Chemomechanical Simulation of Soap Film Flow on Spherical Bubbles

WEIZHEN HUANG, JULIAN ISERINGHAUSEN, and TOM KNEIPHOF, University of Bonn, Germany
ZIYIN QU and CHENFANFU JIANG, University of Pennsylvania, USA
MATTHIAS B. HULLIN, University of Bonn, Germany

Fig. 1. Spatially varying iridescence of a soap bubble evolving over time (left to right). The complex interplay of soap and water induces a complex flow on the film surface, resulting in an ever changing distribution of film thickness and hence a highly dynamic iridescent texture. This image was simulated using the method described in this paper, and path-traced in Mitsuba [Jakob 2010] using a custom shader under environment lighting.

Soap bubbles are widely appreciated for their fragile nature and their colorful appearance. The natural sciences and, in extension, computer graphics, have comprehensively studied the mechanical behavior of films and foams, as well as the optical properties of thin liquid layers. In this paper, we focus on the dynamics of material flow within the soap film, which results in fascinating, extremely detailed patterns. This flow is characterized by a complex coupling between surfactant concentration and Marangoni surface tension. We propose a novel chemomechanical simulation framework rooted in lubrication theory, which makes use of a custom semi-Lagrangian advection solver to enable the simulation of soap film dynamics on spherical bubbles both in free flow as well as under body forces such as gravity or external air flow. By comparing our simulated outcomes to videos of real-world soap bubbles recorded in a studio environment, we show that our framework, for the first time, closely recreates a wide range of dynamic effects that are also observed in experiment.

CCS Concepts: • **Computing methodologies** → **Physical simulation**; *Reflectance modeling*.

ACM Reference Format:
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. Graph.* 39, 4, Article 1 (July 2020), 14 pages. https://doi.org/10.1145/3386569.3392094

1 MOTIVATION

The beauty of soap films and bubbles is of great appeal to people of all ages and cultures, and the scientific community is no exception. In the computer graphics community, it is now widely understood how films, bubbles and foams form, evolve and break. On the rendering side, it has become possible to recreate their characteristic iridescent appearance in physically based renderers. The main parameter governing this appearance, the thickness of the film, has so far only been driven using ad-hoc noise textures [Glassner 2000], or was assumed to be constant. The resulting renderings appear largely plausible but static, as they lack the rich dynamics known from real-world soap films.

With this paper, we aim to close this gap in order to achieve greater realism. We do so by contributing a chemomechanical framework targeted specifically at simulating the rich and detailed microscopic flow on spherical soap bubbles. Our framework employs a leading-order approximation for the soap film dynamics developed by [Chomaz 2001; Ida and Miksis 1998a]. A soap bubble is modeled as a two-dimensional flow on a static spherical surface with two associated scalar fields: the film thickness and the soap concentration. We are able to show that this state-of-the-art model, paired with a custom solver, is capable of expressing the intricate flows found on real-world soap bubbles (Figure 1) under the mutual influence of mechanical stress, film thickness and surfactant concentration as well as body and surface forces like gravity and air friction. Our simulation is performed on a staggered grid, using finite differences in space and time. An advection scheme based on BiMocq^2 [Qu et al. 2019] minimizes numerical dissipation in order to prevent fine details from washing out over time. The resulting thickness maps are presented in real time using a custom, very efficient polarization-aware spectral rendering scheme.

Besides the underlying physical model, our framework is enabled by the following key contributions:

- We propose a novel advection scheme for vector and scalar quantities in spherical coordinates. Our scheme, which constructs a local coordinate frame aligned with the direction of the flow, is

This work was supported by the European Research Council under ERC starting grant "ECHO", the National Science Foundation (CAREER IIS-1943199 and CCF-1813624) and the Department of Energy (ORNL 4000171342).
Authors' e-mail addresses: {whuang|iseringhausen|kneiphof}@cs.uni-bonn.de, {ziyinq|cffjiang}@seas.upenn.edu, hullin@cs.uni-bonn.de.

unconditionally stable and maintains continuous behavior even near and across the poles.
- We propose an implicit update step for the soap concentration, which avoids solving a stiff indefinite system and enables the use of significantly larger time steps when applying body forces.
- We introduce a novel real-time shader that is designed to reflect spectrally and polarization-dependent effects under environment lighting in a physically accurate manner. This even holds for contributions which are reflected multiple times in a spherical bubble.
- We investigate the influence of material parameters, geometric scale and external forces on the flow, and compare our results to real-world examples captured under lab conditions.

2 RELATED WORK

With their tendency to evolve into minimal surfaces, soap films embody a fundamental mathematical and physical principle in a way that is immediately relatable and fascinating to experts and laypersons alike. Consequently, they have inspired a large body of research in mathematics, physics and materials science. Some mathematicians even went so far as to use them as analog computers to solve mathematical minimization problems [Isenberg 1978]. Within the computer graphics community, the geometric properties of minimal surfaces as well as the formation, evolution and destruction of films, bubbles and foams have inspired a large number of groundbreaking works [Da et al. 2015; Glassner 2000; Ishida et al. 2017; Iwasaki et al. 2004; Kim et al. 2015; Ďurikovič 2001; Zhu et al. 2014].

Besides their geometry, the beauty of soap bubbles also stems from their dynamic iridescent patterns. The chaotic mixing, highly nonlinear vortices and turbulence of the fluid flow are not only visually interesting, but they have also led to a body of scientific work that is just as varied and colorful as its subject of study. Examples range from the visualization and study of 2D flow [Gharib and Derango 1989] or using soap bubbles as a small-scale surrogate model for planetary atmospheres [Meuel et al. 2013; Seychelles et al. 2008] via the visualization of sound and music [Gaulon et al. 2017] to using soap films as volatile display surfaces [Ochiai et al. 2013]. In computer graphics, the simple optical effect behind the characteristic iridescent colors (thin film interference) has long been understood and used [Belcour and Barla 2017; Glassner 2000; Iwasaki et al. 2004; Jaszkowski and Rzeszut 2003; Kneiphof et al. 2019; Smits and Meyer 1992; Sun 2006]. However, the film thickness, which is the main governing parameter besides the liquid's refractive index, has rarely been driven by proper physical simulation. While some works [Saye and Sethian 2013, 2016; Zhu et al. 2014] have coupled thickness in their models, they use it more or less as an intermediate variable that influences the macroscopic motion and serves as a bursting condition. Most recently, such models have been equipped with the ability to propagate turbulent flow across Plateau boundaries [Ishida et al. 2020]. In contrast, our goal is to simulate the rich and detailed dynamics of the microscopic flow that is observable through thin film interference, while staying as close as possible to a state-of-the-art physical model.

We turn to the fluid mechanics and physics communities, where several comprehensive models for soap film flow have been devised. Chomaz et al. [2001] derived a highly accurate model for the dynamics of a flat soap film, which is based on the asymptotic lubrication theory, assuming the thickness of the film is small compared to its lateral extent. The main contribution of that work is on the construction of similarities between soap film flows and compressible fluid flows in a planar, two-dimensional domain. The model provided by Ida and Miksis [1998a] is in principle capable of expressing general and time-varying three-dimensional soap films. In order to simulate flow using their model [Ida and Miksis 1998b], they employ a pseudospectral Chebychev spatial collocation and restrict the solution to the one-dimensional axisymmetric case.

Since the pioneering work by Foster and Metaxas [1996], physics-based animation of fluids has been an important topic in computer graphics due to its wide range of applications for capturing effects of smoke [Fedkiw et al. 2001], free surface flow [Foster and Fedkiw 2001], or fire [Nguyen et al. 2002]. We note that even lubrication theory, which forms the foundation of our framework, has at least once before been employed by other members of the computer graphics community: [Azencot et al. 2015] used it to simulate the flow of thin liquid films across solid surfaces. Regarding the geometric discretization, various possible choices exist [Ando et al. 2015; Bridson 2015; Ferstl et al. 2016; Ihmsen et al. 2014; Macklin and Müller 2013]. Grid-based Eulerian simulation of fluid remains popular and widely adopted due to its superior efficiency and versatility, despite its well-known problems in numerical diffusion. In graphics, the semi-Lagrangian advection scheme presented by Stam [1999] builds the foundation for many more advanced future developments on Eulerian fluids, including some recent impressively successful examples [Narain et al. 2019; Qu et al. 2019; Zehnder et al. 2018]. Indeed, the advection equation is such a mathematically simple, yet numerically challenging equation that consistently draws a lot of attention. The difficulty is pronounced at an even higher level when one tries to solve for fluid dynamics on a spherical geometry [Hill and Henderson 2016; Yang et al. 2019] due to the notoriously difficult "pole singularity problem" [Randall et al. 2002]. In this paper, we look into an even more challenging scenario where we need to efficiently and robustly advect multiple physical quantities related to the chemomechanical physics on a soap film.

3 PHYSICAL MODEL

The mechanical properties of soap solutions are characterized by the interplay of water and the soap dissolved in it. Soap molecules, which have a polar (hydrophobic) and a nonpolar (hydrophilic) end, tend to settle at the water surface, so that their hydrophobic part can avoid the contact with water. As a result, the soap concentration at surfaces is usually much higher than in the bulk fluid. Soap further acts as a surfactant, i.e., the presence of soap molecules reduces the surface tension of the fluid. When the distance between soap molecules at the surface increases, surface tension increases accordingly. By adding soap to water, it becomes possible to make bubbles that can last several seconds to minutes, since the surface tension prevents them from bursting. The resulting structure of soap films consists of three layers [Couder et al. 1989]: two water-air

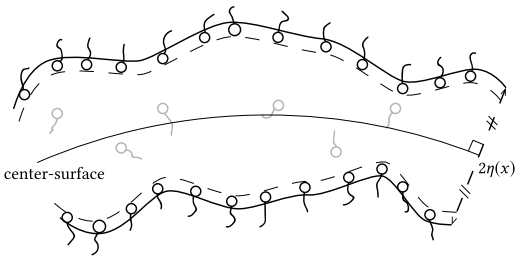


Fig. 2. Cross-section of a soap film (image adapted from Couder et al. [1989]). A thin layer of liquid (thickness 2η) is centered around a macroscopically defined surface. The polar chemistry of surfactant molecules causes them to concentrate at the liquid-air interface. We assume the number of soap molecules in the bulk fluid (here marked in gray) to be negligible.

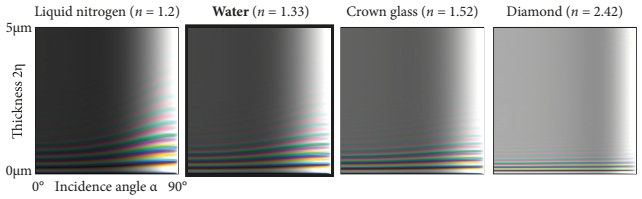


Fig. 3. Interference color of light reflected by a thin layer of dielectric as a function of film thickness and incidence angle. Shown are simulations with a spectral resolution of 5nm for materials of different refractive index, including the most relevant material for our purpose, water. Colors are scaled so that white corresponds to a reflectance of 100%. We refer the reader to established literature [Belcour and Barla 2017; Glassner 2000] on how to compute these colors. Somewhat counterintuitively, we note that the optical path difference *decreases* with increasing angle. Therefore, under oblique observation it takes a *thicker* film to produce the same color.

interfaces populated by soap molecules and a thin layer of bulk fluid in between (Figure 2). The thickness of a soap film is usually around 1µm, which explains the colorful interference between light reflected at the two interfaces (Figure 3).

As a soap solution's refractive index is only weakly affected by the soap concentration, the two dominant influences on the color of the film are its large-scale geometry and hence the viewing angle, and the spatially varying thickness of the fluid layer. For the purpose of this paper, we assume the shape of the bubble to be fixed. This leaves material transport within the film manifold as the main source of texture. In order to recreate the intricate dynamics found in real-world soap films, we have to understand, model and simulate this flow.

3.1 Governing equations in 3D

We formulate the fluid flow in terms of the incompressible three-dimensional Navier-Stokes equations,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} = \frac{1}{\rho}\nabla\cdot\sigma + \mathbf{f}, \tag{1a}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{1b}$$

where ∇ is the nabla operator in three dimensions, $\mathbf{u}$ is the fluid velocity, σ is the Cauchy stress tensor [Batchelor 1967, Ch. 1.3], ρ is the mass density, and $\mathbf{f}$ represents body accelerations such as gravity and air friction.

At a film surface, the stress condition applies as [Couder et al. 1989; Ida and Miksis 1998a]

$$\sigma\cdot\mathbf{n} = (2\mathscr{C}\gamma - p_a)\mathbf{n} + \nabla_s\gamma, \tag{2}$$

where $\mathbf{n}$ is the outward normal vector at the respective surface, p_a the air pressure, γ the surface tension, and $2\mathscr{C} = -\nabla\cdot\mathbf{n}$ is twice the mean surface curvature. (In general, the values of these quantities differ between one surface of the film and the other.) The 2D gradient operator ∇_s within the surface acts on a scalar field Φ as

$$\nabla_s\Phi = \nabla\Phi - \mathbf{n}(\mathbf{n}\cdot\nabla\Phi). \tag{3}$$

The surface tension depends on the surfactant concentration Γ, i.e., the concentration of soap molecules on the surface,

$$\gamma = \gamma_a - \gamma_r\Gamma, \tag{4}$$

where γ_a is the surface tension of pure water, and γ_r accounts for the elasticity of the film. In the small concentration range, γ_r is considered to be constant [Couder et al. 1989].

Finally, the surfactant concentration Γ is driven by the advection-diffusion equation

$$\frac{\partial\Gamma}{\partial t} + \nabla_s\cdot(\mathbf{u}\Gamma) = D_s\big(\nabla_s\big)^2\Gamma, \tag{5}$$

with D_s being the diffusivity for surfactant molecules.

3.2 Thin-film analysis and governing equation on spheres

In the following, we will restrict ourselves to spherical bubbles with radius R. For small (centimeter-sized) bubbles, this approximation is reasonable. It is thus convenient to parameterize the problem in spherical coordinates (r,θ,ϕ). The fluid velocity $\mathbf{u}$ then reads as

$$\mathbf{u} = (u_r, u_\theta, u_\phi)^\top. \tag{6}$$

We further note that the thickness of a soap bubble is very small compared to its lateral extent. Using lubrication theory [Oron et al. 1997; Reynolds 1886], we reduce the three-dimensional problem to a two-dimensional one. The extent of the film along the normal direction (the thickness) is introduced as a variable rather than a third dimension of the simulation domain. Suppose the inner and outer sides of the bubble are symmetrically deformed with half thickness η to either side (Figure 2), then the kinematic condition (see supplemental material) describing the interaction between the time and spatially varying film thickness $\eta = \eta(\theta,\phi,t)$ and the velocity $\mathbf{u}$ at the interface $r = R \pm \eta$ can be written as

$$\frac{\partial\eta}{\partial t} + \frac{u_\theta}{R}\frac{\partial\eta}{\partial\theta} + \frac{u_\phi}{R\sin\theta}\frac{\partial\eta}{\partial\phi} = \pm u_r. \tag{7}$$

With the mean half thickness η_0 and the expansion parameter $\epsilon = \eta_0/R$, we non-dimensionalize the variables as

$$\begin{gathered}\eta = \eta_0\eta', \quad u_\theta = Uu_\theta', \quad u_\phi = Uu_\phi', \quad \sigma = \frac{\mu U}{R}\sigma', \\ u_r = \epsilon U u_r', \;\; r = R + \epsilon R r', \;\; t = \frac{R}{U}t', \;\; \Gamma = \Gamma_0\Gamma', \;\; p = \frac{\mu U}{R\epsilon}p',\end{gathered} \tag{8}$$

where U is the characteristic velocity and Γ_0 is the mean surfactant concentration. We substitute these variables in Equations (1), (2), (4), (5) and (7), expand $\mathbf{u}$, Γ and η asymptotically with a power series, and drop all terms except those with leading order of ϵ (see

[Chomaz 2001; Ida and Miksis 1998a] for more details). Note that our non-dimensionalization is adapted to spherical coordinates and thus differs from the literature examples. The nabla operator ∇ now only acts within the surface,

$$\nabla = \nabla_s = \mathbf{e}_\theta \frac{\partial}{\partial \theta} + \frac{1}{\sin\theta}\mathbf{e}_\phi \frac{\partial}{\partial \phi}, \tag{9}$$

where $\mathbf{e}_\theta$ and $\mathbf{e}_\phi$ are the respective basis vectors. The governing equations are thus reduced to

$$\begin{cases} \dfrac{D\mathbf{u}'}{Dt'} = -\dfrac{M}{\eta'}\nabla\Gamma' + \mathbf{f} + Re^{-1}\mathbf{V}, & \text{(10a)} \\ \dfrac{D\Gamma'}{Dt'} = -\Gamma'\nabla\cdot\mathbf{u}' + D_s'\nabla^2\Gamma', & \text{(10b)} \\ \dfrac{D\eta'}{Dt'} = -\eta'\nabla\cdot\mathbf{u}', & \text{(10c)} \end{cases}$$

where $M = \Gamma_0\gamma_r/\rho\eta_0U^2$ is the Marangoni number, $Re = {}^{UR\rho}/_{\mu}$ is the Reynolds number, μ and ρ are the dynamic viscosity and the mass density of the soap solution, respectively. The thermodynamic quantity $\gamma_r = \bar{R}T$ combines gas constant $\bar{R}$ and temperature T [Couder et al. 1989]. $D_s' = {}^{D_s}/_{UR}$ is the scaled diffusivity, and the total derivative

$$\frac{D}{Dt'} = \frac{\partial}{\partial t'} + \mathbf{u}'\cdot\nabla. \tag{11}$$

The vector $\mathbf{V} = (V_\theta, V_\phi)^\top$ represents viscous terms including second order terms as

$$\frac{\partial^2 u_\theta'}{\partial\phi^2}, \frac{\partial\eta'}{\partial\theta}\frac{\partial u_\theta'}{\partial\theta}, \dots$$

The complete terms are provided in the supplementary document, Equation (84). For readability, we drop the primes from this point onward. Further, within the scope of this paper, we assume that the soap molecules are not diffusive ($D_s = 0$) and that viscosity can be neglected. Readers interested in viscous effects may refer to Section 7.1. In spherical coordinates, the total derivative of $\mathbf{u}$ can be written as

$$\begin{aligned} \frac{D\mathbf{u}}{Dt} = &\left(\frac{\partial u_\theta}{\partial t} + u_\theta\frac{\partial u_\theta}{\partial\theta} + \frac{u_\phi}{\sin\theta}\frac{\partial u_\theta}{\partial\phi} - \frac{u_\phi^2}{\tan\theta}\right)\mathbf{e}_\theta \\ &+\left(\frac{\partial u_\phi}{\partial t} + u_\theta\frac{\partial u_\phi}{\partial\theta} + \frac{u_\phi}{\sin\theta}\frac{\partial u_\phi}{\partial\phi} + \frac{u_\theta u_\phi}{\tan\theta}\right)\mathbf{e}_\phi \end{aligned} \tag{12}$$

and its divergence as

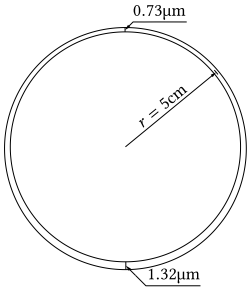
$$\nabla\cdot\mathbf{u} = \frac{1}{\sin\theta}\left[\frac{\partial u_\phi}{\partial\phi} + \frac{\partial}{\partial\theta}\left(u_\theta\sin\theta\right)\right]. \tag{13}$$

The derivatives of a scalar field Φ (which could either be the thickness η or the soap concentration Γ), are

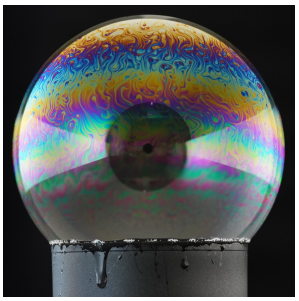
$$\nabla\Phi = \left(\frac{\partial\Phi}{\partial\theta}, \frac{1}{\sin\theta}\frac{\partial\Phi}{\partial\phi}\right)^\top, \tag{14}$$

$$\frac{D\Phi}{Dt} = \frac{\partial\Phi}{\partial t} + u_\theta\frac{\partial\Phi}{\partial\theta} + \frac{u_\phi}{\sin\theta}\frac{\partial\Phi}{\partial\phi}, \tag{15}$$

$$\nabla^2\Phi = \frac{1}{\sin\theta}\left[\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial\Phi}{\partial\theta}\right) + \frac{\partial}{\partial\phi}\left(\frac{1}{\sin\theta}\frac{\partial\Phi}{\partial\phi}\right)\right]. \tag{16}$$



(a) Typical dimensions



(b) Soap bubble photograph

Fig. 4. Under the influence of gravity and with surface tension as opposing force, bubbles assume an equilibrium state where the film thickness gradually increases from top to bottom. This is reflected by the horizontal fringe pattern observed on real-world bubbles.

There are mainly two contributions for the evolution of soap film thickness η and surfactant concentration Γ: they are passively advected by the flow field [Yang et al. 2019], but also affected by inflow or outflow as expressed by the divergence terms in Equations (10b) and (10c). Unlike the full 3D incompressible flow (Equation (1b)), the 2D flow within a thin film behaves like a compressible, elastic medium.

3.3 Surface and body forces

The most important forces governing the motion of a soap film within its manifold are surface tension, gravity and air friction. The interaction between gravity and surface tension will cause thinner films to float upwards and thicker films to drop downwards, so that a soap bubble (or film) at its equilibrium state always assumes a wedge shape (Figure 4). Let g be the gravitational acceleration scaled by ${}^{U^2}/_{R}$. If we assume the north pole of the bubble to be pointing upwards, then the gravity vector $\mathbf{g}$ in the spherical coordinate system $(\mathbf{e}_\theta, \mathbf{e}_\phi)$ is $\mathbf{g} = (g\sin\theta, 0)^\top$.

As a soap bubble is very thin, the film is easily set into motion by tangential air flow. If the surrounding air is still, it damps the flow motion. For simplicity, we assume a linear Stokes drag $\mathbf{f}_{\text{air}} = ({}^{Cr}/_{\eta})(\mathbf{u}_{\text{air}} - \mathbf{u})$, with Cr being the drag coefficient. Taking gravity and air friction into account, Equation (10) becomes

$$\begin{cases} \dfrac{D\mathbf{u}}{Dt} = -\dfrac{M}{\eta}\nabla\Gamma + \dfrac{Cr}{\eta}(\mathbf{u}_{\text{air}} - \mathbf{u}) + \mathbf{g}, & \text{(17a)} \\ \dfrac{D\Gamma}{Dt} = -\Gamma\nabla\cdot\mathbf{u}, & \text{(17b)} \\ \dfrac{D\eta}{Dt} = -\eta\nabla\cdot\mathbf{u}. & \text{(17c)} \end{cases}$$

Interestingly, from Equation (17a), we observe that surface forces as surface tension and air drag are divided by the film thickness, so that thinner films are more easily driven into motion, whereas body forces like gravity act throughout the volume of the body and thus do not depend on the thickness.

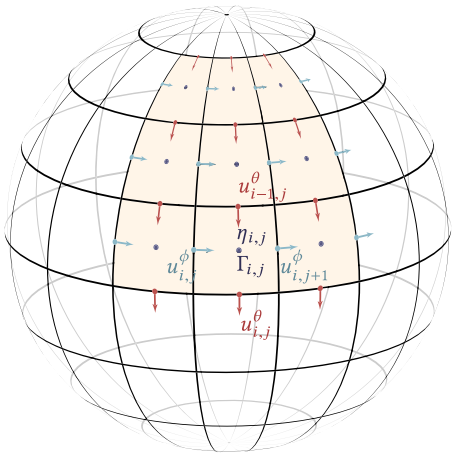


Fig. 5. The staggered spherical grid. Film thickness η and soap concentration Γ are evaluated at the cell center, while the θ and ϕ components of the velocity vector $\mathbf{u}$ are sampled at the midpoints of cell boundaries.

4 METHOD

In this section we develop novel spatial and temporal discretization schemes for the governing equations. In particular, via using a staggered spherical grid (Section 4.1), we develop an unconditionally stable advection scheme (Section 4.2) that can smoothly propagate flow across the poles, as well as a projection-like implicit solver for handling chemomechanical forces (Section 4.3).

4.1 Spatial discretization

We discretize the spherical domain with a staggered grid [Yang et al. 2019], where velocities and scalar quantities are stored at different locations (illustrated in Figures 5 and 6). This allows the accurate evaluation of the concentration gradient $\nabla\Gamma$ and the velocity divergence $\nabla \cdot \mathbf{u}$ using central differences without the formation of checkerboard patterns [Bridson 2015, Ch. 2.4]. Also, a regular grid discretization makes it possible to build a symmetric positive definite linear system that can be solved using the conjugate gradient method.

The velocity vector is split in two components, u_ϕ and u_θ, which are located at the center of the cell boundaries, whereas the concentration Γ and the thickness η are sampled at the cell center. Assuming the staggered grid consists of $N_\theta \times N_\phi$ cells, then the dimension of u_ϕ, Γ and η is $N_\theta \times N_\phi$, and the dimension of u_θ is $(N_\theta - 1) \times N_\phi$.

Quantities that do not lie exactly on the respective grid points are bi-linearly interpolated between neighboring grid points (Figure 7). Special care needs to be taken when sampling near the poles, as neighborhood relations reach across the pole. At this point, both $\mathbf{e}_\theta$ and $\mathbf{e}_\phi$ experience a sign change, so velocity samples drawn from across the pole have to be negated.

4.2 Advection

Taking an operator splitting approach, we first solve the material derivative (D/Dt) parts of $\mathbf{u}$, η, and Γ in Equation (17) along the velocity field through advection. Afterwards, we treat the remaining force terms in treated in a separate step (Section 4.3).

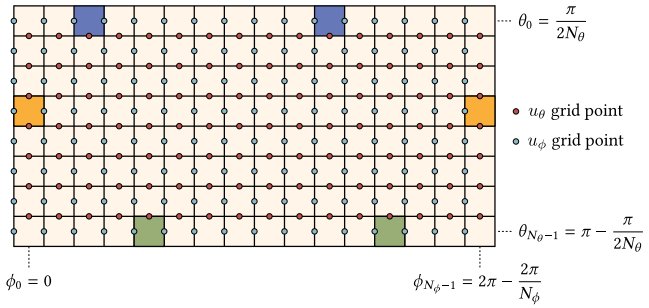


Fig. 6. The staggered spherical grid unrolled in θ and ϕ direction, with the north pole at $\theta = 0$ and the south pole at $\theta = \pi$. Cells with the same color indicate direct neighborhood. Specifically, in ϕ-direction, periodic boundary conditions are employed, while in θ-direction, ϕ is shifted by 180° when crossing the pole. The velocity vector is not stored explicitly at the poles and instead we sample the vectors close to the pole bi-linearly from neighboring cells.

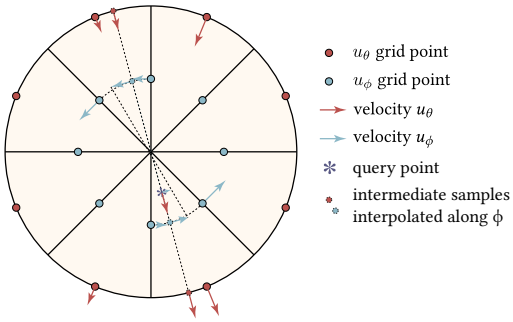


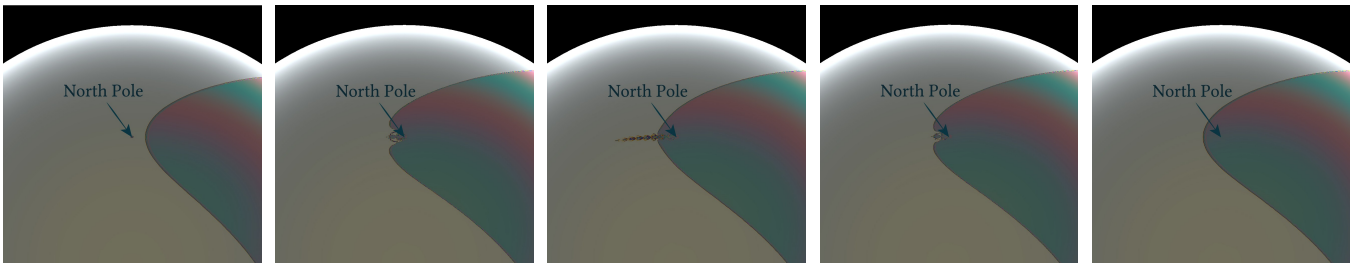
Fig. 7. Sampling velocities bi-linearly at the pole. As samples are interpolated across the pole, it is important to take into account the sign flip of $\mathbf{u}$ caused by the parameterization singularity.

On the spherical domain, the pure advection of a time-dependent scalar field $\Phi(\mathbf{x}, t)$, $\mathbf{x} = (\theta, \phi)^\top$, along the velocity field $\mathbf{u}(\mathbf{x}, t) = (u_\theta(\mathbf{x}, t), u_\phi(\mathbf{x}, t))^\top$ can be written as the initial value problem [Pironneau et al. 1992]

$$\begin{aligned} \frac{D\Phi}{Dt} &= \frac{\partial \Phi}{\partial t} + u_\theta \frac{\partial \Phi}{\partial \theta} + \frac{u_\phi}{\sin\theta} \frac{\partial \Phi}{\partial \phi} = 0, \\ \Phi(\mathbf{x}, 0) &= \Phi^0(\mathbf{x}). \end{aligned} \tag{18}$$

We seek to evaluate the advected quantity at a grid point $\mathbf{x}_{ij} = (\theta_i, \phi_j)^\top$. In keeping with standard practice in fluid simulation, we introduce a virtual particle that passes the grid point at time t, and trace it backward in time. This results in the time-dependent trajectory $X^{(\mathbf{x}_{ij}, t)}(\tau)$ for the seed point $(\mathbf{x}_{ij}, t)$ and time parameter $\tau < t$. Substituting this trajectory into Equation (18) yields

$$\begin{aligned} 0 &= \frac{D}{D\tau} \Phi \left(X^{(\mathbf{x}_{ij}, t)}(\tau), \tau \right) \Big|_{\tau = t} \\ &\approx \frac{1}{\Delta t} \left(\Phi(X^{(\mathbf{x}_{ij}, t)}(t)) - \Phi(X^{(\mathbf{x}_{ij}, t)}(t - \Delta t)) \right), \end{aligned} \tag{19}$$



(a) Before (b) Single step (order-1, global) (c) RK3 (order-3, global) (d) [Yang et al. 2019] (order-2, global) (e) Ours (order-2, aligned)

Fig. 8. A scalar field with a sharp transition is advected across the north pole. Shown are film thickness maps before (a) and after (b–e) passing the pole, advected over 4 time steps using different methods. Using a velocity-aligned coordinate frame introduced in Section 4.2, our method is the only one to solve this problem without visible artifacts.

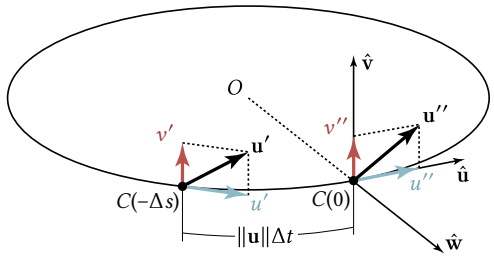


Fig. 9. A single advection step for the velocity field on the sphere, executed in a local, flow-aligned coordinate frame (great circle C). A detailed description of the procedure is provided in Section 4.2.

where the finite difference corresponds to a discretization of the time domain around t with step size Δt. A particle that undergoes pure advection experiences Φ as being constant. We exploit this property to approximate the value for $\Phi(X^{(\mathbf{x}_{ij},t)}(t))$ using a sample taken a step of Δt backward through the flow field. A numerical integration step like Euler or Runge-Kutta can be used to obtain $\Phi(X^{(\mathbf{x}_{ij},t)}(t-\Delta t))$.

Advecting vectors on a sphere is much more challenging. As $\partial \mathbf{e}_\theta/\partial\phi$ and $\partial \mathbf{e}_\phi/\partial\phi$ are not equal to zero (Appendix A) as they would be on a Cartesian grid, Equation (12) contains the additional terms $-u_\phi^2/\tan\theta$ and $u_\theta u_\phi/\tan\theta$ that are not present in Equation (18). Yang et al. [2019] advect vectors in a scalar-like manner with $(u_\theta, u_\phi/\sin\theta)^\top$, and treat the two additional terms as body forces in an additional backward Euler intergration step. Their method does not work well in practice when additional force terms are present. Instead, we are able to perform unconditionally stable vector advection in a single step. We note that at the equator $\theta = \pi/2$, where $\tan^{-1}\theta = 0$, these two extra terms vanish. There, the advection of a vector field falls back to the known case of advecting a scalar field. We exploit this insight in order to obtain this desirable property anywhere on the sphere. By constructing a local coordinate system at each grid point that aligns with the velocity, we can treat the quantities there as if they were on the equator.

For each grid point $\mathbf{x}_{ij} = X^{(\mathbf{x}_{ij},t)}(t)$, we draw a great circle on the sphere that passes through this point and is tangent to the velocity $\mathbf{u}$ at this point (see Figure 9). We denote the unit vector in $\mathbf{u}$-direction to be $\hat{\mathbf{u}}$, and the unit vector pointing from the sphere center O to $\mathbf{x}_{ij}$ to be $\hat{\mathbf{w}}$, and the binormal unit vector to be $\hat{\mathbf{v}} = \hat{\mathbf{w}} \times \hat{\mathbf{u}}$. Then the great circle can be parameterized by

$$C(s) = \sin(s)\hat{\mathbf{u}} + \cos(s)\hat{\mathbf{w}}, \quad s \in \mathbb{R}; \tag{20}$$

i.e., changing the arc parameter s translates the current grid point $\mathbf{x}_{ij} = C(0) = \hat{\mathbf{w}}$ back or forward in time with unit velocity. Following $\mathbf{u}$ backward for a time step Δt results in a change in the arc parameter of $-\Delta s = -\|\mathbf{u}\|\Delta t$, taking us to the point

$$C(-\Delta s) = \sin(-\Delta s)\hat{\mathbf{u}} + \cos(-\Delta s)\hat{\mathbf{w}}. \tag{21}$$

This results in the following single-step advection scheme (Figure 9):

(1) Evaluate the velocity $\mathbf{u}$ and establish the great circle C in its direction.
(2) Perform an interpolated lookup of the velocity $\mathbf{u}'$ at $C(-\Delta s)$ using the technique described in Section 4.1.
(3) Decompose $\mathbf{u}'$ into tangent and binormal components u' and v' with respect to the great circle C at this point.
(4) Move components back along the circle to $\mathbf{x}_{ij} = C(0)$ with $v'' = v', u'' = u'$.
(5) Project the advected quantity $\mathbf{u}'' = v''\hat{\mathbf{v}} + u''\hat{\mathbf{u}}$ back to global spherical coordinate system to obtain u_θ^* and u_ϕ^*.

The main distinction of this approach to existing work is in the choice of coordinate frame. Rather than operating in the *global* spherical coordinate system, it locally creates an orthonormal coordinate frame that is defined by the velocity vector and therefore *data-aligned*. Since the method only interpolates values from the last step, it is unconditionally stable.

When higher accuracy is desired, the same principle can also be used to implement higher-order schemes involving multi-step updates. As an example, we lay out the construction of a 2^{nd}-order "half-step" scheme (Figure 21) in Appendix B, which was used to generate the results shown throughout this paper.

By design, our advection scheme does not produce artifacts when advecting quantities (scalars or vectors) around the poles, see Figure 8. [Yang et al. 2019] adapts the advection scheme in [Hill and Henderson 2016] to staggered grids, where boundary conditions are introduced at both poles to remove *singularities*. However, according to [Hill and Henderson 2016], their method is not free of *artifacts*: small disturbances will appear near the pole due to variation of grid

spacing (the $1/\sin\theta$ term becomes prohibitively large near the pole). By constructing a local frame in our advection scheme, the $1/\sin\theta$ term always takes the value 1, so the variation in grid spacing does not affect the advection. Moreover, the so-called geometry term in [Hill and Henderson 2016; Yang et al. 2019] that is caused by coordinate orientation changes in curvilinear coordinate systems (see Appendix A) is in our case implicitly handled when transforming from global to local coordinates, and does not need to be solved separately. By performing our global-to-local coordinate transformation everywhere on the sphere, we show that all the discretization points on the sphere can be treated equally and there is no need for any *special pole treatment*. We expect this method to perform equally well as on a Cartesian grid.

4.2.1 Preserving details. The interpolated look-up in our advection scheme causes numerical diffusion known from all semi-Lagrangian methods. This is acceptable for $\mathbf{u}$ and Γ, since both PDEs include diffusive terms (see Equation (10)). However, the transport equation for η is non-diffusive. Therefore, it is important to prevent high-frequency details in the film thickness from blurring out over time.

To achieve this goal, we make use of BiMocq^2 [Qu et al. 2019] and extend the method to spherical coordinates. Essentially, BiMocq^2 keeps a backward mapping

$$\mathcal{X}(\mathbf{x}(T)) : \mathbf{x}(T) \rightarrow \mathbf{x}(t_0) \tag{22}$$

which maps a spatial point $\mathbf{x}(T)$ back to its position at the initial time t_0, as well as a forward mapping

$$\mathcal{Y}(\mathbf{x}(t_0)) : \mathbf{x}(t_0) \rightarrow \mathbf{x}(T) \tag{23}$$

which maps a spatial point at the initial position $\mathbf{x}(t_0)$ to its current position at $t = T$. Instead of repeatedly blurring the features from the last time step, we acquire the initial state directly from the backward mapping, which corresponds to the pure advection part D/Dt. The additional changes ($-\eta\nabla\cdot\mathbf{u}$ in our case) are accumulated along the forward mapping and added to the acquired value from backward mapping. At each simulation step, both mappings are updated via the advection method in Section 4.2, and the coordinates are interpolated using spherical linear interpolation. When the distortion between forward and backward mapping becomes too large, both mappings are re-initialized. This, however, introduces a small amount of numerical diffusion. We found that re-initializing both mappings when the distortion is larger than $\pi/128$ provides a good trade-off between sharpness and noise. For other implementation details, such as error correction, we refer the reader to the original paper.

4.3 Concentration splitting

After solving the advection part $D/Dt = 0$, we now deal with force and divergence terms in the right hand sides of Equation (17). Soap film exhibits elastic properties similar to a mass-spring system. Solving such systems using an explicit time integrator would require prohibitively small step sizes to achieve a stable simulation. Instead, we construct a projection-like implicit system for Γ and $\mathbf{u}$. To keep the system linear, we still solve for η explicitly.

We temporally discretize the continuous equations into

$$\begin{cases} \dfrac{\mathbf{u}-\mathbf{u}^*}{\Delta t} = -\dfrac{M}{\eta^*}\nabla\Gamma + \dfrac{Cr}{\eta^*}(\mathbf{u}_{\text{air}}-\mathbf{u}) + \mathbf{g}, & \text{(24a)} \\ \dfrac{\Gamma-\Gamma^*}{\Delta t} = -\Gamma^*\nabla\cdot\mathbf{u}, & \text{(24b)} \\ \dfrac{\eta-\eta^*}{\Delta t} = -\eta^*\nabla\cdot\mathbf{u}, & \text{(24c)} \end{cases}$$

where Γ^*, η^*, and $\mathbf{u}^*$ denote the respective quantities after applying the advection step. First, we solve for Γ by rewriting Equation (24a) and applying divergence to both sides of the result,

$$\nabla\cdot\mathbf{u} = \nabla\cdot\frac{\eta^*\mathbf{u}^* + Cr\Delta t\mathbf{u}_{\text{air}} + \Delta t\eta^*\mathbf{g}}{\eta^* + Cr\Delta t} - M\Delta t\nabla\cdot\frac{\nabla\Gamma}{\eta^* + Cr\Delta t}. \tag{25}$$

Afterwards, we combine Equations (24b) and (25) and eliminate $\nabla\cdot\mathbf{u}$,

$$\begin{aligned} &\frac{\Gamma}{\Gamma^*\Delta t} - M\Delta t\nabla\cdot\frac{\nabla\Gamma}{\eta^* + Cr\Delta t} \\ =&\frac{1}{\Delta t} - \nabla\cdot\frac{\eta^*\mathbf{u}^* + Cr\Delta t\mathbf{u}_{\text{air}} + \Delta t\eta^*\mathbf{g}}{\eta^* + Cr\Delta t}. \end{aligned} \tag{26}$$

Finally, we express this linear system as a sparse matrix (Appendix C), solve it for Γ using a preconditioned conjugate gradient method [Naumov et al. 2015], and update $\mathbf{u}^{n+1}$ and η^{n+1} using Equations (24a) and (24c). Note that this system is strictly symmetric positive definite unless Γ^* approaches infinity, in which case we end up with a Poisson equation that is closely related to pressure projection in nearly incompressible mixed finite elements.

Our implicit treatment of the concentration based on its evolution Equation (24b) allows us to take significantly larger time steps compared to what could be permitted when treating the forces explicitly. Our method also avoids solving an extremely stiff indefinite system as in standard Newton-based elasticity solvers [Gast et al. 2015].

4.4 Implementation and runtime performance

We implemented our method using CUDA and AmgX [Naumov et al. 2015], and executed it on an NVIDIA GeForce GTX 1080 graphics card. A typical resolution for our simulation grid is 1024×2048 with a step size of 0.002 s. At this setting, a single time step typically takes 16–17 conjugate gradient iterations and 1.1 s to execute. The bulk of the compute time is spent on divergence/force calculation and advection with 75% and 25%, respectively.

5 SOAP BUBBLE RENDERING

The iridescent effects produced by thin films are wave-optical effects that arise from constructive and destructive interference of light waves. To compute the light power that is reflected or transmitted when interacting with a thin film, we need to consider the corresponding complex amplitudes of the electromagnetic wave. This section describes a real-time renderer for spherical soap bubbles that is specifically targeted at the correct handling of polarization and spectral sampling.

5.1 Thin film model

We follow the modeling of Belcour and Barla [2017] for the reflection and transmission through a single thin film layer. In our case, the

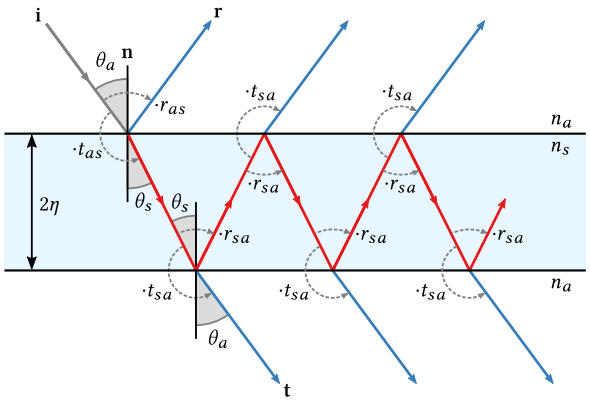


Fig. 10. Light paths in soap film. Light is refracted into the film, and attenuated at each film interaction. The emitted light paths at the top and bottom of the film produce interference, respectively.

light interacts multiple times with the soap bubble surface on a given light path, so their analytical solution to the spectral integration does not directly apply here. After each reflection, we would have to retract to an RGB representation of the light, introducing an error that increases with each interaction with the soap bubble. Therefore, we numerically integrate the compounded reflectance and transmittance for each light path over the wavelengths λ, which we sample at 5 nm intervals, and compute the fractional light power that is carried by each light path for each wavelength.

Material model. The surface of the soap bubble is modeled locally as a thin film with parallel interfaces, sandwiched between two layers of air with refractive index $n_a = 1$. The refractive index of the soap water in between is $n_s = 1.33$. For a single interaction of the light with the soap bubble, the film thickness 2η is assumed to be constant. For a given light direction $\mathbf{i}$ and surface normal $\mathbf{n}$, we define the angle of incidence θ_a in air via $\cos\theta_a = \mathbf{i} \cdot \mathbf{n}$. Upon refraction into the soap film, the angle at which the light travels is found via Snell's law: $n_a \sin\theta_a = n_s \sin\theta_s$. Since the film interfaces are assumed to be parallel, the angle at which the light is refracted out of the film at both interfaces equals θ_a, and the direction of the light transmitted through the film is uninterrupted from air to air. We assume that the light enters and leaves the film at the same location on a macroscopic scale, since the film thickness is much smaller than the lateral extent of the soap bubble.

Thin film reflectance. The polarization-dependent reflectance R and transmittance T are the ratios of outgoing to incoming light powers at an interface between two media. In order to compute these values for two media separated by a thin film, we consider the complex-valued electro-magnetic wave amplitudes. The complex-valued reflection coefficient r and transmission coefficient t describe the ratios of outgoing to incoming wave amplitudes. Since the power carried by a light wave is proportional to the square of the wave amplitude, we have $R = |r|^2$ and $T = |t|^2$. In addition to the power ratio, they also encode a phase shift of the light wave, which leads to constructive and destructive interference when two light paths interfere and their reflection or transmission coefficients are added.

The reflectance and transmittance produced by an interaction with the film is computed for each wavelength and polarization by taking all light paths inside the thin film (shown in Figure 10) into account and accumulating their wave amplitudes. The light waves are affected by the reflection coefficients r_{as}, r_{sa} and transmission coefficients t_{as}, t_{sa}, defined by Fresnel's equations [Born and Wolf 1970], where r_{as} and t_{as} act at the air-to-soap interface and r_{sa} and t_{sa} on the soap-to-air interface. The light waves of each path are also affected by a wavelength-dependent phase shift, induced by the difference in path length between successively emitted light rays at each film interface. This difference in path length is known as the optical path difference $\mathcal{D} = 4\eta n_s \cos\theta_s$, which introduces a phase shift $\Delta\phi = 2\pi\frac{\mathcal{D}}{\lambda}$ of a light path with respect to its predecessor. The phase shifts with respect to the first light ray accumulate linearly, such that the k-th ray is phase-shifted by $k\Delta\phi$. Summing the contributions from the infinitely many emitted light rays at the top and bottom interface yields the reflectance R,

$$\begin{aligned} R(\lambda) &= \left| r_{as} + \sum_{k=0}^{\infty} t_{as} r_{sa} \left(r_{sa}^2 e^{i\Delta\phi} \right)^k e^{i\Delta\phi} t_{sa} \right|^2 \\ &= \left| r_{as} + \frac{t_{as} r_{sa} t_{sa} e^{i\Delta\phi}}{1 - r_{sa}^2 e^{i\Delta\phi}} \right|^2, \end{aligned} \tag{27}$$

and transmittance T of the thin-film

$$\begin{aligned} T(\lambda) &= \left| \sum_{k=0}^{\infty} t_{as} \left(r_{sa}^2 e^{i\Delta\phi} \right)^k t_{sa} \right|^2 \\ &= \left| \frac{t_{as} t_{sa}}{1 - r_{sa}^2 e^{i\Delta\phi}} \right|^2. \end{aligned} \tag{28}$$

Since we are not dealing with total internal reflections, $|r_{sa}| < 1$ holds and taking the limit of the geometric series yields a closed form solution for each wavelength.

5.2 Soap bubble ray tracing

The wave nature of light only needs to be taken into account when computing the reflectance and transmittance for a single interaction with the soap bubble. Since soap bubbles are inherently transparent, many light paths contribute to a given view ray. To compute the light transport along these paths, we multiply the reflectance and transmittance produced at each soap bubble interaction along the path. The light transport $\mathcal{R}^{(n)}$ for the n-th order light path is defined as

$$\mathcal{R}^{(0)} = R^{(0)} \quad \text{and} \quad \mathcal{R}^{(k)} = T^{(0)} \prod_{i=1}^{k-1} R^{(i)} T^{(k)}, \tag{29}$$

for $k \geq 1$, where $R^{(i)}$ and $T^{(i)}$ are the iridescent reflectance and transmittance produced at the i-th interaction with the soap bubble along the path traced backwards from the observer. The reflectance $R^{(i)}$ and transmittance $T^{(i)}$ produced by different interactions along a path change only due to differences in film thickness. The angle of incidence is constant for all interactions along a light path (see Figure 11) and the index of refraction is constant as well.

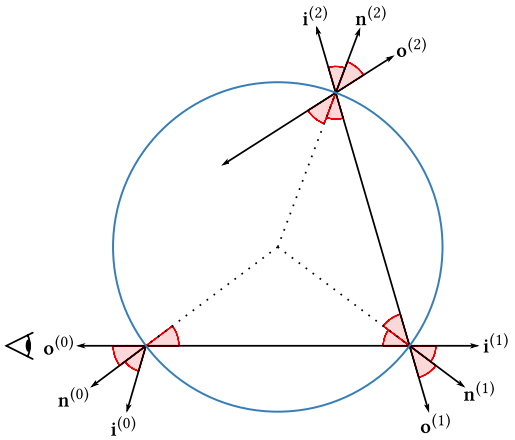


Fig. 11. Light paths in a spherical soap bubble. All angles marked in red are equal.

Polarization. Due to the spherical geometry of the soap bubble, the incoming and outgoing light directions for all interactions lie in the same plane. This implies that the light polarization basis (the decomposition into s and p component) does not change between successive soap film interactions. Therefore, we are able to correctly handle polarization effects by first evaluating Equation (29) for each polarization direction independently, and then averaging both frames.

Tracing rays. For the computation of the light paths through the soap bubble, illustrated in Figure 11, we exploit the assumption that the soap bubble is spherical. We evaluate the first $N = 8$ light paths through the soap bubble. Since we assume distant illumination, we do not track the world-space location of the film interactions, and only consider the relevant directions: the incoming light direction $\mathbf{i}^{(i)}$ of the i-th order light path to sample the environment map, the surface normal $\mathbf{n}^{(i)}$ to sample the film thickness 2η at the i-th interaction, and the (virtual) viewing direction $\mathbf{o}^{(i)}$, which is used to compute the directions for the succeeding interaction with the soap film. Given these values for the i-th film interaction, the directions for the $i+1$-th interaction are defined as

$$\begin{aligned} \mathbf{n}^{(i+1)} &= \mathbf{n}^{(i)} - 2\mathbf{n}^{(i)} \cdot \mathbf{o}^{(i)}\mathbf{o}^{(i)}, \\ \mathbf{i}^{(i+1)} &= -\mathbf{o}^{(i)}, \\ \mathbf{o}^{(i+1)} &= \mathbf{o}^{(i)} - 2\mathbf{o}^{(i)} \cdot \mathbf{n}^{(i+1)}\mathbf{n}^{(i+1)}. \end{aligned} \tag{30}$$

The surface normal $\mathbf{n}^{(i+1)}$ is defined by a reflection of $-\mathbf{n}^{(i)}$ at the outgoing light direction $\mathbf{o}^{(i)}$, and the new outgoing light direction $\mathbf{o}^{(i+1)}$ is then the reflection of the incoming light direction $\mathbf{i}^{(i+1)}$ at the surface normal $\mathbf{n}^{(i+1)}$ (see Figure 11).

Spectral integration. To produce an sRGB color image, we have to integrate the response to the respective color-matching functions s_j for $j \in \{R, G, B\}$ of the color space:

$$L_{o,j} = \int s_j(\lambda) \cdot \sum_{n=0}^{\infty} \mathcal{R}_n(\lambda) L_i^{(n)}(\lambda)\, \mathrm{d}\lambda, \tag{31}$$

Table 1. Typical values and ranges of dimensional parameters used in our simulation. For values associated with air and soap solution, such as density and viscosity, we take common literature values for air and water, respectively, under standard conditions.

Description	Symbol	Value / Range	Unit
Bubble radius[a]	R	0.02–0.1	m
Mean half thickness[a]	η_0	4×10^{-7} –1×10^{-6}	m
Characteristic velocity	U	1	$\mathrm{m\,s^{-1}}$
Mean soap concentration[b]	Γ_0	1×10^{-8} –1×10^{-6}	$\mathrm{mol\,m^{-2}}$
Surface tension of water-air interface	γ_a	7.275×10^{-2}	$\mathrm{N\,m^{-1}}$
Gas constant	$\bar{R}$	8.3144598	$\mathrm{J\,mol^{-1}\,K^{-1}}$
Room Temperature	T	298.15	K
Water mass density	ρ	997	$\mathrm{kg\,m^{-3}}$
Water dynamic viscosity	μ	8.9×10^{-4}	Pa s
Air density	ρ_a	1.184	$\mathrm{kg\,m^{-3}}$
Air kinematic viscosity	ν_a	1.562×10^{-5}	$\mathrm{m^2\,s^{-1}}$
Gravitational acceleration	G	9.8	$\mathrm{m\,s^{-2}}$
Surfactant diffusivity[c]	D_s	(0)	$\mathrm{m^2\,s^{-1}}$

[a]Empirical values.
[b]Taken from [Couder et al. 1989, Figure 1(a)] in the low concentration range.
[c]We assume advection to be the dominant transport mechanism.

Table 2. Typical values of dimensionless parameters.

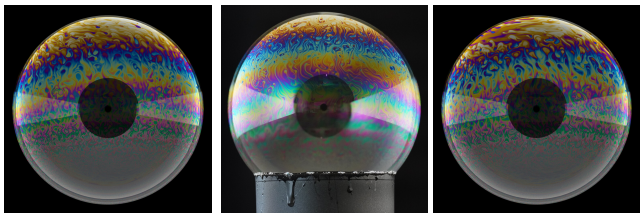
Description	Symbol	Definition	Value
Expansion parameter	ϵ	$\frac{\eta_0}{R}$	1×10^{-5}
Marangoni number	M	$\frac{\Gamma_0 \bar{R} T}{\rho \eta_0 U^2}$	0.83
Reynolds number	Re	$\frac{URp}{\mu}$	5.6×10^4
Drag coefficient	Cr	$\frac{\rho_a \sqrt{\nu_a R}}{\rho \eta_0 \sqrt{U}}$	2.1
Scaled gravitational acceleration	g	$\frac{GR}{U^2}$	0.49

where $L_{o,j}$ is the integrated response for the j-th color channel, and $L_i^{(n)}$ is the incoming light of the n-th light path. Since we use an RGB environment map for illumination, $L_i^{(n)}$ is not know. For each color channel j, we assume $L_i^{(n)} = L_{i,j}^{(n)}$ to be constant. Under this assumption Equation (31) simplifies to

$$L_{o,j} \approx \sum_{n=0}^{N} L_{i,j}^{(n)} \cdot \int s_j(\lambda) \mathcal{R}_n(\lambda)\, \mathrm{d}\lambda. \tag{32}$$

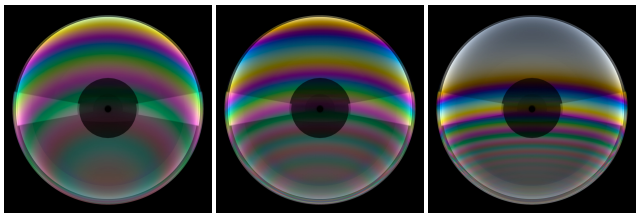
6 RESULTS

In the following, we perform a selection of synthetic experiments and discuss the influence of the most important parameters and variables. From various sources, we attempted to gather as realistic a set of parameters as possible. The numbers used for simulations throughout this section, as well as sources for the more exotic values, are listed in Tables 1 and 2.



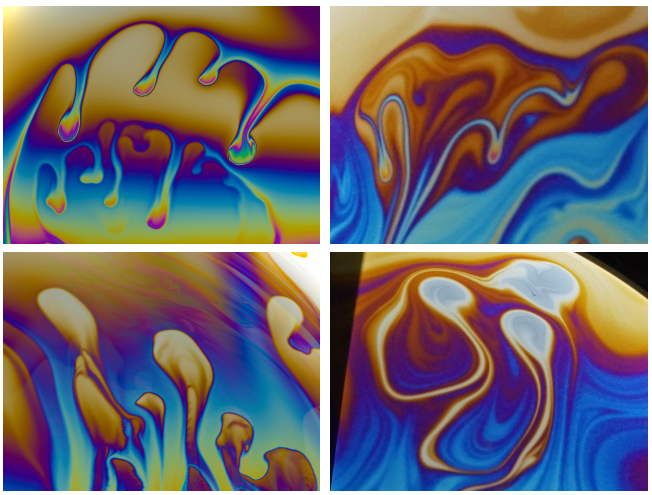
$\Gamma_0 = 3.33 \times 10^{-8}$ $R = 0.0225$ Reference photograph $\Gamma_0 = 6.67 \times 10^{-8}$ $R = 0.045$

Fig. 12. Influence of soap concentration Γ_0 and bubble radius R on the thickness gradient in equilibrium state. The ratio of both parameters, Γ_0/R, is kept constant for both simulations, causing a similar appearance.



$R = 0.025$ $R = 0.05$ $R = 0.1$

Fig. 13. Equilibrium state as a function of R, for a film initialized with soap concentration $\Gamma_0 = 6.67 \times 10^{-8}$ and thickness $\eta_0 = 4 \times 10^{-7}$ and relaxed under standard gravity. For larger bubbles, the gravity drag causes a stronger displacement of material from top to bottom.



(a) Simulation (b) Photograph

Fig. 14. Under the influence of gravity, thicker (heavier) regions form downward-moving tears; thinner (lighter) regions rise from the bottom. These effects can be observed both in simulation (a) and experiment (b).

6.1 Mean surfactant concentration and bubble radius

According to the momentum equation (17a), a soap film under gravity but without other sources of excitation has its equilibrium state at

$$-\frac{M}{\eta}\frac{\partial \Gamma}{\partial \theta} + g\sin\theta = 0. \tag{33}$$

The other two Equations (17b) and (17c) can be rewritten as

$$\frac{D(\Gamma/\eta)}{Dt} = 0, \tag{34}$$

i.e., Γ/η remains constant. If the simulation is started with uniform surfactant concentration and thickness, we can non-dimensionalize the variables so that $\Gamma(t=0) = \eta(t=0) = 1$. Combining the above two equations, this yields

$$-\frac{M}{\eta}\frac{\partial \eta}{\partial \theta} + g\sin\theta = 0, \tag{35}$$

the solution of which is

$$\eta = \frac{\pi}{\int_0^\pi e^{-\frac{g\cos\theta}{M}} d\theta} e^{-\frac{g\cos\theta}{M}}. \tag{36}$$

From this, we can draw at least three conclusions:

- Soap films under the influence of gravity are thinner at the top and thicker at the bottom, leading to colorful bands on soap bubbles (Figure 12). As noted in Figure 3, the film color is also influenced by the viewing angle. On a bubble, the bands are bent downwards; on a flat film they appear horizontal.
- A constant ratio g/M will lead to the same equilibrium state (albeit through a different dynamic process). From the definitions of M and g (Table 2), this is equivalent to keeping R/Γ_0 constant.
- The larger R/Γ_0, the thinner the film is at the top, and thicker at the bottom.

This expected behavior is confirmed in experiment and simulation (Figures 12 and 13).

6.2 Gravity and buoyancy

The momentum equation of a soap bubble

$$\frac{D\mathbf{u}}{Dt} = -\frac{M}{\eta}\nabla\Gamma + \frac{Cr}{\eta}(\mathbf{u}_{\text{air}} - \mathbf{u}) + \mathbf{g} \tag{37}$$

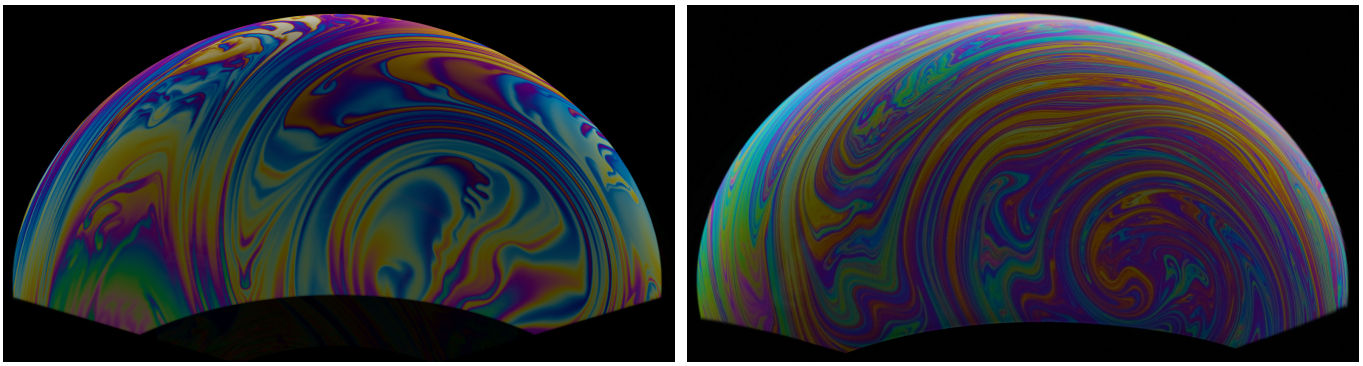
has great resemblance with the compressible Navier-Stokes equation

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla^2\mathbf{u} + \frac{\mu}{3\rho}\nabla(\nabla\cdot\mathbf{u}) + \mathbf{g}, \tag{38}$$

where the surfactant concentration Γ takes the role of pressure p and the variable thickness η substitutes the variable density ρ. In fact, just as smoke with smaller density flows upwards in the air, thinner soap film regions also tend to flow upwards. This is confirmed by our observation. As thinner regions on a soap bubble flow upwards (and thicker regions downwards), they form drop-shaped "islands", and leave "rivers" behind (Figure 14).

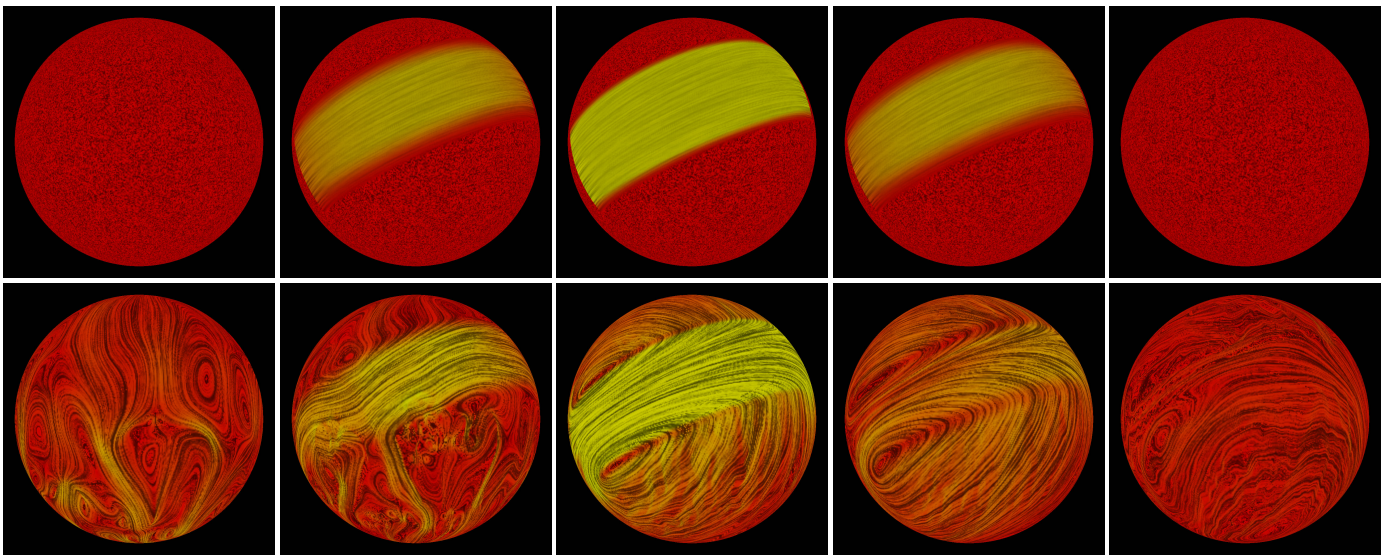
6.3 Air friction

Soap films are highly susceptible to air flow, and assume beautiful patterns in windy environments. When a bubble is blown, a rapidly rotating wind field is produced inside and induces a shear motion on the soap film. Advection along the air flow results in thin stripes that remain stable even after the external influence has stopped (Figure 15). See Figure 16 for a false-color visualization of the external air flow and the resulting velocity field in multiple time steps of an experiment.



(a) Simulation (b) Photograph

Fig. 15. Inflating a soap bubble with a straw generates a rotating "ball of air" trapped inside the bubble. The resulting velocity gradient leads to a shear motion and the formation of thin stripes.



(a) $t = 0.52\,\mathrm{s}$ (b) $t = 1.00\,\mathrm{s}$ (c) $t = 2.04\,\mathrm{s}$ (d) $t = 3.08\,\mathrm{s}$ (e) $t = 3.56\,\mathrm{s}$

Fig. 16. False-color visualization of a time-varying airflow (top) and the resulting velocity field (bottom) on a soap bubble using line integral convolution [Cabral and Leedom 1993]. From left to right, the correlation of the lines shows the direction of the respective vector field, while the magnitude is encoded in the color. (a) After initialization with a noise texture, the fluid sags down under the influence of gravity. (b)–(d) Over a corridor on the surface, air gradually starts flowing and slows down again. The soap fluid follows the excitation. (e) After the air has stopped flowing, the bubble remains in a rotating motion.

6.4 Evaporation

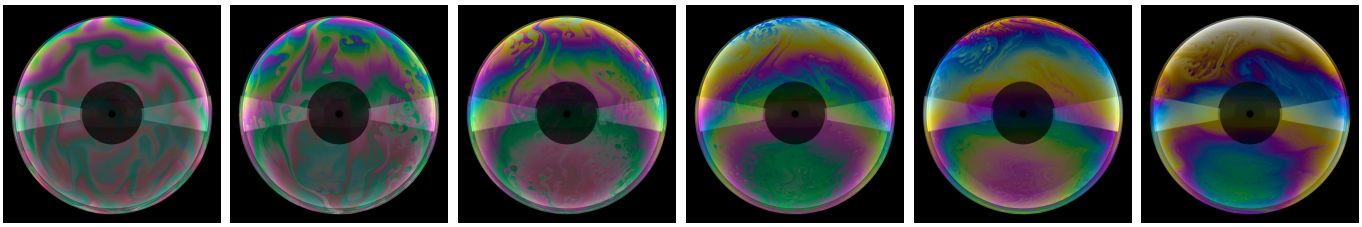
Due to evaporation, a soap bubble exposed to air becomes thinner and thinner and eventually breaks down. Since evaporation mostly depends on the exposed surface (which is constant), we model this effect by subtracting a small constant amount of η at each simulation step. Once a point on the surface reaches thickness zero, the simulation is terminated. Our model in its current form does not support the simulation of bursting bubbles. Figure 17 shows a soap bubble over its whole lifetime. Starting from a random thickness distribution, thicker regions are moving downwards and horizontal color bands. Shortly before bursting, the top of the bubble becomes very thin and exhibits a gray appearance.

6.5 Real-world experiments

To capture stills and videos of real-world bubbles under laboratory conditions, we constructed simple studio environments consisting of 1200 mm×600 mm LED panels, black theater curtain and a Sony ILCE-7RM3 system camera with a ZEISS Batis 135 mm $f/2.8$ lens. We use Pustefix brand soap solution for all experiments.

7 DISCUSSION AND FUTURE WORK

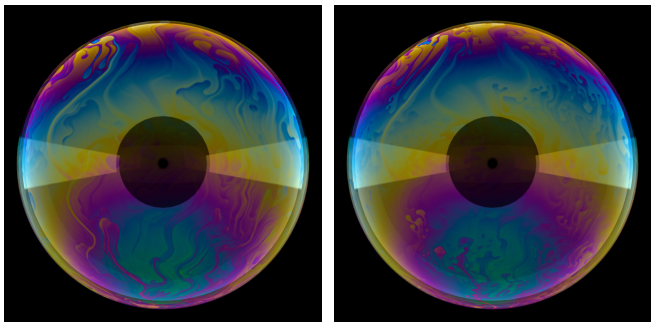
We have been able to show that our model and solver, which is fast and stable, can recreate the most prominent effects found on spherical soap bubbles in the real world. An obvious next step will be to look into more general cases, like complex film shapes or groups of bubbles. Although we focus on spherical geometry in this paper, the



(a) t=1 s (b) t=3 s (c) t=5 s (d) t=7 s (e) t=9 s (f) t=11 s

Fig. 17. Life of a soap bubble. The bubble was initialized with a Perlin noise texture and excited by curl-noise air flow. As water evaporates and the film becomes thinner, the colorful bands move gradually downwards and the top of the bubble fades to gray.

Fig. 18. A vertical soap film with marginal regeneration. Picture taken from [Nierstrasz and Frens 1999].



(a) Viscous soap film (b) Inviscid soap film

Fig. 19. Two simulations with/without viscosity term $Re^{-1}\mathbf{V}$, otherwise under the same condition and after same frame numbers. A viscous film tends to keep its texture longer in shape and has a reduced tendency to break into fractal structures.

ideas underlying our scheme are not limited to spherical domains. The model in [Ida and Miksis 1998a] is valid for general manifolds; following their derivation, one arrives exactly at Equation (10), with the definition of the differential operators adjusted to the corresponding curvilinear coordinate frame. Furthermore, the idea of our advection scheme is independent of the underlying manifold shape, as long as a proper local coordinate frame is constructed. Finally, the special force and divergence terms treatment in Equation (26) holds for arbitrary shapes and can be expressed as a sparse matrix, as long as the neighborhood of each point is well-defined. However, nice-to-have properties (such as the matrix being symmetric positive definite) might be lost in other types of meshes or grids. Consequently, our scheme should not be difficult to generalize to bubbles that are diffeomorphic to a sphere. Groups of bubbles with Plateau borders, as well as bounded films, are not manifolds and hence have to be left to future consideration. The treatment of film boundaries deserves particular attention also because a complicated mechanism called *marginal regeneration* [Isenberg 1978] causes the film to become even thinner at the boundaries, producing regions that flow upwards erratically (see Figure 18).

7.1 Viscous film

Soap films with larger viscosity tend to be more stable and last longer, which can be achieved by adding glycerin to home-made soap solution. Some commercial soap solution also includes additional formula to make it more viscous. The soap solution we used in experiments, for example, has a dynamic viscosity of 1.2×10^{-1} Pa s, which is about 100 times greater than that of water and thus shows different dynamics. The example in Figure 19 was obtained by adding a very basic explicit step to compute viscosity; however, this is slow and unstable and thus not included in our standard solver.

7.2 Black film

As a film keeps thinning through evaporation or gravity drag, at some point it becomes so thin that destructive interference takes place, and the film appears completely black. The thickness is then about 5 nm–30 nm [Isenberg 1978; Rücker 1877]. At such a small scale, molecular forces come into play, such as Van der Waals attraction, electrostatic repulsion, and Born repulsion. These molecular forces cause black film to be surprisingly stable and form sharply defined "islands" within the colorful film (Figure 20). In future work, it will be interesting to include such molecular forces in an extended model.

REFERENCES

Ryoichi Ando, Nils Thuerey, and Chris Wojtan. 2015. A stream function solver for liquid simulations. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9.

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

G. K. Batchelor. 1967. *An Introduction to Fluid Dynamics*. Cambridge University Press.

Laurent Belcour and Pascal Barla. 2017. A Practical Extension to Microfacet Theory for the Modeling of Varying Iridescence. *ACM Trans. Graph.* 36, 4, Article 65 (July

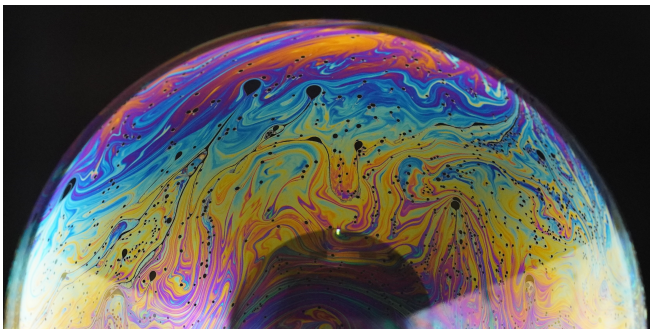


Fig. 20. Black film on a soap bubble (photograph)

2017), 14 pages. https://doi.org/10.1145/3072959.3073620

Max Born and Emil Wolf. 1970. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light.* Pergamon Press. 40 pages.

Robert Bridson. 2015. *Fluid simulation for computer graphics.* AK Peters/CRC Press.

Brian Cabral and Leith Casey Leedom. 1993. Imaging Vector Fields Using Line Integral Convolution. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (Anaheim, CA) *(SIGGRAPH '93)*. Association for Computing Machinery, New York, NY, USA, 263–270. https://doi.org/10.1145/166117.166151

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

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 Trans. on Graphics (SIGGRAPH 2015)* (2015).

Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. 2001. Visual simulation of smoke. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 15–22.

Florian Ferstl, Ryoichi Ando, Chris Wojtan, Rüdiger Westermann, and Nils Thuerey. 2016. Narrow band FLIP for liquid simulations. 35, 2 (2016), 225–232.

Nick Foster and Ronald Fedkiw. 2001. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 23–30.

Nick Foster and Dimitri Metaxas. 1996. Realistic animation of liquids. *Graphical models and image processing* 58, 5 (1996), 471–483.

Theodore F Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M Teran. 2015. Optimization integrator for large time steps. *IEEE transactions on visualization and computer graphics* 21, 10 (2015), 1103–1115.

C Gaulon, C Derec, T Combriat, Philippe Marmottant, and F Elias. 2017. Sound and vision: visualization of music with a soap film. *European Journal of Physics* 38, 4 (2017), 045804.

Morteza Gharib and Philip Derango. 1989. A liquid film (soap film) tunnel to study two-dimensional laminar and turbulent shear flows. *Physica D: Nonlinear Phenomena* 37, 1-3 (1989), 406–416.

Andrew Glassner. 2000. Soap Bubbles: Part 2. *IEEE Computer Graphics and Applications* 20, 6 (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.

MP Ida and Michael J Miksis. 1998a. The dynamics of thin films I: General theory. *SIAM J. Appl. Math.* 58, 2 (1998), 456–473.

MP Ida and Michael J Miksis. 1998b. The dynamics of thin films II: Applications. *SIAM J. Appl. Math.* 58, 2 (1998), 474–500.

Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. 2014. SPH Fluids in Computer Graphics. In *Eurographics 2014 - State of the Art Reports*, Sylvain Lefebvre and Michela Spagnuolo (Eds.). The Eurographics Association. https://doi.org/10.2312/egst.20141034

Cyril Isenberg. 1978. *The science of soap films and soap bubbles.* Tieto Cleveton, UK.

Sadashige Ishida, Peter Synak, Fumiya Narita, Toshiya Hachisuka, and Chris Wojtan. 2020. A Model for Soap Film Dynamics with Evolving Thickness. *ACM Trans. on Graphics* 39, 4 (2020). https://doi.org/10.1145/3386569.3392405

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

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.

Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

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

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

Tom Kneiphof, Tim Golla, and Reinhard Klein. 2019. Real-time Image-based Lighting of Microfacet BRDFs with Varying Iridescence. *Computer Graphics Forum* 38, 4 (2019).

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

Tinihau Meuel, Yong Liang Xiong, Patrick Fischer, Charles-Henri Bruneau, Miloud Bessafi, and Hamid Kellay. 2013. Intensity of vortices: from soap bubbles to hurricanes. *Scientific reports* 3 (2013), 3455.

Rahul Narain, Jonas Zehnder, and Bernhard Thomaszewski. 2019. A Second-Order Advection-Reflection Solver. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–14.

Maxim Naumov, M Arsaev, Patrice Castonguay, J Cohen, Julien Demouth, Joe Eaton, S Layton, N Markovskiy, István Reguly, Nikolai Sakharnykh, et al. 2015. AmgX: A library for GPU accelerated algebraic multigrid and preconditioned iterative methods. *SIAM Journal on Scientific Computing* 37, 5 (2015), S602–S626.

Duc Quang Nguyen, Ronald Fedkiw, and Henrik Wann Jensen. 2002. Physically based modeling and animation of fire. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 721–728.

Vincent Adriaan Nierstrasz and Gert Frens. 1999. Marginal regeneration and the Marangoni effect. *Journal of colloid and interface science* 215, 1 (1999), 28–35.

Yoichi Ochiai, Alexis Oyama, Takayuki Hoshi, and Jun Rekimoto. 2013. Theory and Application of the Colloidal Display: Programmable Bubble Screen for Computer Entertainment. In *Advances in Computer Entertainment*, Dennis Reidsma, Haruhiro Katayose, and Anton Nijholt (Eds.). Springer, 198–214.

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.

Olivier Pironneau, J Liou, and T Tezduyar. 1992. Characteristic-Galerkin and Galerkin/least-squares space-time formulations for the advection-diffusion equation with time-dependent domains. *Computer Methods in Applied Mechanics and Engineering* 100, 1 (1992), 117–141.

Ziyin Qu, Xinxin Zhang, Ming Gao, Chenfanfu Jiang, and Baoquan Chen. 2019. Efficient and conservative fluids using bidirectional mapping. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 128.

David A Randall, Todd D Ringler, Ross P Heikes, Phil Jones, and John Baumgardner. 2002. Climate modeling with spherical geodesic grids. *Computing in Science & Engineering* 4, 5 (2002), 32–41.

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.

AW Rücker. 1877. On Black Soap Films. *Nature* 16 (1877), 331–333.

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. https://doi.org/10.1126/science.1230623 The authors protest being forced to cite this work as condition for acceptance. Its substance is a proper subset of the more recent and more comprehensive [Saye and Sethian 2016].

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

F Seychelles, Y Amarouchene, Miloud Bessafi, and H Kellay. 2008. Thermal convection and emergence of isolated vortices in soap bubbles. *Physical review letters* 100, 14 (2008), 144501.

Brian E. Smits and Gary W. Meyer. 1992. *Newton's Colors: Simulating Interference Phenomena in Realistic Image Synthesis.* Springer Berlin Heidelberg, Berlin, Heidelberg, 185–194. https://doi.org/10.1007/978-3-662-09287-3_13

Jos Stam. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 121–128.

Yinlong Sun. 2006. Rendering Biological Iridescences with RGB-Based Renderers. *ACM Trans. Graph.* 25, 1 (Jan. 2006), 100–129. https://doi.org/10.1145/1122501.1122506

Roman Ďurikovič. 2001. Animation of Soap Bubble Dynamics, Cluster Formation and Collision. *Comput. Graph. Forum* 20 (09 2001), C–67–C–75. https://doi.org/10.1111/1467-8659.00499

Bowen Yang, William Corse, Jiecong Lu, Joshuah Wolper, and Chenfanfu Jiang. 2019. Real-Time Fluid Simulation on the Surface of a Sphere. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 1, Article 4 (June 2019), 17 pages. https://doi.org/10.1145/3320285

Jonas Zehnder, Rahul Narain, and Bernhard Thomaszewski. 2018. An advection-reflection solver for detail-preserving fluid simulation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–8.

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), 111.

A MATERIAL DERIVATIVE IN SPHERICAL COORDINATES

The material derivative describes the rate of change of a quantity moving with a time-dependent velocity field $\mathbf{u}$. We denote a dimensionless scalar quantity in spherical coordinates as $\Phi(\theta, \phi, t)$. Applying the chain rule yields

$$\frac{D\Phi}{Dt} = \frac{\partial \Phi}{\partial t} + \frac{\partial \Phi}{\partial \theta}\frac{d\theta}{dt} + \frac{\partial \Phi}{\partial \phi}\frac{d\phi}{dt}. \tag{39}$$

The dimensionless velocities in spherical coordinates are defined as

$$u_\theta = \frac{d\theta}{dt}, \qquad u_\phi = \sin\theta \frac{d\phi}{dt}. \tag{40}$$

Note that we included the sphere radius R in our non-dimensionalization of t in Section 3.2, thus we do not need to take account of the sphere radius when taking the derivatives. Substituting Equation (40) in Equation (39) gives

$$\frac{D\Phi}{Dt} = \frac{\partial \Phi}{\partial t} + u_\theta \frac{\partial \Phi}{\partial \theta} + \frac{u_\phi}{\sin\theta}\frac{\partial \Phi}{\partial \phi}. \tag{41}$$

The material derivative of a vector quantity, for example velocity, is similarly written as

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{u}}{\partial \theta}\frac{d\theta}{dt} + \frac{\partial \mathbf{u}}{\partial \phi}\frac{d\phi}{dt}, \tag{42}$$

where

$$\frac{\partial \mathbf{u}}{\partial t} = \frac{\partial u_\theta}{\partial t}\mathbf{e}_\theta + \frac{\partial u_\phi}{\partial t}\mathbf{e}_\phi + u_\theta \frac{\partial \mathbf{e}_\theta}{\partial t} + u_\phi \frac{\partial \mathbf{e}_\phi}{\partial t}. \tag{43}$$

To evaluate the remaining two partial derivatives, we take a step back to look at the derivatives of unit vectors in sphere coordinates. Neglecting the radial (r-dependent) component, they are

$$\frac{\partial \mathbf{e}_\theta}{\partial \theta} = 0, \quad \frac{\partial \mathbf{e}_\theta}{\partial \phi} = \cos\theta \mathbf{e}_\phi, \quad \frac{\partial \mathbf{e}_\phi}{\partial \theta} = 0, \quad \frac{\partial \mathbf{e}_\phi}{\partial \phi} = -\cos\theta \mathbf{e}_\theta. \tag{44}$$

Thus,

$$\frac{\partial \mathbf{u}}{\partial \theta} = \frac{\partial u_\theta}{\partial \theta}\mathbf{e}_\theta + \frac{\partial u_\phi}{\partial \theta}\mathbf{e}_\phi + u_\theta \frac{\partial \mathbf{e}_\theta}{\partial \theta} + u_\phi \frac{\partial \mathbf{e}_\phi}{\partial \theta} = \frac{\partial u_\theta}{\partial \theta}\mathbf{e}_\theta + \frac{\partial u_\phi}{\partial \theta}\mathbf{e}_\phi, \tag{45}$$

and

$$\begin{aligned}\frac{\partial \mathbf{u}}{\partial \phi} &= \frac{\partial u_\theta}{\partial \phi}\mathbf{e}_\theta + \frac{\partial u_\phi}{\partial \phi}\mathbf{e}_\phi + u_\theta \frac{\partial \mathbf{e}_\theta}{\partial \phi} + u_\phi \frac{\partial \mathbf{e}_\phi}{\partial \phi} \\ &= \frac{\partial u_\theta}{\partial \phi}\mathbf{e}_\theta + \frac{\partial u_\phi}{\partial \phi}\mathbf{e}_\phi + u_\theta \cos\theta \mathbf{e}_\phi - u_\phi \cos\theta \mathbf{e}_\theta.\end{aligned} \tag{46}$$

Combining Equations (40), (43), (45) and (46) we obtain

$$\begin{aligned}\frac{D\mathbf{u}}{Dt} = &\left(\frac{\partial u_\theta}{\partial t} + u_\theta \frac{\partial u_\theta}{\partial \theta} + \frac{u_\phi}{\sin\theta}\frac{\partial u_\theta}{\partial \phi} - \frac{u_\phi^2}{\tan\theta}\right)\mathbf{e}_\theta \\ &+\left(\frac{\partial u_\phi}{\partial t} + u_\theta \frac{\partial u_\phi}{\partial \theta} + \frac{u_\phi}{\sin\theta}\frac{\partial u_\phi}{\partial \phi} + \frac{u_\theta u_\phi}{\tan\theta}\right)\mathbf{e}_\phi.\end{aligned} \tag{47}$$

B 2ND-ORDER HALF-STEP UPDATE

Based on the single-step scheme detailed in Section 4.2, we construct a multi-step scheme inspired by second-order Runge Kutta that can deliver higher accuracy (Figure 21). First, the velocity vector sampled at the grid point (red) defines a great circle (also red) which is used to construct a local coordinate frame. Following this direction backward by a *half* time step, a second velocity vector (blue) is sampled. After transforming this vector back to the original point, it generates a second great circle (also blue). A *full* backward step along this circle takes us to the look-up location (∗) for the value that is advected to the grid point.

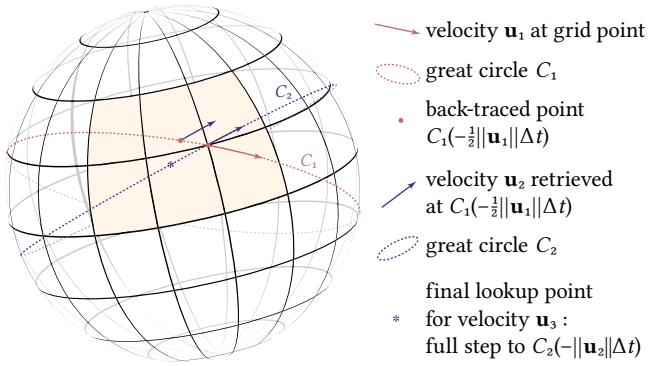


Fig. 21. Construction of a multi-step scheme similar to an order-2 Runge-Kutta update.

C LINEAR SYSTEM

We solve Equation (26) on a grid of dimension $m \times n$ and cell size $\Delta s \times \Delta s$ in matrix form $A\mathbf{\Gamma} = \mathbf{b}$, where

$$\mathbf{\Gamma} = \left[\Gamma_{0,0}, \ldots, \Gamma_{m-1,n-1}\right]^\top \in \mathrm{R}^{mn}, \quad \boldsymbol{b} = \left[b_{0,0}, \ldots, b_{m-1,n-1}\right]^\top \in \mathrm{R}^{mn}$$

and $A \in \mathrm{R}^{mn \times mn}$ is a sparse, block diagonal matrix with five elements in each row, corresponding to the cell itself and its four direct neighbors. In order to make A symmetric positive definite, we multiply both sides of Equation (26) by $\sin\theta$. Then, in each row, the entries of A are given by

$$\begin{aligned}A_{i,\mathrm{mid}} = \quad & \sin\theta(\Delta t \Gamma^*_{\theta,\phi})^{-1} + M\Delta t \Delta s^{-2} \\ & \left[\frac{\sin^{-1}\theta}{\eta^*_{\theta,\phi-\frac{\Delta s}{2}} + Cr\Delta t} + \frac{\sin^{-1}\theta}{\eta^*_{\theta,\phi-\frac{\Delta s}{2}} + Cr\Delta t} + \right. \\ & \left. \frac{\sin\left(\theta + \frac{\Delta s}{2}\right)}{\eta^*_{\theta+\frac{\Delta s}{2},\phi} + Cr\Delta t} + \frac{\sin\left(\theta - \frac{\Delta s}{2}\right)}{\eta^*_{\theta-\frac{\Delta s}{2},\phi} + Cr\Delta t}\right], \\ A_{i,\mathrm{left|right}} = & -\sin^{-1}\theta M\Delta t \Delta s^{-2}(\eta^*_{\theta,\phi\mp\frac{\Delta s}{2}} + Cr\Delta t)^{-1}, \\ A_{i,\mathrm{up}} = \quad & -\sin\left(\theta - \frac{\Delta s}{2}\right) M\Delta t \Delta s^{-2}(\eta^*_{\theta-\frac{\Delta s}{2},\phi} + Cr\Delta t)^{-1}, \\ A_{i,\mathrm{down}} = \quad & -\sin^{-1}\theta M\Delta t \Delta s^{-2}(\eta^*_{\theta,\phi+\frac{\Delta s}{2}} + Cr\Delta t)^{-1},\end{aligned}$$

where θ and ϕ denote the cell center of the respective mid cell. The right hand side b is given by

$$b_{i,j} = \sin\theta\left(\frac{1}{\Delta t} - \nabla \cdot \frac{\eta^*_{i,j}\mathbf{u}^*_{i,j} + Cr\Delta t(\mathbf{u}_{\mathrm{air}})_{i,j} + \Delta t \eta^*_{i,j}\mathbf{g}_{i,j}}{\eta^*_{i,j} + Cr\Delta t}\right).$$