A Model for Soap Film Dynamics with Evolving Thickness

SADASHIGE ISHIDA*, PETER SYNAK*, IST Austria
FUMIYA NARITA, Unaffiliated
TOSHIYA HACHISUKA, The University of Tokyo
CHRIS WOJTAN, IST Austria

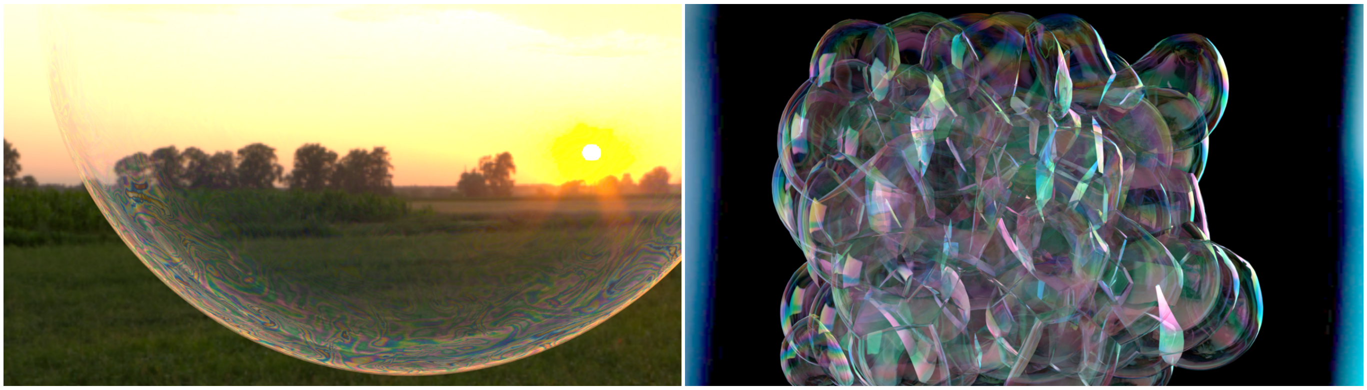

Fig. 1. Our model simulates the evolution of soap films, leading to detailed advection patterns (left) and interplays between draining, evaporation, capillary waves, and ruptures in a foam (right).

Previous research on animations of soap bubbles, films, and foams largely focuses on the motion and geometric shape of the bubble surface. These works neglect the evolution of the bubble's thickness, which is normally responsible for visual phenomena like surface vortices, Newton's interference patterns, capillary waves, and deformation-dependent rupturing of films in a foam. In this paper, we model these natural phenomena by introducing the film thickness as a reduced degree of freedom in the Navier-Stokes equations and deriving their equations of motion. We discretize the equations on a non-manifold triangle mesh surface and couple it to an existing bubble solver. In doing so, we also introduce an incompressible fluid solver for 2.5D films and a novel advection algorithm for convecting fields across non-manifold surface junctions. Our simulations enhance state-of-the-art bubble solvers with additional effects caused by convection, rippling, draining, and evaporation of the thin film.

CCS Concepts: • **Mathematics of computing** → *Geometry*; *Partial differential equations*; • **Computing methodologies** → *Physics*; *Animation.*

Additional Key Words and Phrases: Soap films, Fluid dynamics, Physical modeling

ACM Reference Format:
Sadashige Ishida*, Peter Synak*, Fumiya Narita, Toshiya Hachisuka, and Chris Wojtan. 2020. A Model for Soap Film Dynamics with Evolving Thickness. *ACM Trans. Graph.* 39, 4, Article 31 (July 2020), 11 pages.

1 INTRODUCTION

This paper concerns the animation of soap films, bubbles, and foams. These natural phenomena exhibit fascinating and beautiful complexity in their geometry, dynamics, and color. Interestingly, the influence of fluid forces cascades all the way down to their surface appearance — surface tension and body forces change the film's curvature and thickness, and the film thickness in turn causes swirling interference patterns when it interacts with light. The typical approach to animating these phenomena is to first use a surface-tension solver to simulate dynamic foams while holding the thickness constant, and then retroactively model a noisy film thickness in a surface shader during the rendering step. However, we argue that several interesting phenomena are missed by not directly simulating the evolution of the film thickness. In particular, dynamic film thickness is responsible for the appearance of swirling vortices, ripple patterns, gravity-dependent thickness variation, as well as the bursting of bubbles and thus the ultimate shape of large foam structures.

In this paper, we propose to model soap film thickness within a bubble simulation. We introduce the film thickness as a reduced degree of freedom in the Navier-Stokes equations and derive the relevant equations of motion. These thickness dynamics result in a physical model for film advection, mass conservation, draining, evaporation, and surface tension ripples. We discretize the equations on a non-manifold triangle mesh surface and couple it to an existing bubble solver [Ishida et al. 2017]. We use the resulting simulation framework to reproduce the aforementioned thickness-dependent natural phenomena, as illustrated in Figure 1. Our contributions are:

Authors' addresses: Sadashige Ishida*, Peter Synak*, IST Austria; Fumiya Narita, Unaffiliated; Toshiya Hachisuka, The University of Tokyo; Chris Wojtan, IST Austria, *joint first authors (alphabetical order).

2020. 0730-0301/2020/7-ART31
https://doi.org/

- Equations of motion describing time evolution of both surface deformation and film thickness in a compatible manner.
- An extension of semi-Lagrangian advection algorithms to non-manifold geometry.
- A pressure projection scheme for compressible fields that enables local conservation of mass.
- A computational scheme of the above equations that allows simulation of thin-film turbulence, draining, capillary waves, and evaporation.

2 RELATED WORK

This section first provides a background on bubble physics. Afterward, we review the computer graphics literature on the simulation of bubbles, segueing into techniques for simulating the geometry and motion of thin films.

2.1 Background

Bubbles and foams are nearly incompressible liquids which evolve according to surface tension and pressure forces [Chomaz 2001; Couder et al. 1989]. The surface tension causes bubbles to minimize their film area subject to a volume constraint imposed by the enclosed air, so mathematicians often reduce the problem of finding a soap film's equilibrium geometry to that of Plateau's problem for enclosing a given volume with a surface of minimal area [Eppstein 2012; Hutchings et al. 2002; Weaire and Phelan 1994]. This minimal surface idea led to several popular tools for numerically minimizing surfaces, such as the Surface Evolver of Brakke [1992].

However, we argue that geometry is not the only interesting part of bubble physics. The thickness of a bubble also evolves according to the Navier-Stokes equations, causing intricate swirl patterns in the presence of convective currents [Seychelles et al. 2008]. The thickness also evolves according to the gravity and pressure forces, leading to rainbow-like patterns oriented with the body force [Atkins and Elliott 2010; Gil et al. 2019], and an eventual structured rupturing of bubbles in a foam as the top bubbles thin out faster than the bottom ones [Saye and Sethian 2013].

2.2 Bubble Animation

In computer graphics, soap bubbles were first modeled as static geometry [Glassner 2000a], with rainbow-like colors resulting from thin-film interference [Belcour and Barla 2017; Glassner 2000b; Huang et al. 2020; Iwasaki et al. 2004]. Researchers often model the dynamics of many small bubbles using particles [Busaryev et al. 2012; Cleary et al. 2007; Greenwood and House 2004; Hong et al. 2008; Langlois et al. 2016], typically assuming that each bubble is a sphere with a variable radius. Volumetric simulation can model extremely deformable bubbles [Kim et al. 2007; Saye and Sethian 2013; Zheng et al. 2006], but these methods often require extreme care to avoid accidental topology changes (rupturing films) or mass-conservation errors (shrinking bubbles). Patkar et al. [2013] modeled bubbles that transition between particles and volumes.

To avoid the modeling errors associated with each of these extremes, many researchers choose to model bubbles as a surface enclosing a given volume. Early work on this problem use mass-spring networks to model a collection of bubbles [Durikovic 2001] and individual giant bubbles [Kim et al. 2015]. Advances in triangle mesh topology processing tools [Brochu and Bridson 2009; Campen and Kobbelt 2010; Pavić et al. 2010; Wojtan et al. 2009, 2010; Zhou et al. 2016], especially the *Los Topos* algorithm [Da et al. 2014], paved the way for simulating more complicated foam structures with non-manifold intersections. Da et al. [2015] introduced a new method for evolving the triangle mesh by modeling soap films as vortex sheets. Their method enforces the incompressibility of the air phase and the preservation of circulation by using divergence-free vorticity primitives as simulation degrees of freedom. Ishida et al. [2017], on the other hand, enforced an integral constraint on the air phase to preserve volume, and developed a geometric flow for surface tension computation. Their approach does not guarantee circulation preservation, but it is more computationally efficient than Da et al.'s N-body vortex solver. These mesh-based surface tracking tools have also proven useful for animating liquid surfaces [Brochu et al. 2010; Da et al. 2016; Thürey et al. 2010; Yu et al. 2012; Zhang et al. 2012] and even stratified fluids [Brochu et al. 2012]. Zhu et al. [2014] introduced a hybrid method for combining many of the ideas above (particles, meshes, and volumes) to efficiently simulate bubbles and liquids by tailoring the relevant simulation primitives to their physical dimension.

2.3 Thin Film Simulation

With the exceptions of Saye and Sethian [2013], which requires a highly detailed volumetric simulation, and the concurrent work of Huang et al. [2020], which simulates iridescent patterns on a spherical bubble, all of the film animation methods described above assume trivial dynamics within the soap film itself. The recent foam simulation advances of Da et al. [2015] and Ishida et al. [2017], for example, effectively assume that the soap film is infinitesimally thin and has no influence on the foam evolution.

On the contrary, we are particularly interested in film thickness dynamics and the visual effects it can create. In this direction, several researchers proposed efficient and detailed simulations for advecting a scalar field along a surface, but they either assume static triangle meshes [Azencot et al. 2014; Shi and Yu 2004; Stam 2003] or perfect spheres [Hill and Henderson 2016; Huang et al. 2020]. Researchers have also investigated wave equations [Bojsen-Hansen et al. 2012; Thürey et al. 2010; Yang et al. 2016], shallow water equations [Angst et al. 2008; Wang et al. 2007], thin plate equations [Yu et al. 2012], and water wave dynamics [Kim et al. 2013] on moving surfaces.

More recently, researchers in computer graphics derived discrete governing equations for viscous film dynamics [Azencot et al. 2015b; Vantzos et al. 2018]. Similar to mathematical models for soap film dynamics [Schwartz and Roy 1999], this method employs the "lubrication approximation" [Oron et al. 1997; Reynolds 1886] which reduces the dimensionality of the Navier-Stokes equations based on a small length-scale assumption. This direction comes closest to our goals, as it models surface tension and draining due to gravity by evolving a scalar thickness function. However, previous work on thin-film simulation in computer graphics is limited to highly viscous dynamics on static, manifold surface geometry. In contrast, we seek to simulate lively, convective dynamics of thin films coupled with a dynamically evolving bubble solver with non-manifold surface junctions.

Table 1. Symbols

Symbol	Meaning
S	surface representing film center
h	thickness of film
S^{u}	surface representing upper air-liquid interface
S^{l}	surface representing lower liquid-air interface
$\boldsymbol{x}$	3D position of a liquid particle
$\boldsymbol{x}^{\mathrm{u}}$	3D position of a particle on S^{u}
$\boldsymbol{x}^{\mathrm{l}}$	3D position of a particle on S^{l}
$\boldsymbol{n}$	surface normal of S
$\boldsymbol{n}^{\mathrm{u}}$	surface normal of S^{u}
$\boldsymbol{n}^{\mathrm{l}}$	surface normal of S^{l}
Γ	vertical line segment connecting S^{u} and S^{l}
$\boldsymbol{X}$	average position of $\boldsymbol{x}$ over Γ
p	pressure at a point in the liquid
P	pressure integrated over Γ
H	mean curvature of S
H^{u}	mean curvature of S^{u}
H^{l}	mean curvature of S^{l}
σ	surface tension coefficient
ρ	density coefficient
c_{evap}	evaporation rate per unit time

3 SOAP FILM DYNAMICS WITH VARYING THICKNESS

In this section, we derive equations describing the evolution of soap film thickness. We start by introducing a thin-film assumption into the Navier-Stokes equations, which reduces the film thickness to a single scalar function at each point on the surface. We ultimately arrive at three differential equations describing the film evolution: one for normal motions, one for tangential motions, and one for evolving the thickness of the film.

3.1 Definitions and Setting

Motion. We suppose that the motion of fluid in the films is governed by the inviscid Navier-Stokes equations (Euler equations) with surface tension:

$$\rho\frac{D\dot{\boldsymbol{x}}}{Dt} = -2\sigma(\delta_{S^{\mathrm{u}}}H^{\mathrm{u}}\boldsymbol{n}^{\mathrm{u}} + \delta_{S^{\mathrm{l}}}H^{\mathrm{l}}\boldsymbol{n}^{\mathrm{l}}) - \nabla p + f \tag{1}$$

$$\nabla\cdot\dot{\boldsymbol{x}} = 0 \tag{2}$$

where $\dot{\boldsymbol{x}}$ is velocity, p is pressure, f is a body force, and σ is surface tension strength; $H^{\mathrm{u}}, H^{\mathrm{l}}$ and $\boldsymbol{n}^{\mathrm{u}}, \boldsymbol{n}^{\mathrm{l}}$ are the mean curvatures and normals on the air-liquid interface S^{u} and liquid-air interface S^{l}; and $\delta_{S^{\mathrm{u}}}, \delta_{S^{\mathrm{l}}}$ are the Dirac delta functions that vanish outside $S^{\mathrm{u}}, S^{\mathrm{l}}$. (We use the notation u and l to distinguish between the "upper" and "lower" interfaces, and we list the remainder of out notation in Table 1.) The surface tension acts only on the two interfaces while the pressure and its spatial gradient are defined everywhere else. We assume that p is twice differentiable everywhere except on the interfaces, and takes a constant value in each air region separated by the films, similar to Ishida et al. [2017].

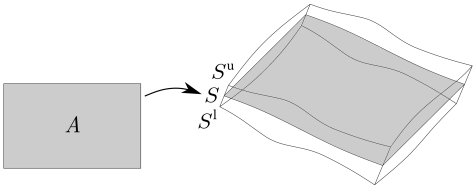

Fig. 2. Our representation of soap films by the surfaces S, S^{u} and S^{l} as embeddings of $A \in \mathbb{R}^2$ into $\mathbb{R}^3$. S^{u} and S^{l} represent the interfaces between air and liquid.

Geometry. We model liquid films as an evolving height field on surfaces embedded in three-dimensional Euclidean space. A surface is a smooth map

$$S : A \longrightarrow \mathbb{R}^3 \tag{3}$$

where A is a connected subset of $\mathbb{R}^2$. For each $S(a,b) \in \mathbb{R}^3$, we define local coordinates given by the local basis

$$\boldsymbol{e}_z := S_a \times S_b / |S_a \times S_b| = \boldsymbol{n} \tag{4}$$

$$\boldsymbol{e}_x := S_a / |S_a| \tag{5}$$

$$\boldsymbol{e}_y := \boldsymbol{e}_z \times \boldsymbol{e}_x \tag{6}$$

where S_a and S_b are partial derivatives of S by a and b, and $\boldsymbol{n}$ is the normal. Neighboring points of S can now be parameterized with the local coordinates x, y, z associated with $\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z$.

We define a *thickness field* by a height function

$$h : A \longrightarrow \mathbb{R}. \tag{7}$$

To represent a thin film, we define two maps

$$S^{\mathrm{u}} := S + \frac{1}{2}h\boldsymbol{n}, \tag{8}$$

$$S^{\mathrm{l}} := S - \frac{1}{2}h\boldsymbol{n}. \tag{9}$$

They represent the upper boundary and the lower boundary of the liquid, which correspond to the interfaces of air-liquid and liquid-air respectively as in Figure 2. We denote the normals of S^{u} and S^{l} by $\boldsymbol{n}^{\mathrm{u}}$ and $\boldsymbol{n}^{\mathrm{l}}$. We assume that h and $\nabla_s h$ are small (no steep slopes on the film surface). Both assumptions are valid for soap films as their surfaces are measured to be smooth [Gao et al. 2012]. Accordingly, we can approximate $\boldsymbol{n}^{\mathrm{u}}$ and $\boldsymbol{n}^{\mathrm{l}}$ by $\boldsymbol{n}$, which implies $\boldsymbol{n}^{\mathrm{u}} \approx \boldsymbol{n}^{\mathrm{l}} \approx (0,0,1)$ in local coordinates.

Similar to previous work [Saye and Sethian 2013; Schwartz and Roy 1999], our equations of motion assume that the film's thickness is far smaller than its other length scales. However, as our primary goal is for animation rather than physics, we play rather liberally with the traditional "thin-film" approximations in order to get an efficient discretization that meets our goals. We occasionally employ small length-scale approximations through the course of our derivation, but we do not take the ideas to their Stokes-flow limit in order to retain lively fluid flows. Our main strategy will be to consider integrated quantities along the film's normal direction, rather than treating every point within the film independently. To integrate over a column of water, we first define the integration domain running from one side of the film to the other:

$$\Gamma_{a,b}(z) := \left\{ S(a,b) + z\boldsymbol{n}(a,b) : z \in \left[-\frac{h(a,b)}{2}, \frac{h(a,b)}{2} \right] \right\}. \tag{10}$$

with end points $\boldsymbol{x}^{\mathrm{u}}(a,b) := \Gamma_{a,b}(h/2)$ and $\boldsymbol{x}^{\mathrm{l}}(a,b) := \Gamma_{a,b}(-h/2)$. We denote the average of position $\boldsymbol{x}$ over the normal-oriented column $\Gamma_{a,b}([-h/2, h/2])$ as

$$\boldsymbol{X} := \frac{1}{h} \int_{\Gamma_{a,b}([-h/2,h/2])} \boldsymbol{x}\, d\gamma. \tag{11}$$

where γ is the Lebesgue measure on $\Gamma_{a,b}([-h/2, h/2])$. We use X to denote S represented under the local coordinates. For the sake of brevity, we henceforth omit indices a, b and the integration domain, unless otherwise stated.

In the following subsections, we separately discuss the overall normal and tangential motion of the film, and then we discuss how the film's thickness evolves over time.

3.2 Normal Acceleration

We exploit the relation

$$\rho \int_{\Gamma} \frac{D\dot{\boldsymbol{x}}}{Dt} d\gamma = \rho h \frac{D\dot{\boldsymbol{X}}}{Dt}. \tag{12}$$

For normal acceleration, we need to take only the z-component of the derivative into account. We use subscript z to denote normal components and subscript xy to denote tangential components. By integrating Equation 1 along Γ, we obtain

$$\begin{aligned} \rho \int_{\Gamma} \frac{D\dot{\boldsymbol{x}}_z}{Dt} d\gamma &= \int_{\Gamma} -2\sigma(\delta_{S^{\mathrm{u}}} H^{\mathrm{u}} \boldsymbol{n}^{\mathrm{u}}_z + \delta_{S^{\mathrm{l}}} H^{\mathrm{l}} \boldsymbol{n}^{\mathrm{l}}_z) - \frac{\partial p}{\partial z} + f_z\, d\gamma \\ &= -\sigma(\nabla^2_{s^u} S^{\mathrm{u}} + \nabla^2_{s^l} S^{\mathrm{l}})_z - \int_{\gamma} \frac{\partial p}{\partial z} d\gamma + h f_z \end{aligned} \tag{13}$$

due to the property of the Dirac delta function and the equivalences $2H^{\mathrm{u}}\boldsymbol{n}^{\mathrm{u}} = \nabla^2_{s^u} S^{\mathrm{u}}$ and $2H^{\mathrm{l}}\boldsymbol{n}^{\mathrm{l}} = \nabla^2_{s^l} S^{\mathrm{l}}$ (where ∇^2_s is the Laplace-Beltrami operator on a surface S). Employing the thin film approximation, we assume that the body force is roughly constant from one side of the film to the other, and we assume the sum $\nabla^2_{s^u} S^{\mathrm{u}} + \nabla^2_{s^l} S^{\mathrm{l}}$ is approximately equal to $2\nabla^2_s S$.

For the pressure term we consider three domains: $\Gamma(\{\frac{h}{2}\}), \Gamma(\{\frac{-h}{2}\})$, and the open interval $(\frac{-h}{2}, \frac{h}{2})$. For the end points $\Gamma(\{\frac{h}{2}\})$ and $\Gamma(\{\frac{-h}{2}\})$, we follow Ishida et al's [2017] approach and have

$$\int_{\Gamma(\{\frac{h}{2}\})} \nabla p\, d\gamma = \left(\lim_{\boldsymbol{x} \downarrow \boldsymbol{x}^{\mathrm{u}}} p(\boldsymbol{x}) - \lim_{\boldsymbol{x} \uparrow \boldsymbol{x}^{\mathrm{u}}} p(\boldsymbol{x}) \right) \boldsymbol{n}^{\mathrm{u}} \tag{14}$$

and

$$\int_{\Gamma(\{\frac{-h}{2}\})} \nabla p\, d\gamma = \left(\lim_{\boldsymbol{x} \downarrow \boldsymbol{x}^{\mathrm{l}}} p(\boldsymbol{x}) - \lim_{\boldsymbol{x} \uparrow \boldsymbol{x}^{\mathrm{l}}} p(\boldsymbol{x}) \right) \boldsymbol{n}^{\mathrm{l}} \tag{15}$$

by considering a small domain around $\Gamma(\{\frac{h}{2}\})$ and $\Gamma(\{\frac{-h}{2}\})$, and taking the infinitesimal limits. For the open interval $\Gamma(\frac{-h}{2}, \frac{h}{2})$, it follows

$$\begin{aligned} \int_{\Gamma((\frac{-h}{2}, \frac{h}{2}))} \frac{\partial p}{\partial z} d\gamma &= \lim_{\epsilon \to 0} \int_{\Gamma((\frac{-h}{2}+\epsilon, \frac{h}{2}-\epsilon))} \frac{\partial p}{\partial z} d\gamma \\ &= \lim_{\boldsymbol{x} \uparrow \boldsymbol{x}^u} p(\boldsymbol{x}) - \lim_{\boldsymbol{x} \downarrow \boldsymbol{x}^l} p(\boldsymbol{x}) \end{aligned} \tag{16}$$

from the definition of improper integral and the fundamental theorem of calculus. Applying $\boldsymbol{n}^{\mathrm{u}} \approx \boldsymbol{n}^{\mathrm{l}} \approx \boldsymbol{n}$ and summing these integrals together, we get the integral over the entire domain:

$$\int_{\Gamma} \frac{\partial p}{\partial z} d\gamma = \lim_{\boldsymbol{x} \downarrow \boldsymbol{x}^{\mathrm{u}}} p(\boldsymbol{x}) - \lim_{\boldsymbol{x} \uparrow \boldsymbol{x}^l} p(\boldsymbol{x}) =: \Delta p. \tag{17}$$

This is the pressure difference through the film surface. We thus obtain

$$\boxed{\frac{D\dot{X}_z}{Dt} = -\frac{1}{\rho h}\left(2\sigma\nabla^2_s S + \Delta p\right) + \frac{f_z}{\rho}} \tag{18}$$

This is the equation responsible for deforming the surface geometry by the means of normal acceleration — it moves the surface based on surface tension (curvature), air pressure on both sides of the film, and external forces. This equation is identical to Ishida et al. [2017], except that it accounts for spatially varying thickness h.

3.3 Tangential Acceleration

In the tangential direction, we have

$$\begin{aligned} \rho \int_{\Gamma} \frac{D\dot{\boldsymbol{x}}_{xy}}{Dt} d\gamma &= \int_{\Gamma} -2\sigma(\delta_{S^{\mathrm{u}}} H^{\mathrm{u}} \boldsymbol{n}^{\mathrm{u}}_{xy} + \delta_{S^{\mathrm{l}}} H^{\mathrm{l}} \boldsymbol{n}^{\mathrm{l}}_{xy}) - (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})p + f_{xy}\, d\gamma \\ &= -2\sigma(H^{\mathrm{u}} \boldsymbol{n}^{\mathrm{u}}_{xy} + H^{\mathrm{l}} \boldsymbol{n}^{\mathrm{l}}_{xy}) - (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}) \int_{\Gamma} p\, d\gamma \; + h f_{xy} \end{aligned} \tag{19}$$

based on the twice differentiability of p by x, y. Applying $\boldsymbol{n}^{\mathrm{u}} \approx \boldsymbol{n}^{\mathrm{l}} \approx \boldsymbol{n}$ and the fact that the tangent component of the normal is zero, we obtain

$$\boxed{\frac{D\dot{X}_{xy}}{Dt} = -\frac{\nabla_s P}{\rho h} + \frac{f_{xy}}{\rho}} \tag{20}$$

where ∇_s is the surface gradient and $P := \int_{\Gamma} p d\gamma$ is the integrated pressure over the water column Γ. This equation is responsible for flows along the surface geometry — the material derivative on the left hand side advects tangential velocities based on pressure gradients (enforcing mass conservation) and external forces (like gravity or wind).

3.4 Thickness Evolution

We note that the thickness at a point $S(a, b)$ is

$$h(a,b) = \boldsymbol{x}^{\mathrm{u}}_z(a,b) - \boldsymbol{x}^{\mathrm{l}}_z(a,b). \tag{21}$$

The normal forces acting on the points $\boldsymbol{x}^{\mathrm{u}}(a, b)$ and $\boldsymbol{x}^{\mathrm{l}}(a, b)$ are

$$\begin{aligned} \frac{D\dot{\boldsymbol{x}}^{\mathrm{u}}_z}{Dt} &= \int_{\Gamma(\{\frac{h}{2}\})} \frac{D\dot{\boldsymbol{x}}_z}{Dt} d\gamma \\ &= -\sigma\nabla^2_{s^u} S^{\mathrm{u}}_z + \lim_{\boldsymbol{x} \uparrow \boldsymbol{x}^u} p(\boldsymbol{x}) - \lim_{\boldsymbol{x} \downarrow \boldsymbol{x}^u} p(\boldsymbol{x}) + f_z \end{aligned} \tag{22}$$

and

$$\begin{aligned} \frac{D\dot{\boldsymbol{x}}^{\mathrm{l}}_z}{Dt} &= \int_{\Gamma(\{\frac{-h}{2}\})} \frac{D\dot{\boldsymbol{x}}_z}{Dt} d\gamma \\ &= -\sigma\nabla^2_{s^l} S^{\mathrm{l}}_z + \lim_{\boldsymbol{x} \uparrow \boldsymbol{x}^l} p(\boldsymbol{x}) - \lim_{\boldsymbol{x} \downarrow \boldsymbol{x}^l} p(\boldsymbol{x}) + f_z \end{aligned} \tag{23}$$

from Equation 14 and 15. Plugging these into Equation 21, and approximating $\nabla^2_{s^l} S^{\mathrm{l}} - \nabla^2_{s^u} S^{\mathrm{u}}$ with $\nabla^2_s h$, we obtain

$$\boxed{\frac{D\dot{h}}{Dt} = \frac{1}{\rho}\left(\sigma\nabla^2_s h - Q\right)} \tag{24}$$

where Q is the difference between the pressures outside of the film compared to the pressure inside:

$$Q = \underbrace{\lim_{\boldsymbol{x}\downarrow\boldsymbol{x}^u} p(\boldsymbol{x}) + \lim_{\boldsymbol{x}\uparrow\boldsymbol{x}^l} p(\boldsymbol{x})}_{\text{external pressure}} - \underbrace{\left(\lim_{\boldsymbol{x}\uparrow\boldsymbol{x}^u} p(\boldsymbol{x}) + \lim_{\boldsymbol{x}\downarrow\boldsymbol{x}^l} p(\boldsymbol{x})\right)}_{\text{internal pressure}}. \tag{25}$$

Equation 24 determines the evolution of film thickness h — ignoring the Q term for a moment, this is an advected wave equation for h; it describes how the thickness flows along the film, and it is responsible for capillary (surface tension) waves. The Q term squeezes the film when the external pressure exceeds the outside, and it is generally responsible for volume conservation of the thickness field h, just as pressure acts to preserve fluid volume in a typical incompressible flow solver. When it reaches equilibrium, this equation becomes the sum of the Young-Laplace equation for the upper and the lower boundary, which describes the balance between the surface tension force and the pressure jump through the interface. We note that in our implementation, the Laplace-Beltrami operator $\nabla^2_s h$ is defined both on manifold and non-manifold geometry, following the technique proposed by Ishida et al. [2017].

3.5 Pressure

The past three sections described a reduced model for Equation 1 in terms of normal accelerations, tangential accelerations, and thickness propagation. However, we have not yet described how to compute the pressures.

We first describe how to compute the pressure jump term Δp in Equation 18. Δp is effectively a correction term resulting from the conservation of volume for each air bubble. We incorporate this term following Ishida et al. [2017], by off-setting the surface with the amount Δd in the normal direction such that the air bubbles preserve volume.

Next, we describe how to compute the integrated pressure P for Equation 20. Conservation law form of h gives us

$$\frac{\partial h}{\partial t} = -\nabla_s \cdot (h\dot{X}_{xy}). \tag{26}$$

We then execute steps exactly analogous to those typically used to enforce incompressibility in a fluid solver with a first-order operator splitting approach [Stam 1999]. After computing external forces and advection in Equation 20, and numerically integrating it with a time step Δt, we are left with

$$\dot{X}_{xy}(t+\Delta t) = \tilde{\dot{X}}_{xy}(t) - \frac{\Delta t}{\rho h}\nabla_s P(t), \tag{27}$$

where $\tilde{\dot{X}}_{xy}$ is the "dirty" velocity before enforcing the conservation law. We then multiply by h, take the divergence of both sides, and plug it into Equation 26 to get:

$$\boxed{\frac{\Delta t}{\rho}\nabla^2_s P = \nabla_s \cdot (h\tilde{\dot{X}}_{xy}) + \frac{\partial h}{\partial t}} \tag{28}$$

which is our new Poisson equation for enforcing our film incompressibility constraint. This equation is almost exactly the same as a typical 2D incompressibility solver for triangle meshes, except that it allows mass to transfer between tangential motions $\tilde{\dot{X}}_{xy}$ and normal motions $\frac{\partial h}{\partial t}$. Thus, the tangent velocity field after enforcing the constraint $\dot{X}_{xy}$ will be *compressible*; it will exhibit sinks and sources wherever the film thickness h expands and contracts.

We note that our global projection technique is starkly different from related work on thin films [Azencot et al. 2015b; Saye and Sethian 2013, 2016], which embed mass conservation into a PDE for viscous film evolution. Instead, our solver preserves lively vortices in the tangential velocity field, which are useful for visual effects in computer animation.

Finally, the pressure term Q in Equation 24 should in principle be used for two-way coupling between the film and the surrounding air. It should cause the film to expand or thicken the film based on the balance between internal liquid pressure and the surrounding air pressure, while simultaneously enforcing volume conservation of the liquid. However, because our underlying foam solver [Ishida et al. 2017] treats air pressure as constant per bubble, the external pressure in Q (Equation 25) is simply a constant term per connected film component. This simplified air model leads to only subtle mass-transfer effects due to differences in pressure at non-manifold junctions. We ultimately decided that these coupling effects were too subtle for our purposes.

However, Q can still be useful for enforcing volume conservation within the film. Just like any advection-projection solver, our incompressibility solver described above may be susceptible to numerical drift and cannot exactly preserve film mass for finitely large time steps. We use Q to compensate for measurable drifts in total volume — following the geometric volume conservation approach of Müller [2009], we implement Q as a global mass correction at the end of each time step by scaling up film thickness such that the total mass per connected component is exactly preserved when desired.

3.6 Evaporation and Bursting Bubbles

Now that we have explicit control over the film thickness, we can also incorporate approximate evaporation effects. Current evaporation models feature a complex interdependence of pressure and heat transfer [Ahmed and Pandey 2019]. For our purposes, we found it sufficient to heuristically model evaporation by simply reducing the film thickness at a constant rate: $h \mathrel{-}= c_{\mathrm{evap}}\Delta t$.

We model film bursting by simply deleting a manifold surface region, similar to the simulations of Da et al. [2015]. However, instead of randomly deleting films, we delete a film when its thickness h falls below a critical threshold at some point on the surface. To approximate the effect of surface tension retracting a burst film, we compute the mass of the burst film and distribute a fraction of it to the neighboring bubbles by increasing h in the elements along the film boundary.

4 IMPLEMENTATION

Each time step of our implementation must numerically integrate Equations 18, 20, 28, and 24. We integrate the normal acceleration (Equation 18) in a Lagrangian reference frame; we integrate tangential acceleration (Equation 20) in an Eulerian reference frame by adding body forces, computing advection, and then enforcing the conservation law (Equation 28); and we evolve thickness (Equation 24) using an Eulerian discretization of advection and wave dynamics. These steps are outlined in Algorithms 1, 2, and 3. We will use notations L and E to distinguish motions evaluated in Lagrangian frame and in Eulerian frame. The total velocity is therefore expressed as $\dot{\boldsymbol{X}} = \dot{\boldsymbol{X}}_L + \dot{\boldsymbol{X}}_E$.

4.1 Discretization

We use a triangle mesh to represent surfaces. Each triangle stores a tangential velocity, and each vertex stores a thickness value similar to a staggered grid [Shi and Yu 2004]. For differential operators, we use the discretized divergence of de Goes et al. [2016] and the discretized Laplacian of Ishida et al. [2017].

Our implementation uses *LosTopos* [Da et al. 2014] to track evolution and topology changes of a non-manifold triangle mesh. At the end of each time step, we supply LosTopos with proposed trajectories for the vertices of the mesh. LosTopos then performs remeshing while handling topology changes via operations such as edge splitting, edge collapse, T1 rearrangement processes, and vertex snapping, and returns a watertight mesh satisfying the proposed trajectories as nearly as possible. To adjust this new mesh to our simulation, we linearly interpolate quantities on newly created vertices and triangles from their neighbors, while forcing local liquid volume (area times thickness) to remain unchanged. We use first-order symplectic Euler time integration.

We note that we can attach films to wire boundaries by constraining their degrees of freedom. We constrain vertex positions to remain fixed on the boundary, and we set components of velocity and $\nabla_s h$ which are perpendicular to the boundary equal to zero.

4.2 Advection

We advect velocities originating from normal acceleration in a Lagrangian manner by integrating Equation 18 and using the result to update the position and velocity of mesh vertices. We advect velocities originating from tangential acceleration and thicknesses (Equations 20 and 24) using semi-Lagrangian advection based on the method of Shi and Yu [2004] with an optional second-order upgrade using BFECC [Kim et al. 2005]. We opted for this Lagrangian-Eulerian splitting of advection to reduce the frequency of re-meshing.

We also extend Shi and Yu [2004]'s method for multiple films with non-manifold junctions. To do so, we first perform standard backtracing until the trajectory reaches a non-manifold edge e. Adjacent to e are the last backtraced triangle T_0 and two or more backtrace candidate triangles $T_1, T_2, \dots$. For each candidate triangle T_i we measure the absolute flux through e, given by $f_i^e := \dot{\boldsymbol{X}}_{xy}^{T_i} \cdot \boldsymbol{n}_e^{T_i}$, where $\dot{\boldsymbol{X}}_{xy}^{T_i}$ is the tangential velocity at T_i and $\boldsymbol{n}_e^{T_i}$ is the edge normal of e in the plane of T_i, oriented away from T_i. Assuming the absolute flux f_i^e is positive (i.e. liquid is flowing from T_i towards T_0), we measure the relative flux fr_i^e as the proportion of f_i^e to the sum of all positive fluxes over e,

$$fr_i^e := \frac{f_i^e}{\sum_{f_j^e>0} f_j^e}. \tag{29}$$

For each triangle T_i with a positive absolute flux, we independently continue the backtrace through T_i, therefore generally obtaining multiple backtrace paths, see Figure 3. When the backtrace is complete, for every path we determine its weight as a product of all the relative fluxes over non-manifold edges along this path. Summing up weighted contributions from all the backtrace paths gives us the final advection velocity. BFECC paths are traced analogously. Figure 4 shows how film thickness drains downward on a bubble cluster, advecting through non-manifold junctions. We use direct sparse Cholesky decomposition for solving our pressure projection (Equation 28).

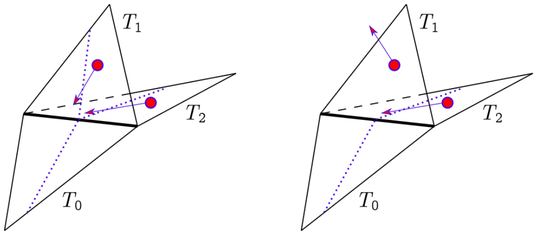

Fig. 3. When the semi-Lagrangian backtrace (dotted line) reaches a non-manifold edge from triangle T_0, we evaluate the flow on the remaining adjacent triangles T_1 and T_2. If the velocities on both T_1 and T_2 indicate that liquid flows into T_0, we split the backtrace (left). If the liquid on $T1$ flows away from T, while liquid on T_2 flows toward T, we don't split the path and only continue the backtrace through T_2 (right).

4.3 Deformation Response and Mass Conservation

Since the film deforms over time, we need to properly handle the changes introduced by deforming local frames. As the mesh deforms, we update Eulerian velocity according to the transformation affecting the deformed triangle. Before deformation occurs, the Eulerian velocity in its local basis is expressed as $\dot{\boldsymbol{X}}_E \;= c_1 e_1 + c_2 e_2$ with triangle edge vectors e_1, e_2 serving as a local basis (generally not orthogonal or normalized) and real coefficients c_1, c_2. After deformation, we update the velocity vector using the same coefficients c_1, c_2 but with the transformed triangle edges e_1', e_2' forming the transformed local basis. This velocity update compensates for a dynamically changing metric on the mesh surface due to affine transformations including non-rigid motions such as scaling and shearing.

Deformation also affects mass conservation. Considering both Eulerian and Lagrangian frames, Equation 28 becomes

$$\frac{\Delta t}{\rho}\nabla_s^2 P = \nabla_s \cdot (h\tilde{\dot{\boldsymbol{X}}}_{xy,E}) + \left(\frac{\partial h}{\partial t}\right)_E + \nabla_s \cdot (h\tilde{\dot{\boldsymbol{X}}}_{xy,L}) + \left(\frac{\partial h}{\partial t}\right)_L. \tag{30}$$

We apply a first-order time splitting scheme to solve this equation one term at a time. We address the two Lagrangian terms by ensuring that liquid volume is conserved while the mesh expands, by locally rescaling the thickness such that the local volume remains constant:

$h_{\text{new}}A_{\text{new}} = h_{\text{old}}A_{\text{old}}$, where A is the Voronoi area of the vertex. This conservation law during surface deformation effectively sets the Lagrangian terms to equal zero, leaving us with only the Eulerian terms for computing P.

ALGORITHM 1: Motion caused by normal acceleration

```
1 Add the surface tension force and external force
  $\dot{\boldsymbol{X}}'_L\ (t) = \dot{\boldsymbol{X}}_L\ (t) + 2\sigma\nabla_s^2 S\frac{\Delta t}{\rho h} + f_z\frac{\Delta t}{\rho}$
  $\boldsymbol{X}'(t) = \boldsymbol{X}(t) + \Delta t\dot{\boldsymbol{X}}'_L\ (t)$
2 Perform air volume preservation
  $\boldsymbol{X}(t + \Delta t) = \boldsymbol{X}'(t) + \Delta d\boldsymbol{n}$
  $\dot{\boldsymbol{X}}_L\ (t + \Delta t) = \dot{\boldsymbol{X}}'_L\ (t) + (\Delta d/\Delta t)\boldsymbol{n}$
```

ALGORITHM 2: Motion caused by tangential acceleration

```
1 Add external force and perform advection along $\dot{\boldsymbol{X}}_E$
  $\tilde{\dot{\boldsymbol{X}}}_E\ (t) = \dot{\boldsymbol{X}}_E\ (t) + \Delta t(\dot{\boldsymbol{X}}_E\ (t) \cdot \nabla_s)\dot{\boldsymbol{X}}_E\ (t) + \Delta t\frac{f_{xy}}{\rho}$
2 Compute the pressure term
  $\dot{\boldsymbol{X}}_E\ (t + \Delta t) = \tilde{\dot{\boldsymbol{X}}}_E\ (t) - \nabla_s P\frac{\Delta t}{\rho h}$
  by solving for $P$: $\frac{\Delta t}{\rho}\ \nabla_s^2 P = \nabla_s \cdot (h\tilde{\dot{\boldsymbol{X}}}_E\ (t)) + (\partial h/\partial t)_E$
```

ALGORITHM 3: Thickness evolution

```
1 Perform advection along $\dot{\boldsymbol{X}}_E$
  $\tilde{\dot{h}}_E\ (t) = \dot{h}_E\ (t) + \Delta t(\dot{\boldsymbol{X}}_E\ (t) \cdot \nabla_s)\dot{h}_E\ (t)$
  $\tilde{h}(t) = h(t) + \Delta t(\dot{\boldsymbol{X}}_E\ (t) \cdot \nabla_s)h(t)$
2 Add the surface tension force
  $\dot{h}_E\ (t{+}\Delta t) = \tilde{\dot{h}}_E\ (t) + \sigma\nabla_s^2\tilde{h}(t)\frac{\Delta t}{\rho}$
  $h(t + \Delta t)\ = \tilde{h}(t) + \Delta t\dot{h}_E\ (t{+}\Delta t)$
3 Perform liquid volume preservation (Section 3.5)
4 Account for evaporation
```

ALGORITHM 4: Main algorithm

```
1 Move constrained vertices based on user input
2 Integrate tangential acceleration (Algorithm 2)
3 Evolve thickness (Algorithm 3)
4 Integrate normal acceleration (Algorithm 1)
5 Update thickness and Eulerian velocity according to deformation
6 Remesh by LosTopos
```

5 RESULTS

We used our simulator to produce a number of soap-film simulations. We begin by demonstrating the method's behavior on static surfaces and validating it against physical experiments. Next, we showcase the benefits of our method with detailed animations of soap-film dynamics.

5.1 Demonstrations and Validation

Figures 4 and 5 illustrate how our thin film evolution differs from typical 2D incompressible Navier-Stokes solvers by simulating the behavior of a film in the presence of gravity. The film's thickness, which is originally constant over the shape, drains downward over time. In order to concentrate film thickness at the bottom of this shape, the velocity field needs to exhibit sources and sinks. In contrast, 2D incompressible fluid solvers like Shi and Yu [2004] only allow divergence-free velocity fields on their surfaces, which prohibit these mass concentrations.

As a vertical planar soap film drains downward, its mass thins at the top and thickens at the bottom. Light interfering with the the film's thickness variation creates horizontal bands of rainbows. The film reaches a steady state when the gravitational forces, the surface tension forces, and the pressure forces balance; the thickness distribution at equilibrium can be analytically described as an exponential function in height [Couder et al. 1989]. In Figure 6, we show how our simulation compares to both a real-world experiment [Atkins and Elliott 2010] and the analytical solution. To minimize mesh-dependent bias, we implemented this example on a regular grid instead of a triangle mesh.

Seychelles et al. [2008] photographed the effects of temperature change and surfactant concentration gradients on a curved soap film surface, revealing beautiful convected rainbow patterns. Figure 7 shows a comparison between their experiment and an animation produced using our algorithm with an empirically chosen thickness and initial velocity field. The left image in Figure 1 shows subtle swirls resulting from a high-resolution simulation initialized by a high-frequency curl-noise pattern.

5.2 Animations

Figure 8 shows a simulation of a large deformed sphere, initialized with a thickness field proportional to height. As the simulation evolves, we observe both surface deformation tending towards a regular spherical shape and the varying of color patterns, resulting from the interplay between surface deformation and thickness evolution. We compare this result to the identical simulation rendered with constant thickness in the same figure.

We can also use our soap-film simulation algorithm to animate clusters of several bubbles. Figure 9 shows a cluster of ten bubbles bursting and rearranging. We can compare this animation to a similar one by Da et al. [2015]. We note that our algorithm enhances the simple bubble rearrangement with slowly convecting interference patterns as the film drains, and sudden rainbow ripples as the film rearranges. Importantly, the bursting in this example is entirely driven by our thin film solver, not by a random film-deletion strategy.

As we increase the number of bubbles in a cluster, we begin to approximate complex dynamics found in soap foams. We believe these foam structures especially highlight the utility of our algorithm, because they show how all of the subtle film forces work together to create novel dynamics and visual effects. For example, the right image in Figure 1 exhibits the following sequence of events in a foam consisting of one hundred bubbles: first, gravity and advection cause the film to drain from the top of the foam downward; the thinner film near the top of the cluster evaporates away and causes

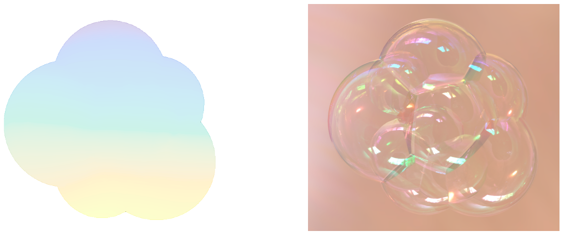

Fig. 4. Film thickness drains due to gravity on a bubble cluster, creating a smooth gradient in thickness even across non-manifold junctions. Left: Film thickness varies from 900nm (yellow) to 500nm (blue) Right: Photorealistic rendering of these bubbles.

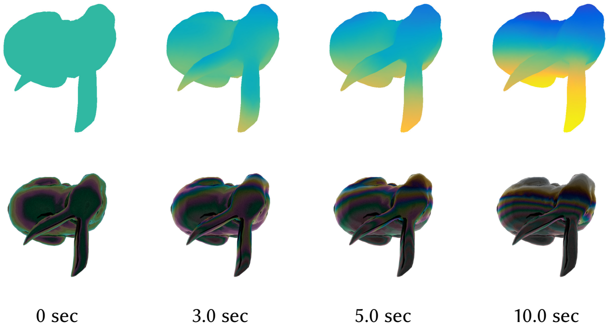

Fig. 5. A bunny-shaped film draining due to gravity. Top: Film thickness varies from 1200nm (yellow) to 200nm (blue). Bottom: The corresponding soap film rendered under white light.

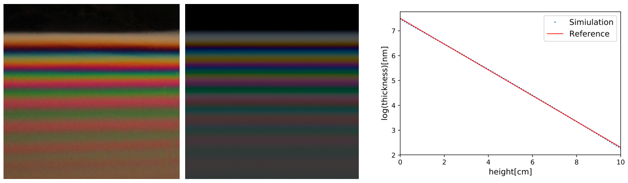

Fig. 6. The "Newton pattern" photographed from a real soap film (left) and a simulated soap film (middle). Our simulation setting includes a vertically standing square film with side length 10cm and uniform thickness 500nm, draining with gravity coefficient 9.81m/s^2 and surface tension coefficient 10^{-5} . Plotting the simulated thickness distribution on a logarithmic scale (right) shows that our simulated result is nearly an exponential function at the equilibrium state, as predicted.

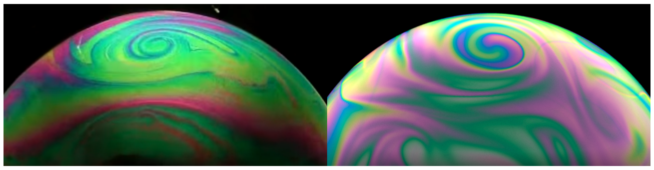

Fig. 7. Comparison between a real-world swirling soap film experiment by Seychelles et al. [2008] (left) and our simulation (right). The simulation was initialized on a sphere with thickness field linearly graded from 1200nm on the front side to 200nm on the back side. To achieve the swirling motion, we set the velocity field in clockwise horizontal direction in the upper hemisphere, counter-clockwise horizontal direction in the lower hemisphere, and evolve it for several seconds.

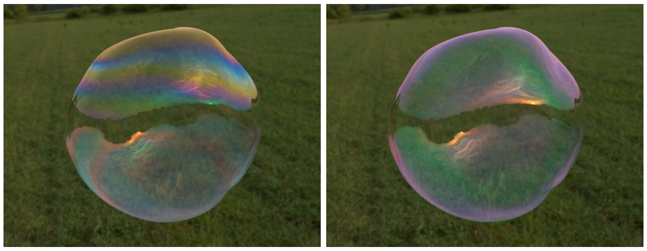

Fig. 8. A single large deforming bubble simulated using our model rendered with variable thickness (left) and with a simplified constant thickness (right). Note how our result exhibits interference patterns depending on the film's height and deformation.

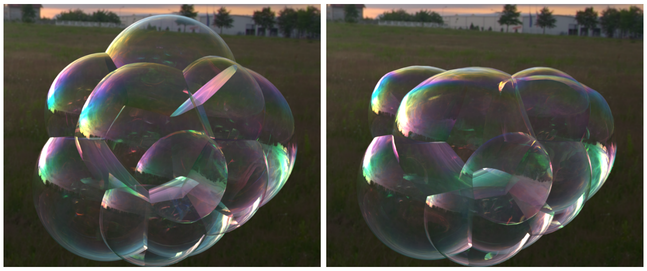

Fig. 9. A cluster of ten bubbles bursting and rearranging, showing color variations due to convection and capillarity.

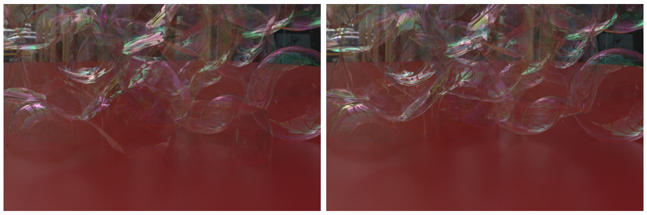

Fig. 10. Foam colliding with a boundary causes bubbles to deform (left) and burst (right).

a bubble to burst; the sudden imbalance from the burst sends capillary waves rippling around the neighboring bubbles; these ripples sometimes constructively interfere, leading to more thin films and more bursting; the film then drains again and the process repeats.

Figure 10 shows how foams can be destroyed via deformation. Here, the bubbles are pushed downward against a solid boundary, causing them to change shape. The volume conservation of the trapped air causes the surface area to increase, forcing the film to thin out due to mass conservation. The thin film leads to bursting, as described above.

Table 2. Performance timings in seconds per frame. We performed the computations on an Intel Core i7-6850K 3.6GHz CPU.

Scene	Figure #	Vertices	Triangles	BFECC	Surface Flow + Deformation	Surface Flow Only	Overhead of Our Method
100 bubble cluster	Fig 1 right	25k	53k	no	9.14s	0.59s	7%
10 bubble cluster	Fig 9	8k	15k	no	1.41s	0.06s	4%
Large deforming sphere	Fig 8	29k	58k	yes	5.93s	1.41s	31%
Sphere patterns	Fig 1 left	102k	204k	yes	N/A	7.46s	N/A

We implemented our thin film simulator on top of Ishida et al. [2017]'s source code for simulating bubble dynamics. The overhead added by simulating film thickness depends on the choice of advection scheme. For basic semi-Lagrangian advection, our implementation added up to 7% to the computation time. When using the more expensive BFECC advection scheme, our simulation added up to 31% to the total computation time. See table 2 for more details on the performance of our prototype implementation. We note that we have not optimized our implementation for performance; we believe a parallel implementation will be a simple way to gain a significant speed-up.

6 DISCUSSION

Modeled phenomena. We believe there are a number of avenues for improvement in our method. First of all, we would like to make our derivation more rigorous, so we can determine exactly when it is a valid simulation regime. Our current model assumes inviscid flow as in the works of Da et al. [2015, 2016] and Ishida et al. [2017] which model inviscid behavior and allow users to add heuristic damping as a tunable parameter. However, since viscosity can affect small scale behaviors, a more consistent treatment may require that we incorporate viscous forces into our model. The scenarios in our results feature relatively slow motions, so we neglected fictitious forces resulting from accelerations. We should incorporate air drag, centrifugal, and Coriolis forces if we wish to model spinning bubbles more accurately. Additionally, while we calculate vertex accelerations in the normal direction, vertices can still have non-zero components of velocity in tangential directions. In order to improve our simulation's physical realism, we should rigorously model how vertex tangential velocities and the tangential flow interact with each other.

We are also interested in more sophisticated models for air-liquid coupling. We know that air currents can lead to beautiful convective film vortices [Seychelles et al. 2008], and our heuristic treatment of the coupling term Q together with the constant air pressure model of Ishida et al. [2017] currently forbids coupling with nontrivial air dynamics. We would also like to explore a more accurate surfactant model, as surface tension gradients can lead to curious visual phenomena like the Marangoni effect.

Next, we are also interested in incorporating effects from related work into our model. Notably, we do not actually animate the process of a film retracting and tearing apart into droplets when it bursts, so we would like to incorporate the thin film rupture model of Zhu et al. [2014].

We also note that our model can be used as a post-process that follows a simulation of surface deformation. A post-process application of our algorithm efficiently animates draining-dependent colors and swirls, but it fails to exhibit thickness-dependent bubble bursting, and it exhibits subtle differences in normal motions (Equation 18), where the pressure gradient accelerates thinner films faster than thicker ones.

Numerical stability, efficiency, and scalability. Our discretization is not unconditionally stable. As our surface deformation scheme is essentially equivalent to the previous work [Ishida et al. 2017], it shares the same numerical stability constraints: the maximal time step is approximately proportional to the inverse of the maximal value of the generalized mean curvature, and depends on other factors such as the stability of the Los Topos surface tracker. Since our model is independent of the discretization or the surface tracker, we expect that more robust ones can potentially improve the stability issue.

We can also make our method more efficient by incorporating multiple time scales [Saye and Sethian 2016]. Perhaps even a simple adaptive time-stepping scheme can help us more efficiently model fast bursting and slow film draining in the same simulation. Another direction for further efficiency is investigating applicability of recently developed surface discretizations [Azencot et al. 2013, 2015a] and numerical solvers for PDEs on evolving surfaces [Dziuk and Elliott 2013; Olshanskii and Xu 2016; Petras and Ruuth 2016] to our governing equations.

Finally, we are eager to compute larger scale simulations by parallelizing our code and optimizing our solver. We believe that our solver can be made efficient enough to simulate orders of magnitude more bubbles; such a simulator may be able to handle enough complexity to begin answering statistical and homogenization questions, like determining the effective material properties of a foam material. We would also be interested in using our solver as a starting point for simulating delicious visco-elasto-plastic foam-like materials such as dough, cream, and baked goods.

Conclusion. We have presented a novel algorithm for animating soap films with evolving film thickness. We derived our model by decoupling normal and tangential accelerations, and extracting film thickness as a key degree of freedom. We then derive three equations of motion, leading to both large-scale geometric deformations and small-scale variations in film thickness. We incorporated a number of effects and forces into our thin film model, namely advection, external forces, capillary waves, evaporation, and bursting. To do so, we also introduced a novel 2.5D pressure solver for enforcing approximate mass conservation of the thin film. Our simulations are capable of reproducing real-world phenomena like Newton's patterns, and they are capable of efficiently animating swirling thickness patterns, capillary ripples, and a physically plausible decaying of foams.

ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers and the members of the Visual Computing Group at IST Austria for their valuable feedback, especially Camille Schreck for her help in rendering. This research was supported by the Scientific Service Units (SSU) of IST Austria through resources provided by Scientific Computing. We would like to thank the authors of [Belcour and Barla 2017] for providing their implementation, the authors of [Atkins and Elliott 2010] and [Seychelles et al. 2008] for allowing us to use their results, and Rok Grah for helpful discussions. Finally, we thank Ryoichi Ando for many discussions from the beginning of the project that resulted in important contents of the paper including our formulation, numerical scheme, and initial implementation. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 638176.

REFERENCES

Shahnawaz Ahmed and Manmohan Pandey. 2019. New insights on modeling of evaporation phenomena in thin films. *Physics of Fluids* 31, 9 (2019), 092001.

Roland Angst, Nils Thuerey, Mario Botsch, and Markus Gross. 2008. Robust and efficient wave simulations on deforming meshes. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1895–1900.

Leslie Atkins and Richard Elliott. 2010. Investigating thin film interference with a digital camera. *American Journal of Physics - AMER J PHYS* 78 (12 2010). DOI: http://dx.doi.org/10.1119/1.3490011

Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Maks Ovsjanikov. 2013. An Operator Approach to Tangent Vector Field Processing. *Computer Graphics Forum* (2013). DOI: http://dx.doi.org/10.1111/cgf.12174

Omri Azencot, Maks Ovsjanikov, Frédéric Chazal, and M Ben-Chen. 2015a. Discrete Derivatives of Vector Fields on Surfaces – An Operator Approach. *ACM Transactions on Graphics* 34, 3 (2015). DOI: http://dx.doi.org/10.1145/2723158

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 (SCA '15)*. Association for Computing Machinery, New York, NY, USA, 137–146. DOI: http://dx.doi.org/10.1145/2786784.2786793

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.

Laurent Belcour and Pascal Barla. 2017. A Practical Extension to Microfacet Theory for the Modeling of Varying Iridescence. *ACM Transactions on Graphics* 36, 4 (July 2017), 65. DOI: http://dx.doi.org/10.1145/3072959.3073620

Morten Bojsen-Hansen, Hao Li, and Chris Wojtan. 2012. Tracking surfaces with evolving topology. *ACM Trans. Graph.* 31, 4 (2012), 53–1.

Kenneth A. Brakke. 1992. The Surface Evolver. *Experimental Mathematics* 1, 2 (1992), 141–165.

Tyson Brochu, Christopher Batty, and Robert Bridson. 2010. Matching fluid simulation elements to surface geometry and topology. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 47.

Tyson Brochu and Robert Bridson. 2009. Robust topological operations for dynamic explicit surfaces. *SIAM Journal on Scientific Computing* 31, 4 (2009), 2472–2493.

Tyson Brochu, Todd Keeler, and Robert Bridson. 2012. Linear-time smoke animation with vortex sheet meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 87–95.

Oleksiy Busaryev, Tamal K. Dey, Huamin Wang, and Zhong Ren. 2012. Animating bubble interactions in a liquid foam. *ACM Trans. Graph.* 31, 4 (2012), 63.

Marcel Campen and Leif Kobbelt. 2010. Exact and robust (self-) intersections for polygonal meshes. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 397–406.

Jean-Marc Chomaz. 2001. The dynamics of a viscous soap film with soluble surfactant. *Journal of Fluid Mechanics* 442 (2001), 387–409. DOI: http://dx.doi.org/10.1017/S0022112001005213

Paul W Cleary, Soon Hyoung Pyo, Mahesh Prakash, and Bon Ki Koo. 2007. Bubbling and frothing liquids. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 97.

Y. Couder, J. M. Chomaz, and M. Rabaud. 1989. On the Hydrodynamics of Soap Films. In *Proceedings of the Eighth Annual International Conference of the Center for Nonlinear Studies on Advances in Fluid Turbulence*. North-Holland Publishing Co., NLD, 384–405.

Fang Da, Christopher Batty, and Eitan Grinspun. 2014. Multimaterial Mesh-Based Surface Tracking. *ACM Trans. on Graphics (SIGGRAPH 2014)* 33 (2014), 1–11.

Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double Bubbles Sans Toil and Trouble: Discrete Circulation-Preserving Vortex Sheets for Soap Films and Foams. *ACM Trans. on Graphics (SIGGRAPH 2015)* 34, 4 (2015), 149:1–149:9.

Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-Only Liquids. *ACM Trans. on Graphics (SIGGRAPH 2016)* (2016).

Fernando de Goes, Mathieu Desbrun, and Yiying Tong. 2016. Vector field processing on triangle meshes. In *ACM SIGGRAPH 2016 Courses*. ACM, 27.

Roman Durikovic. 2001. Animation of Soap Bubble Dynamics, Cluster Formation and Collision. *Comput. Graph. Forum* 20, 3 (2001), 67–76.

Gerhard Dziuk and Charles M. Elliott. 2013. Finite element methods for surface PDEs. *Acta Numerica* 22 (2013), 289–396. DOI: http://dx.doi.org/10.1017/S0962492913000056

David Eppstein. 2012. The Graphs of Planar Soap Bubbles. *CoRR* abs/1207.3761 (2012). arXiv:1207.3761 http://arxiv.org/abs/1207.3761

Feng Gao, Hussam Muhamedsalih, and Xiang Jiang. 2012. Surface and thickness measurement of a transparent film using wavelength scanning interferometry. *Optics express* 20 (09 2012), 21450–6. DOI: http://dx.doi.org/10.1364/OE.20.021450

Sanghyeon Gil, Yunji Seok, Kiyeol Park, Jaeseok Yoo, and Seongah Chin. 2019. Soap film flow and thickness for soap bubble rendering. In *25th ACM Symposium on Virtual Reality Software and Technology*. 1–2.

Andrew Glassner. 2000a. Soap bubbles. 1. *IEEE Computer Graphics and Applications* 20, 5 (2000), 76–84.

Andrew Glassner. 2000b. Soap bubbles: part 2. *IEEE Computer Graphics and Applications* 6 (2000), 99–109.

Shannon T Greenwood and Donald H House. 2004. Better with bubbles: enhancing the visual realism of simulated fluid. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 287–296.

David J. Hill and Ronald D. Henderson. 2016. Efficient Fluid Simulation on the Surface of a Sphere. *ACM Trans. Graph.* 35, 2, Article 16 (April 2016), 9 pages.

Jeong-Mo Hong, Ho-Young Lee, Jong-Chul Yoon, and Chang-Hun Kim. 2008. Bubbles alive. *ACM Trans. Graph.* 27, 3 (2008), 48:1–48:4.

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 Trans. on Graphics* 39, 4 (2020). DOI: http://dx.doi.org/10.1145/3386569.3392094

Michael Hutchings, Frank Morgan, Manuel Ritoré, and Antonio Ros. 2002. Proof of the Double Bubble Conjecture. *Annals of Mathematics* 155, 2 (2002), 459–489. http://www.jstor.org/stable/3062123

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

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.

Byungmoon Kim, Yingjie Liu, Ignacio Llamas, Xiangmin Jiao, and Jarek Rossignac. 2007. Simulation of Bubbles in Foam With The Volume Control Method. *ACM Trans. on Graphics (SIGGRAPH 2007)* 26, 3 (2007), 10.

ByungMoon Kim, Yingjie Liu, Ignacio Llamas, and Jaroslaw R Rossignac. 2005. *Flowfixer: Using bfecc for fluid simulation.* Technical Report. Georgia Institute of Technology.

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.

Timothy R Langlois, Changxi Zheng, and Doug L James. 2016. Toward animating water with complex acoustic bubbles. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 95.

Matthias Müller. 2009. Fast and robust tracking of fluid surfaces. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 237–245.

Maxim Olshanskii and Xianmin Xu. 2016. A Trace Finite Element Method for PDEs on Evolving Surfaces. *SIAM Journal on Scientific Computing* 39 (10 2016). DOI: http://dx.doi.org/10.1137/16M1099388

Alexander Oron, Stephen H Davis, and S George Bankoff. 1997. Long-scale evolution of thin liquid films. *Reviews of modern physics* 69, 3 (1997), 931.

Saket Patkar, Mridul Aanjaneya, Dmitriy Karpman, and Ronald Fedkiw. 2013. A hybrid Lagrangian-Eulerian formulation for bubble generation and dynamics. In *The ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA '13, Anaheim, CA, USA, July 19-21, 2013.* ACM Press, New York, NY, USA, 105–114.

Darko Pavić, Marcel Campen, and Leif Kobbelt. 2010. Hybrid booleans. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 75–87.

Argyrios Petras and S.J. Ruuth. 2016. PDEs on moving surfaces via the closest point method and a modified grid based particle method. *J. Comput. Phys.* 312 (02 2016). DOI: http://dx.doi.org/10.1016/j.jcp.2016.02.024

Osborne Reynolds. 1886. IV. On the theory of lubrication and its application to Mr. Beauchamp tower's experiments, including an experimental determination of the viscosity of olive oil. *Philosophical transactions of the Royal Society of London* 177 (1886), 157–234.

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.

Robert I. Saye and James A. Sethian. 2016. Multiscale modelling of evolving foams. *J. Comput. Phys.* 315 (2016), 273–301.

LW Schwartz and RV Roy. 1999. Modeling draining flow in mobile and immobile soap films. *Journal of colloid and interface science* 218, 1 (1999), 309–323.

F. Seychelles, Y. Amarouchene, M. Bessafi, and H. Kellay. 2008. Thermal Convection and Emergence of Isolated Vortices in Soap Bubbles. *Phys. Rev. Lett.* 100 (Apr 2008), 144501. Issue 14. DOI: http://dx.doi.org/10.1103/PhysRevLett.100.144501

Lin Shi and Yizhou Yu. 2004. Inviscid and incompressible fluid simulation on triangle meshes. *Computer Animation and Virtual Worlds* 15, 3-4 (2004), 173–181.

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

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

Nils Thürey, Chris Wojtan, Markus Gross, and Greg Turk. 2010. A multiscale approach to mesh-based surface tension flows. *ACM Transactions on Graphics* 29, 4 (2010), 10.

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

Huamin Wang, Gavin Miller, and Greg Turk. 2007. Solving general shallow wave equations on surfaces. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 229–238.

D. Weaire and R. Phelan. 1994. A counter-example to Kelvin's conjecture on minimal surfaces. *Philosophical Magazine Letters* 69, 2 (1994), 107–110. DOI: http://dx.doi.org/10.1080/09500839408241577 arXiv:https://doi.org/10.1080/09500839408241577

Chris Wojtan, Nils Thürey, Markus Gross, and Greg Turk. 2009. Deforming meshes that split and merge. *ACM Transactions on Graphics (TOG)* 28, 3 (2009), 76.

Chris Wojtan, Nils Thürey, Markus Gross, and Greg Turk. 2010. Physics-inspired topology changes for thin fluid features. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 50.

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.

Jihun Yu, Chris Wojtan, Greg Turk, and Chee Yap. 2012. Explicit mesh surfaces for particle based fluids. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 815–824.

Yizhong Zhang, Huamin Wang, Shuai Wang, Yiying Tong, and Kun Zhou. 2012. A deformable surface model for real-time water drop animation. *IEEE Transactions on Visualization and Computer Graphics* 18, 8 (2012), 1281–1289.

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

Qingnan Zhou, Eitan Grinspun, Denis Zorin, and Alec Jacobson. 2016. Mesh arrangements for solid geometry. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 39.

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