i = \widetilde{\boldsymbol{e}}_1^j = \frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|}, \\ &\widetilde{\boldsymbol{e}}_2^i = \widetilde{\boldsymbol{n}}_i \times \widetilde{\boldsymbol{e}}_1^i, \quad \widetilde{\boldsymbol{e}}_2^j = \widetilde{\boldsymbol{n}}_j \times \widetilde{\boldsymbol{e}}_1^j. \end{aligned} \tag{15}$$

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

$$\begin{aligned} \boldsymbol{u}_i &= u_i \widetilde{\boldsymbol{e}}_1^i + v_i \widetilde{\boldsymbol{e}}_2^i + w_i \widetilde{\boldsymbol{n}}_i, \\ \boldsymbol{u}_j &= u_j \widetilde{\boldsymbol{e}}_1^j + v_j \widetilde{\boldsymbol{e}}_2^j + w_j \widetilde{\boldsymbol{n}}_j. \end{aligned} \tag{16}$$

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

$$\widetilde{\boldsymbol{u}}_{ij} = (u_j - u_i)\, \widetilde{\boldsymbol{e}}_1^i + (v_j - v_i)\, \widetilde{\boldsymbol{e}}_2^i. \tag{17}$$

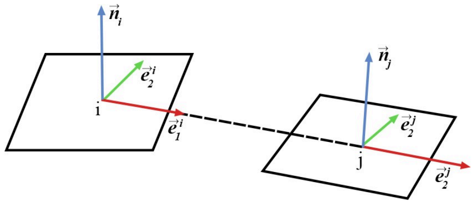

Fig. 6. Definition of $\widetilde{\boldsymbol{u}}_{ij}$ in (13). Here $\vec{n}$ represents normal axes, $\vec{e}_1$ is a a unit vector pointing from i to j, and $\vec{e}_2$ is the cross product of $\vec{n}$ and $\vec{e}_1$.

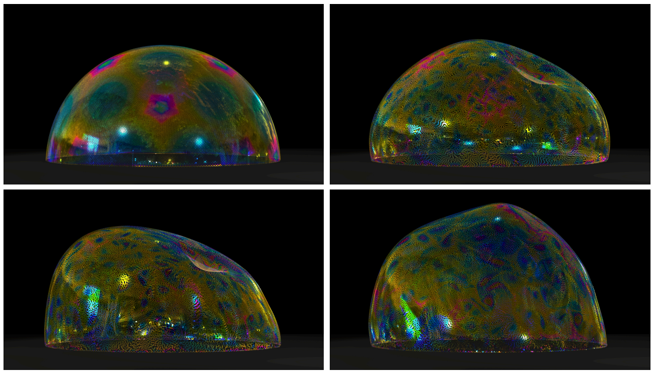

Fig. 7. Half bubble: The surface-tension-driven oscillation for a half bubble standing on a plane. In the beginning, a push toward the center is applied on the top right region of the bubble. A vorticity confinement force is employed to enhance the rotational motion on the bubble surface.

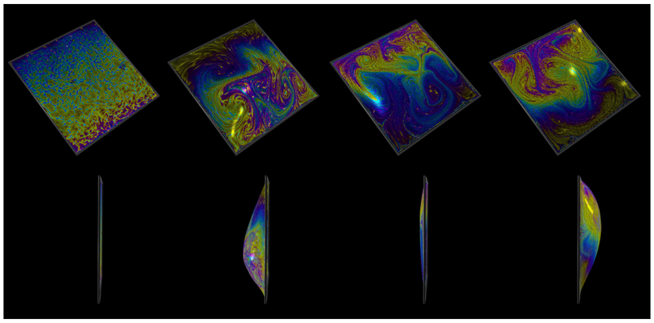

Fig. 8. Thin-film confined in a square: A gravitational pull is being rotated periodically around the surface, expediting the formation of appealing color patterns.

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

$$p_i = \alpha_p \left(\frac{\rho_i}{\rho_0} - 1 \right), \tag{18}$$

with α_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 we derived in the previous section to obtain

$$p = \kappa_h \gamma + \boldsymbol{\nabla}_s \cdot \boldsymbol{u}_s, \tag{19}$$

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

$$p_i = \alpha_h \left(\frac{h_i}{h_0} - 1 \right) + \alpha_k \gamma_i \left(\kappa_h\right)_i + \alpha_d \left(\boldsymbol{\nabla}_s \cdot \boldsymbol{u}_s \right)_i . \tag{20}$$

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 administrates 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 $\kappa_c = 0.5 \boldsymbol{\nabla}_s^2 S_C$, where $S_C : z = z(x, y)$, is calculated using the surface Laplacian operator

$$2\kappa_c = \sum_j \frac{V_j}{h_j} \left(-\boldsymbol{r}_{ij} \cdot \boldsymbol{n}_i \right) \frac{2|\boldsymbol{\nabla}_s W_{ij}|}{|\widetilde{\boldsymbol{r}}_{ij}|}. \tag{21}$$

We further introduce reviseaa 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 ζ_i on every particle i, which is subject to a similar convection-diffusion equation of the surfactant concentration (9). The vorticity confinement force $\boldsymbol{f}_i^{\zeta}$ takes the form $\boldsymbol{f}_i^{\zeta} = -\sum_j \widetilde{\boldsymbol{r}}_{ij} \times \left(\zeta_j \boldsymbol{e}_z^j \right)$.

7 TIME INTEGRATION

Before computing the forces from the SPH discretization of (10), we first calculate the surface tension coefficient γ_i 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*

```
Update surface tension: $\gamma_i = \gamma_0 - \gamma_a \Gamma_i$.
for each particle $j$ in the neighborhood of $i$ do
    Update divergence: $(\boldsymbol{\nabla}_s \cdot \boldsymbol{u}_s)_i = \sum_j \frac{V_j}{h_j} \widetilde{\boldsymbol{u}}_{ij} \cdot \boldsymbol{\nabla}_s W_{ij}$.
    Update local curvature: $(\kappa_h)_i = \sum_j \frac{V_j}{h_j} \left(h_j - h_i\right) \frac{2|\boldsymbol{\nabla}_s W_{ij}|}{|\widetilde{\boldsymbol{r}}_{ij}|}$.
end for
Update pressure:
    $p_i = \alpha_h \left( \frac{h_i}{h_0} - 1 \right) + \alpha_k \gamma_i \left(\kappa_h\right)_i + \alpha_d \left( \boldsymbol{\nabla}_s \cdot \boldsymbol{u}_s \right)_i$.
```

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 $(\kappa_c)_i > \kappa_0$ or $n_i < 2$. Here κ_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

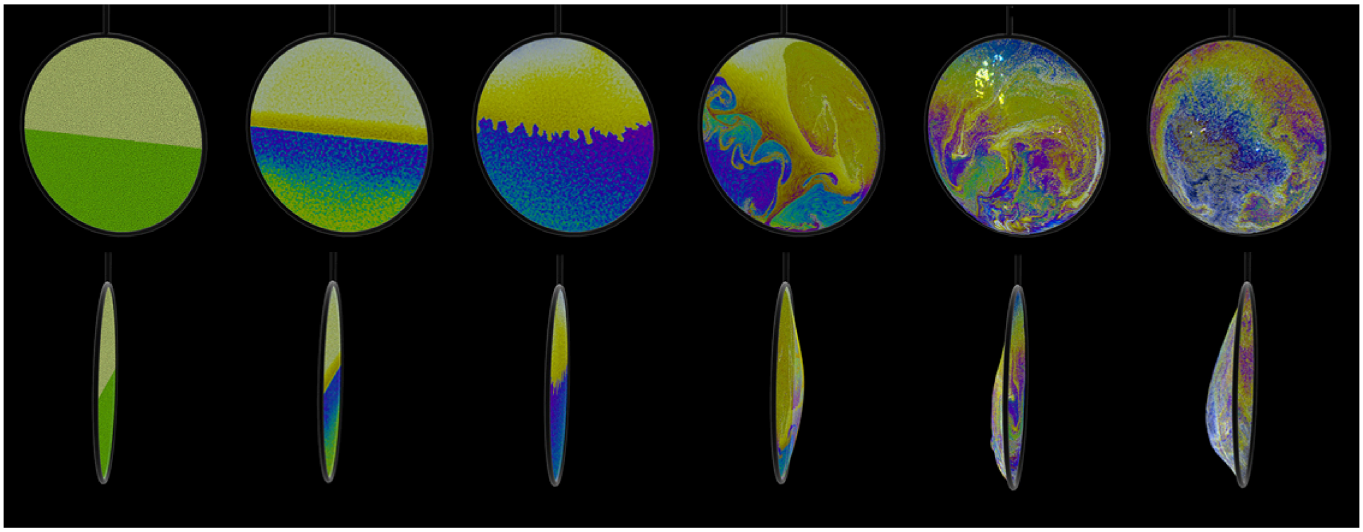

Fig. 9. Rayleigh-Taylor instability simulated on a piece of dynamically deforming thin film. Upon initialization, the upper half of the thin film carries a larger density, and the lower half carries a larger volume. The interface evolves into finger-like patterns, which further disintegrates into numerous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

Algorithm 2 Compute Forces for Particle i

```
Compute external force: $\boldsymbol{f}_i^e = m_i \boldsymbol{g} + \boldsymbol{f}_{ext}$.
for each particle $j$ in the neighborhood of $i$ do
    Compute vorticity confinement force: $\boldsymbol{f}_i^\zeta = -\sum_j \widetilde{\boldsymbol{r}}_{ij} \times (\zeta_j \boldsymbol{n}_j)$.
    Compute pressure force with symmetric form:
        $\boldsymbol{f}_i^p = 2V_i \sum_j h_i V_j \left( \frac{p_i}{h_i^2} + \frac{p_j}{h_j^2} \right) \nabla_s W_{ij}$.
    Compute Marangoni force with difference form:
        $\boldsymbol{f}_i^m = \frac{V_i}{h_i} \sum_j \frac{V_j}{h_j} (\gamma_j - \gamma_i) \nabla_s W_{ij}$.
    Compute capillary force:
        $\boldsymbol{f}_i^c = \frac{\gamma_i V_i}{h_i} \boldsymbol{n}_i \sum_j \frac{V_j}{h_j} (-\boldsymbol{r}_{ij} \cdot \boldsymbol{n}_i) \frac{2|\nabla_s W_{ij}|}{|\widetilde{\boldsymbol{r}}_{ij}|}$.
    Compute viscosity force:
        $\boldsymbol{u}_{ij} = \boldsymbol{u}_j - \boldsymbol{u}_i$,
        $\boldsymbol{f}_i^v = V_i \mu \sum_j \frac{V_j}{h_j} (\boldsymbol{u}_{ij} - (\boldsymbol{u}_{ij} \cdot \boldsymbol{n}_i) \boldsymbol{n}_i) \frac{2|\nabla_s W_{ij}|}{|\widetilde{\boldsymbol{r}}_{ij}|}$.
end for
```

Algorithm 3 Advance a Time Step

```
for each particle $i$ do
    Compute preliminary variables with Algorithm 1.
    Update advected height: $\hat{h}_i += -\hat{h}_i (\nabla_s \cdot \boldsymbol{u}_s)_i \Delta t$.
    Diffuse surfactant concentration:
        $\Gamma_i += \alpha_c \Delta t \sum_j \frac{V_j}{h_j} (\Gamma_j - \Gamma_i) \frac{2|\nabla_s W_{ij}|}{|\widetilde{\boldsymbol{r}}_{ij}|}$.
    Compute forces with Algorithm 2.
    Update velocity: $\boldsymbol{u}_i += \frac{\Delta t}{m_i} \left( \boldsymbol{f}_i^e + \boldsymbol{f}_i^\zeta + \boldsymbol{f}_i^p + \boldsymbol{f}_i^m + \boldsymbol{f}_i^c + \boldsymbol{f}_i^v \right)$.
    Update position: $\boldsymbol{r}_i += \boldsymbol{u}_i \Delta t$.
    Update local frame $\boldsymbol{e}^i$, normal $\boldsymbol{n}_i$ and metric tensor $g_i$.
    Update numerical height: $h_i = \sum_j V_j W ij$.
end for
```

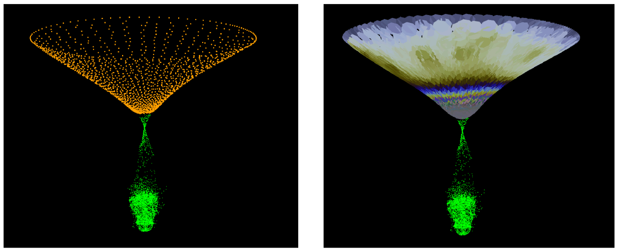

Fig. 10. An example of codimension transition. In the left subfigure, codimension-1 particles are colored with orange and codimension-0 particles with green. In the right subfigure, the surface consisted of codimension-1 particles is also displayed. One can see that at the tip of the cone formed by the 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 $(\boldsymbol{r}_j - \boldsymbol{r}_i) \cdot \boldsymbol{n}_j < threshold$, 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 $p_b V_b = \alpha_p$, where $\alpha_p = nRT$ is a constant number. The volume of closed bubble V_b is given by a sum with pyramid volume formula $V_b = \sum_i \frac{1}{3}\frac{V_i}{h_i}(\boldsymbol{n}_i \cdot \boldsymbol{x}_i)$, with $\boldsymbol{x}_i$ the vector pointing from the gravity center of all particles on the bubble to particle i. Afterwards, an external force $\boldsymbol{f}_{ext} = \frac{p_a - p_b}{2h}\boldsymbol{n}$ 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 $\boldsymbol{r}_{ij} > \textit{threshold}$, a new particle k is reseeded at $\boldsymbol{r}_k = \frac{\boldsymbol{r}_i + \boldsymbol{r}_j}{2} + \boldsymbol{\sigma}$, with a random perturbation $\boldsymbol{\sigma}$. 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 $\boldsymbol{u}_k = \frac{\boldsymbol{u}_j}{2}$. 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.

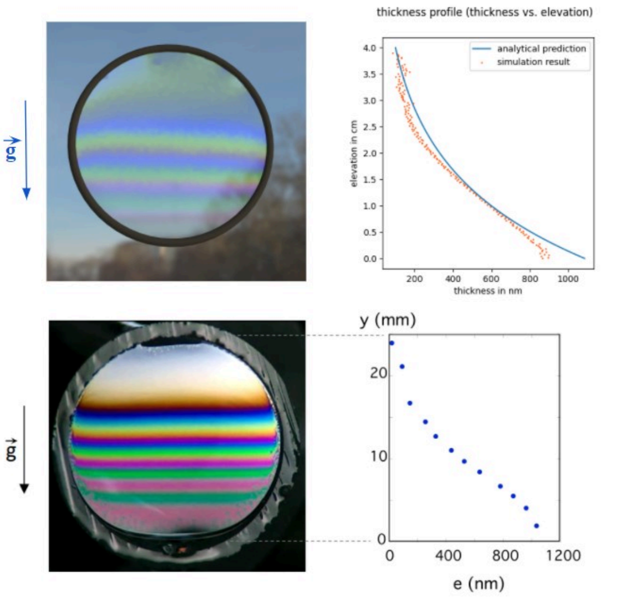

Fig. 11. The Newton Interference pattern. The upper two pictures are from our simulation and the lower two pictures shows a real-world experiment [Gaulon et al. 2017]. In the right column, elevation(horizontal) vs. thickness (vertical) graphs are displayed. Compared with the experimental result, our simulation can generate visually realistic color fringes. As shown in the right figures, the thickness profile generated by our simulation matches the one measured in the real experiment. Overlaid in the top-right figure is the analytical profile (blue curve) predicted according to Couder et al. [1989]; the proximity of it with our simulation results (yellow dots) further confirms our simulation result.

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].

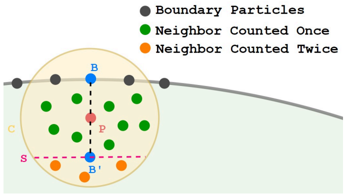

Fig. 12. Illustration of the compensation method to treat boundary.

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

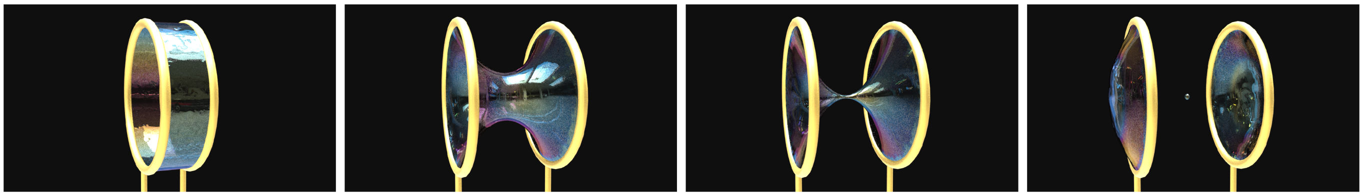

Fig. 13. Catenoid: Two parallel, circular rims are connected via thin film, under surface tension, the thin film would contract in the center and converge to the minimal surface of the catenoid. After the rim separation surpasses the Laplace limit, the catenoid can no longer be sustained by the thin film, which will eventually be separated and pinch off a small droplet.

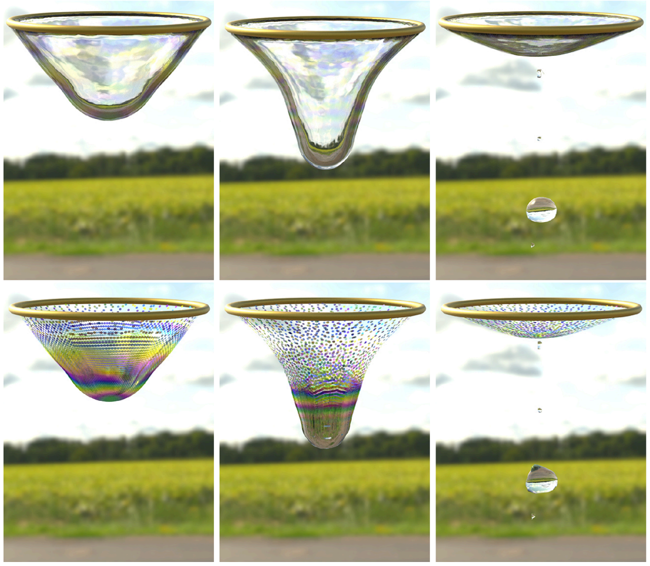

Fig. 14. Thin-film dripping, surface particles at the tip transform into volumetric particles, detaching from the thin film and falls down as spherical drips. The upper and lower subfigures depict the same simulation sequence rendered as mesh and a collection of circles, respectively.

interference for the individual wavelengths as a function of the thin film thickness along with the refractive indices of the external medium and the fluid. As light illuminates a thin film, there exists a difference in distance traveled between the reflected lightwaves and the refracted lightwaves. The shift of phase can cause them to interfere constructively or destructively travel, with the exactitude dependent on the exact thickness of the film. The resultant light intensities for all wavelengths will be converted to a single RGB color value by integrating with the CIE matching functions. There exists various open-source software that performs such tasks; we rely on ColorPy [Kness 2008] in our implementation. We utilize OpenGL for generating experimental visualizations and Houdini for rendering photorealistic images and videos. In Houdini, we render the particles both as circles and as mesh. Circles can highlight the intricate flow patterns, and the mesh is used for better realism. The physically-based spectral color calculated in the abovementioned way will serve as the base color of the particles, and additional 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 η^* and σ^* represents the surface's equilibrium thickness and surface tension coefficient when horizontally placed. Our experiments use $\eta^* = 387.875nm$ and $\sigma^* = 71.996 \cdot 10^{-6} N/m$. 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 and 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 at 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

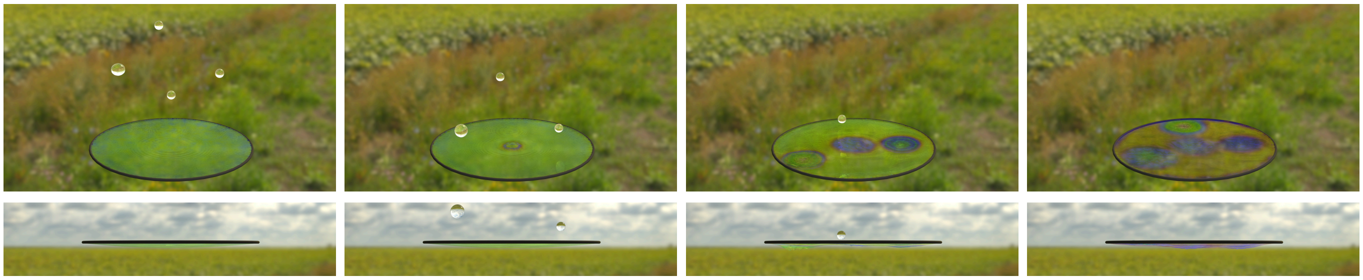

Fig. 15. Droplet Marangoni Effect: Four drops of soap solution is deposited onto the thin film. After the soap solutions is absorbed by the surface, the surface tension of the the impacted area would be lowered, which enables more high-frequency deformation to be observed. Moreover, the droplets induce local non-zero gradients of surface tension, causing the particles to move towards the high surface tension region, which is known as the Marangoni effect. The color change in the surface would reflect such movements.

and tiny droplets. The mechanism for bubble bursting is general to the bubble's shape and dynamics. The shown example directly follows the bubble oscillation example depicted in the upper part of Figure 4. The rupture animation is displayed at 0.1× speed of the oscillation animation. A feature that separates our animation from the existing ones is that our method models the change in the bubble's hue due to surface contraction. As one can see in Figure 5, the bubble's color changes from green to purple, providing an extra layer of richness than if such phenomenon is modeled by a volumetric SPH solver only.

Large-Deforming Thin Films with Rims. 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 ρ_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 $\boldsymbol{F}_{ext} = \beta\left[(\cos\alpha, \sin\alpha, 0) + (0, \cos 2\alpha, \sin 2\alpha)\right]$, in which β 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, countering 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 Γ 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 outwards, 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 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 verisimilitude. 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

Mridul Aanjaneya, Saket Patkar, and Ronald Fedkiw. 2013. A monolithic mass tracking formulation for bubbles in incompressible flow. *J. Comput. Phys.* 247 (2013), 17–61.

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

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

Ryoichi 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, 207–214.

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 Weißmann, 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.

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

Ka Chun Cheung, Leevan Ling, and Steven J Ruuth. 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. IISPH-FLIP for incompressible fluids. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 255–262.

Y Couder, JM Chomaz, and M Rabaud. 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 Gaulon, C Derec, T Combriat, 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 Gissler, Andreas Henne, 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 Gissler, 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.

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

Weizhen Huang, Julian Iseringhausen, Tom Kneiphof, Ziyin Qu, Chenfanfu Jiang, and Matthias B. Hullin. 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 Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. 2013. Implicit incompressible SPH. *IEEE transactions on visualization and computer graphics* 20, 3 (2013), 426–435.

Sadashige Ishida, Peter Synak, Fumiya Narita, Toshiya Hachisuka, and Chris Wojtan. 2020. A Model for Soap Film Dynamics with Evolving Thickness. *ACM Transactions on Graphics* 39, 4, Article 31 (2020), 31:1–31:11 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 Transactions on Graphics (TOG)* 36, 6 (2017), 1–11.

Kei Iwasaki, Keichi Matsuzawa, and Tomoyuki Nishita. 2004. Real-time rendering of soap bubbles taking into account light interference. In *Proceedings Computer Graphics International, 2004.* IEEE, 344–348.

Dariusz Jaszkowski and Janusz Rzeszut. 2003. Interference colours of soap bubbles. *The Visual Computer* 19, 4 (2003), 252–270.

Min Jiang, Yahan Zhou, Rui Wang, Richard Southern, and Jian Jun Zhang. 2015. Blue noise sampling using an SPH-based method. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–11.

Myungjoo Kang, Barry Merriman, and Stanley Osher. 2008. Numerical simulations for the motion of soap bubbles using level set methods. *Computers & fluids* 37, 5 (2008), 524–535.

Richard Keiser, Bart Adams, Philip Dutré, Leonidas J Guibas, and Mark Pauly. 2006. Multiresolution particle-based fluids. *Technical Report/ETH Zurich, Department of Computer Science* 520 (2006).

Namjung Kim, SaeWoon Oh, and Kyoungju Park. 2015. Giant soap bubble creation model. *Computer Animation and Virtual Worlds* 26, 3-4 (2015), 445–455.

Theodore Kim, Jerry Tessendorf, and Nils Thuerey. 2013. Closest point turbulence for liquid surfaces. *ACM Transactions on Graphics (TOG)* 32, 2 (2013), 15.

M. Kness. 2008. ColorPy-A Python package for handling physical descriptions of color and light spectra. (2008).

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*, Wenzel Jakob and Enrico Puppo (Eds.). The Eurographics Association.

Rongjie Lai, Jiang Liang, and Hongkai Zhao. 2013. A local mesh method for solving PDEs on point clouds. *Inverse Problems & Imaging* 7, 3 (2013).

Peter Lancaster and Kes Salkauskas. 1981. Surfaces generated by moving least squares methods. *Mathematics of computation* 37, 155 (1981), 141–158.

David Levin. 1998. The approximation power of moving least-squares. *Mathematics of computation* 67, 224 (1998), 1517–1531.

MB Liu and GR Liu. 2010. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of computational methods in engineering* 17, 1 (2010), 25–76.

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 2D simulation on moving 3D surfaces. 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.

Matthias Müller, David Charypar, and Markus H Gross. 2003. Particle-based fluid simulation for interactive applications.. In *Symposium on Computer animation*. 154–159.

Metin Muradoglu and Gretar Tryggvason. 2008. A front-tracking method for computation of interfacial flows with soluble surfactants. *J. Comput. Phys.* 227, 4 (2008), 2238–2262.

Andrew Nealen. 2004. An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation. *URL: http://www. nealen. com/projects* 130, 150 (2004), 25.

Xingyu Ni, Bo Zhu, Bin Wang, and Baoquan Chen. 2020. A level-set method for magnetic substance simulation. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 29–1.

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

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 Gissler, 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 Ihmsen, 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. 2012. Smoothed particle hydrodynamics and magnetohydrodynamics. *J. Comput. Phys.* 231, 3 (2012), 759–794.

Steven J Ruuth 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 I Saye and James A Sethian. 2013. 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 realistic 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* 83, 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 level-set method for interfacial flows with surfactant. *J. Comput. Phys.* 212, 2 (2006), 590–616.

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

Tao Yang, Jian Chang, Ming C Lin, Ralph R Martin, Jian J Zhang, and Shi-Min Hu. 2017a. A unified particle system framework for multi-phase, multi-material visual simulations. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–13.

Tao Yang, Ralph R Martin, Ming C Lin, Jian Chang, and Shi-Min Hu. 2017b. Pairwise force SPH model for real-time multi-interaction applications. *IEEE transactions on visualization and computer graphics* 23, 10 (2017), 2235–2247.

Jong-Chul Yoon, Hyeong Ryeol Kam, Jeong-Mo Hong, Shin Jin Kang, and Chang-Hun Kim. 2009. Procedural synthesis using vortex particle method for fluid simulation. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 1853–1859.

Wen Zheng, Jun-Hai Yong, and Jean-Claude Paul. 2009. Simulation of bubbles. *Graphical Models* 71, 6 (2009), 229–239.

Bo Zhu, Ed Quigley, Matthew Cong, Justin Solomon, and Ronald Fedkiw. 2014. Codimensional surface tension flow on simplicial complexes. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.

Bo Zhu, Xubo Yang, and Ye Fan. 2010. Creating and preserving vortical details in sph fluid. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 2207–2214.

A MOMENTUM EQUATION AT CENTER SURFACE

The normal and tangential components of (6) are

$$\begin{cases} p - p_a + 2\kappa^{\pm}\gamma & = 2\mu \boldsymbol{n}^{\pm} \cdot (\nabla u) \cdot \boldsymbol{n}^{\pm} + O(h^2), \\ \boldsymbol{t}_x^{\pm} \cdot (\nabla_{s^{\pm}} \gamma) & = \mu \boldsymbol{t}_x^{\pm} \cdot [\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T] \cdot \boldsymbol{n}^{\pm} + O(h^2), \\ \boldsymbol{t}_y^{\pm} \cdot (\nabla_{s^{\pm}} \gamma) & = \mu \boldsymbol{t}_y^{\pm} \cdot [\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T] \cdot \boldsymbol{n}^{\pm} + O(h^2), \end{cases} \tag{22}$$

respectively. Using (3), we obtain

$$\begin{cases} \boldsymbol{t}_x^{\pm} \cdot (\nabla_{s^{\pm}} \gamma) \\ \quad = (\boldsymbol{e}_x \mp \boldsymbol{e}_y \times \nabla_s h) \cdot (\nabla_s \gamma \pm \nabla_s h \boldsymbol{e}_z \cdot \nabla \gamma \pm \boldsymbol{e}_z \nabla_s h \cdot \nabla \gamma) + O(h^2) \\ \quad = \boldsymbol{e}_x \cdot \nabla_s \gamma + O(h^2), \\ \boldsymbol{t}_y^{\pm} \cdot (\nabla_{s^{\pm}} \gamma) \\ \quad = (\boldsymbol{e}_y \pm \boldsymbol{e}_x \times \nabla_s h) \cdot (\nabla_s \gamma \pm \nabla_s h \boldsymbol{e}_z \cdot \nabla \gamma \pm \boldsymbol{e}_z \nabla_s h \cdot \nabla \gamma) + O(h^2) \\ \quad = \boldsymbol{e}_y \cdot \nabla_s \gamma + O(h^2). \end{cases} \tag{23}$$

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

$$\begin{cases} p - p_a + 2(\pm\kappa_c + \kappa_h)\gamma = \\ \quad 2\mu(\pm \boldsymbol{e}_z - \nabla_s h) \cdot (\nabla u) \cdot (\pm \boldsymbol{e}_z - \nabla_s h) + O(h^2), \\ \boldsymbol{e}_x \cdot \nabla_s \gamma = \\ \quad \mu(\boldsymbol{e}_x \mp \boldsymbol{e}_y \times \nabla_s h) \cdot [\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T] \cdot (\pm \boldsymbol{e}_z - \nabla_s h) + O(h^2), \\ \boldsymbol{e}_y \cdot \nabla_s \gamma = \\ \quad \mu(\boldsymbol{e}_y \pm \boldsymbol{e}_x \times \nabla_s h) \cdot [\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T] \cdot (\pm \boldsymbol{e}_z - \nabla_s h) + O(h^2). \end{cases} \tag{24}$$

Further, we substitute the component-wise forms of $\nabla_s h$ and $\nabla \boldsymbol{u}$ into (24) as

$$\begin{cases} p - p_a + 2(\pm\kappa_c + \kappa_h)\gamma = \\ \quad 2\mu \left(\frac{\partial w}{\partial z} \mp \frac{\partial h}{\partial x} \frac{\partial u}{\partial z} \mp \frac{\partial h}{\partial y} \frac{\partial v}{\partial z} \right) + O(h^2), \\ \frac{\partial \gamma}{\partial x} = \\ \quad \mu \left[\pm \frac{\partial w}{\partial x} - 2 \frac{\partial h}{\partial x} \left(\frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right) - \frac{\partial h}{\partial y} \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \pm \frac{\partial u}{\partial z} \right] + O(h^2), \\ \frac{\partial \gamma}{\partial y} = \\ \quad \mu \left[\pm \frac{\partial w}{\partial y} - 2 \frac{\partial h}{\partial y} \left(\frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right) - \frac{\partial h}{\partial x} \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \pm \frac{\partial v}{\partial z} \right] + O(h^2). \end{cases} \tag{25}$$

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 $\boldsymbol{u}_s$ as

$$\begin{cases} u = u(z = 0) + \left. \frac{\partial^2 u}{\partial z^2} \right|_{z=0} z^2 + O(h^4), \\ v = v(z = 0) + \left. \frac{\partial^2 v}{\partial z^2} \right|_{z=0} z^2 + O(h^4), \end{cases} \tag{26}$$

and with the solenoidal condition, we obtain

$$\frac{\partial w}{\partial z} = -\nabla_s \cdot \boldsymbol{u}_s = -\nabla_s \cdot \boldsymbol{u}_s(z = 0) + O(h^2). \tag{27}$$

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

$$\begin{cases} p = p_a - 2[\kappa_h \gamma + \nabla_s \cdot \boldsymbol{u}_s(z = 0)] - \frac{2}{h} \gamma \kappa_c z + O(h^2), \\ \left. \frac{\partial^2 u}{\partial z^2} \right|_{z=0} = \frac{1}{h} \frac{\partial \gamma}{\partial x} + O(h^2), \\ \left. \frac{\partial^2 v}{\partial z^2} \right|_{z=0} = \frac{1}{h} \frac{\partial \gamma}{\partial y} + O(h^2). \end{cases} \tag{28}$$

Then, we substitute (28) into (5) to obtain the momentum equation at $z = 0$:

$$\begin{aligned} & \rho \frac{D\boldsymbol{u}}{Dt} + O(h^2) = \\ & \quad 2\nabla_s(\kappa_h \gamma + \nabla_s \cdot \boldsymbol{u}_s) + \frac{2\gamma}{h} \kappa_c \boldsymbol{n} + \frac{1}{h} \nabla_s \gamma + \mu \nabla_s^2 \boldsymbol{u} + \boldsymbol{f}. \end{aligned} \tag{29}$$

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 κ, which relates to the pressure jump across the air-fluid interface via the Young-Laplace equation $\Delta p = -\gamma\kappa$.

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	5,185~5501†	Server A with 16 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	5154	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 $\Delta t = 0.02s$. We take CFL condition number $C = 0.1$, so a frame may consist of multiple time iterations depending on v_{max}, and the simulation time of a frame is subject to it.

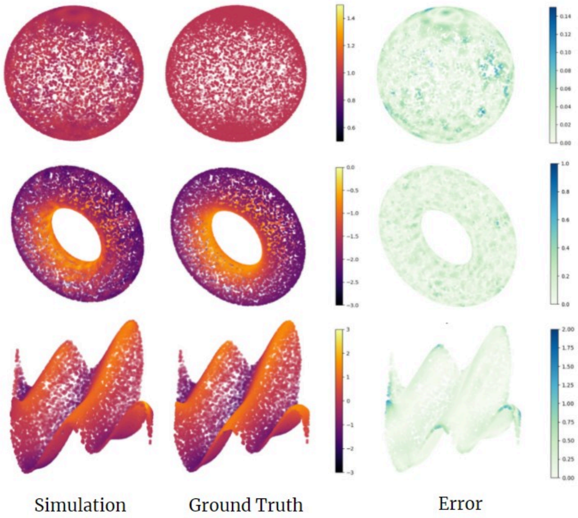

Fig. 16. Estimated Mean Curvature using our codimension-1 differential operators vs. the analytical ground truth. From top to bottom are respectively: the sphere, the torus, and the trigonometric surface.

Our method computes the mean curvature κ_c of the center surface, which are curved 3D surfaces defined by the particles using codimension-1 planar operators. To verify the correctness and robustness of our method under various surface geometries, we compute the mean curvature of three different shapes: a sphere with $R = 1$; a torus

$$(x, y, z) = R\left((c + \cos^{-1}(v))\cos(u), \sin^{-1}(v), (c + \cos^{-1}(v))sin(u)\right)$$

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

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

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.

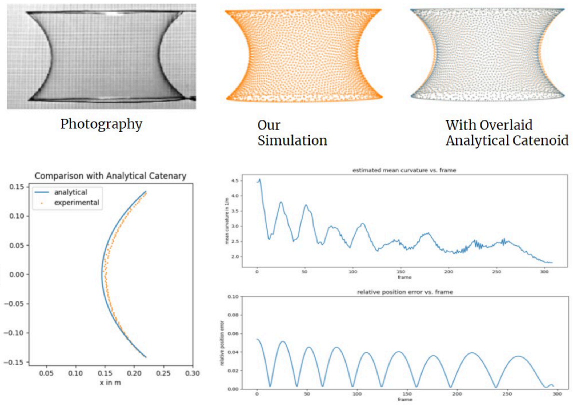

Fig. 17. On the top row, display the thin-film catenoid as photographed, our simulation results, and our results overlaid with a corresponding analytical catenoid. The left figure in the bottom row shows a detailed display of one curve of the cross-section of our simulated thin film with an analytical catenoid. The right two figures plot the average mean curvature and the positional error to the analytical surface.

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. Advected 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%.

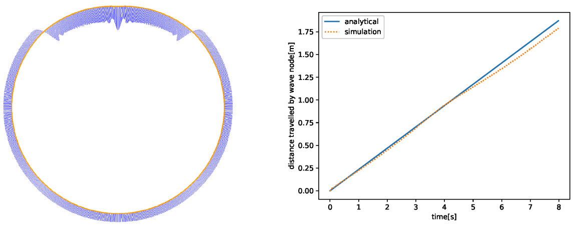

Fig. 18. The capillary wave on a bubble. We take the surface tension coefficient $\gamma = 1.1\times10^{-5}N/m$, thickness $2h = 800nm$, fluid density $\rho = 10^3 kg/m^3$ and bubble radius $R = 0.80m$. The blue lines of the left part are the velocity vectors of particles.

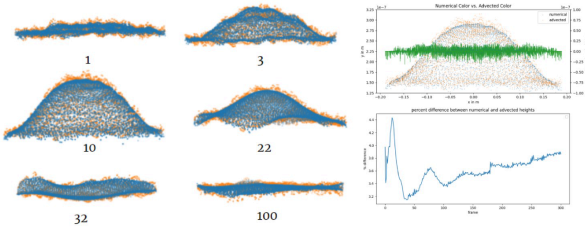

Fig. 19. Left: the surface of numerical height (blue), the surface of advected height (orange). Right: The difference between the two surfaces in a single frame (top) and over time (bottom).

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 $\frac{2\gamma}{h}\kappa_c \boldsymbol{n}$, and thus a capillary wave. Note that κ_c is calculated by half of the Laplacian of surface, then the local dynamics along normal direction system subjects to wave equation

$$\rho \frac{\mathrm{D}^2 r_z^2}{\mathrm{D}t^2} = \frac{\gamma}{h}\frac{\partial^2 r_z}{\partial x^2}, \tag{30}$$

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

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.