Functional Thin Films on Surfaces

Omri Azencot[1] Orestis Vantzos[2] Max Wardetzky[3] Martin Rumpf[2] Mirela Ben-Chen[1]

[1]Technion – Israel Institute of Technology [2]University of Bonn [3]University of Göttingen

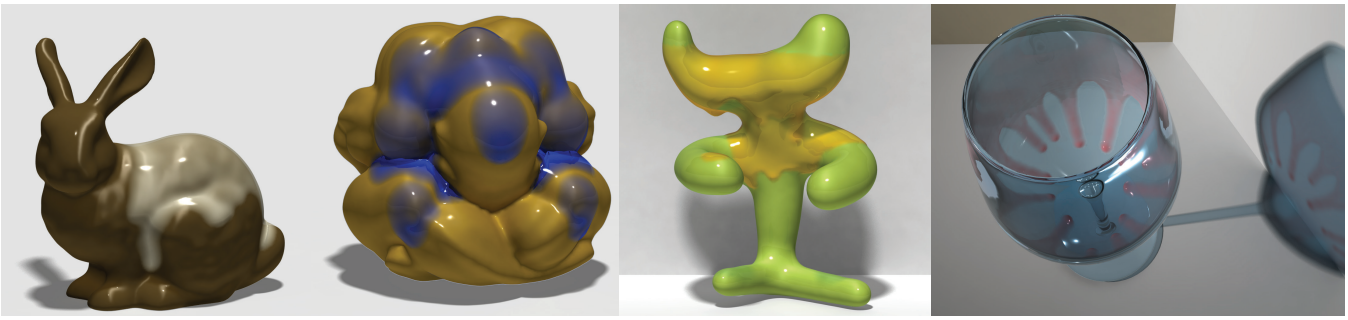

Figure 1: *Various effects of liquid thin film flow achievable with our method.*

Abstract

The motion of a thin viscous film of fluid on a curved surface exhibits many intricate visual phenomena, which are challenging to simulate using existing techniques. A possible alternative is to use a reduced model, involving only the temporal evolution of the mass density of the film on the surface. However, in this model, the motion is governed by a fourth-order nonlinear PDE, which involves geometric quantities such as the curvature of the underlying surface, and is therefore difficult to discretize. Inspired by a recent variational formulation for this problem on smooth surfaces, we present a corresponding model for triangle meshes. We provide a discretization for the curvature and advection operators which leads to an efficient and stable numerical scheme, requires a single sparse linear solve per time step, and exactly preserves the total volume of the fluid. We validate our method by qualitatively comparing to known results from the literature, and demonstrate various intricate effects achievable by our method, such as droplet formation, evaporation, droplets interaction and viscous fingering.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

Keywords: thin films, free surface flows, flows on curved surfaces

1 Introduction

The intricate motion of a viscous thin film subject to external forces, such as gravity, inspires research in physics, mathematics and computer science, among other scientific disciplines. In many scenarios the domain on which the fluid resides is curved rather than flat. The tear film on the cornea of the eye [Braun et al. 2012], the dynamics of lava flows [Griffiths 2000] and the formation of ice on the aerofoil of an aircraft [Myers and Charpin 2004], are all examples related to the evolution of thin films on curved geometries. The goal of this paper is to suggest a method for simulating thin films on surfaces, which is based on *gradient flow* evolution and the *operator view* of the flow induced by tangent vector fields.

Generally, the *Navier–Stokes equations* coupled with appropriate boundary conditions are assumed to give a good approximation of the film's dynamics. However, for the flows we are interested in, these equations are considered difficult to solve numerically, especially on curved domains. Moreover, in the case of thin films we can assume an extremely small height-to-length ratio which leads to a substantial simplification through the *lubrication approximation* [Reynolds 1886]. Namely, under the assumptions of the lubrication model, the evolution of the film's mass density is governed by a fourth-order nonlinear partial differential equation (PDE).

A natural approach to simulate thin films within this reduced model would then be to discretize the resulting PDE (e.g., [Roy et al. 2002]). Choosing such a strategy, however, one will be faced with two main challenges. First, one will need to derive a suitable set of discrete differential operators acting on discrete curved domains (e.g., triangle meshes). Then, the second task will be to construct a proper numerical time integration scheme. While any attempt to discretize general PDEs will encounter these obstacles, in the particular case of thin films, the restriction on the time step size (see e.g., [Greer et al. 2006]) makes the usage of explicit schemes impractical. Although it is possible to use implicit schemes instead, such schemes do not guarantee in general the preservation of the underlying structure. For example, conserved quantities in the continuous setting (such as the total volume of the thin film) may become non-conserved in a discrete framework. Due to the above obstacles, direct discretization of the PDE is usually considered less attractive.

An alternative point of view is to leverage the *gradient flow* structure which is known to exist for thin film equations (see e.g., [Giacomelli and Otto 2003; Rumpf and Vantzos 2013]). In this model, the motion of the film is determined by the minimizer of a certain cost function, which is defined over the manifold of all possible densities of the film with prescribed volume. Intuitively, the cost function is minimized when the resistance of the fluid to flow due to dissipation induced by friction balances the additional forces (e.g., surface tension and gravity) that act on the film. One of the advantages of this approach is that every gradient flow has a natural time discretization which leads to a variational problem. In practice, it allows for significantly larger time steps compared to explicit nu-

merical schemes. Furthermore, by construction, the associated energy is guaranteed to decrease at each step.

However, we still need to address the issues of modeling the underlying mass transport and the conservation of fluid volume. A reasonable choice within the gradient flow model is to minimize the cost function under an additional constraint given by the transport equation. Intuitively, the transport equation describes how the mass density is affected by the motion of the fluid through the corresponding velocity field. Recently, [Azencot et al. 2013] suggested a coordinate-free approach for solving the transport equation on triangulated surfaces by representing tangent vector fields as *linear operators* on scalar functions. Their method is advantageous since it avoids the complicated integration of the fluid's motion, while ensuring the preservation of the integral of the transported quantity.

In this work, we argue that the gradient flow model combined with the operator view of tangent vector fields leads to a robust and highly efficient simulation tool. Specifically, we consider the thin film model of [Rumpf and Vantzos 2013] in the presence of a *precursor layer* (i.e., the film resides on top of a very thin layer defined over the whole domain) and in the geometric setting of triangulated surfaces. Under the assumption that we are given an approximate normal field, we present formulations of discrete curvature operators which are tailored for our model. In addition, we employ insights from [Azencot et al. 2013] to advect the mass function of the thin film in a manner which causes very little numerical dissipation, and is guaranteed to conserve exactly the total volume of the fluid. The resulting method boils down to a *linear* solve of a sparse system per time step. We demonstrate the effects of curvature, gravity (see e.g., Fig. 2) and material parameters on the flow, and qualitatively compare our results to previous numerical simulations. Finally, we present various effects (e.g., droplet formation and interaction) which are achievable within our framework.

1.1 Related Work

As the behaviour of viscous thin films on surfaces has not, to the best of our knowledge, been previously simulated in the graphics community, we focus our attention on Eulerian methods from the computational fluid dynamics community, and to work on similar phenomena which appeared in the computer graphics literature.

The evolution of thin films over arbitrary domains has been an active area of research in CFD for many decades. We refer the interested reader to the seminal review by [Oron et al. 1997] and to the more recent review by [Craster and Matar 2009]. These reviews present a continuous model for thin films, based on *lubrication theory*, which defines a reduced model for the 3D Navier–Stokes equations given the assumption of a small thickness of the film.

One approach to thin film simulation is to directly discretize the governing PDE as was shown for planar (see e.g., [Zhornitskaya and Bertozzi 1999; Grün and Rumpf 2000]) and curved (see e.g., [Roy et al. 2002]) domains. In general, this point of view leads to several challenges, of which the restriction on the time step size for explicit schemes is perhaps the most problematic. Namely, the application of a CFL-type condition leads to the requirement that the time step τ is on the order of $(\delta x)^4$, where δx is the minimal edge length. To overcome this constraint, [Greer et al. 2006] employed convexity splitting for their time integration scheme (within a level-set framework). Nevertheless, their scheme does not guarantee conservation of the fluid's volume, and has additional restrictions due to the level-set formulation.

An alternative discretization for thin films can be derived from the gradient flow model, for which a natural variational time integrator exists. In general, variational integrators are known to conserve the underlying structure, e.g., the variational scheme in [Mullen et al. 2009] preserves a notion of discrete momentum. For the case of thin films over curved domains (see e.g., [Vantzos 2014; Rumpf and Vantzos 2013]), the gradient flow approach leads to an attractive numerical scheme. In the latter work, which is closest to our approach, the authors used Discrete Exterior Calculus (DEC) [Hirani 2003] for the spatial discretization, representing the flux field with discrete 1-forms. Our approach differs from their work as we use a velocity based formulation, leverage [Azencot et al. 2013] for the advection, and suggest discrete curvature operators. These changes allow us to generate stable simulations on arbitrary triangle meshes which are common in graphics. A detailed comparison with [Rumpf and Vantzos 2013] is given in Sections 3 and 4.

We conclude with some representative related work from the graphics community literature. Free surface flows for highly viscous fluids were suggested in [Carlson et al. 2002], where effects such as melting wax are demonstrated. While one could consider adding a surface as a solid boundary and using a similar approach for simulating viscous films, it would be quite difficult to achieve the intricate effects we show without using a very dense grid resolution. More recently, various methods were proposed for modeling thin features in free surface flows by explicitly tracking the free surface mesh [Wojtan et al. 2011; Zhang et al. 2012], by using thickened triangle meshes [Batty et al. 2012], tetrahedral elements [Clausen et al. 2013], or simplicial complexes [Zhu et al. 2014], to mention just a few. Such approaches, however, require careful manipulation of the connectivity and topology of the free surface geometry, which are avoidable when simulating films on surfaces, as the free surface can be represented as a scalar function.

Finally, some approaches simulate water related phenomena. Wang et al. [2005] model the contact angle with the surface, representing the free surface with a level-set based distance field. While various effects are achievable with this approach, the method requires a high-resolution grid which leads to a time-consuming system requiring a few days of computation per simulation. On the other hand, using a height field based method as in [Wang et al. 2007] considerably reduces computational complexity, however, the instabilities and effects we demonstrate below were not shown there.

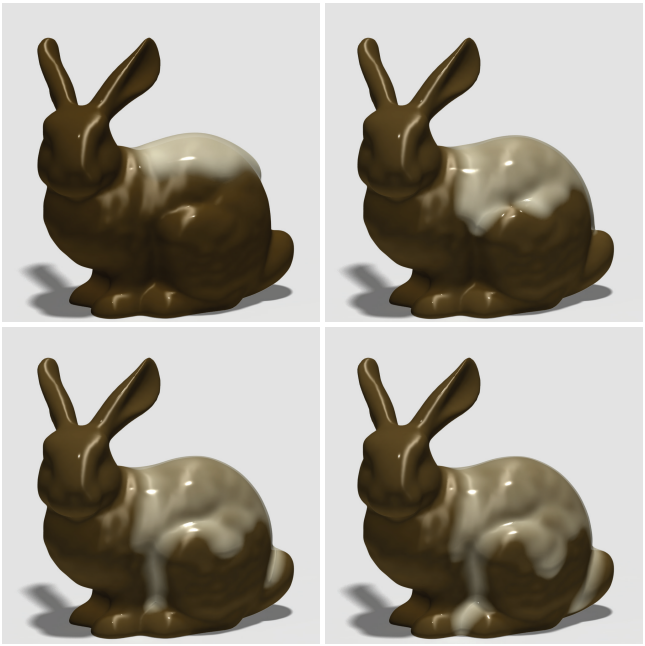

Figure 2: *Vanilla sauce on a chocolate bunny. The physical parameters are* $\mathfrak{b} = 20, \epsilon = 0.1, \beta = 0$.

1.2 Contributions

Our main contributions can be described as follows:

- A discrete model for thin film evolution on *general* triangle meshes.
- An efficient and robust scheme, which exactly preserves the total fluid's volume.
- Simulation of various intricate effects, such as fingering, evaporation and droplet formation and interaction.

2 Dynamics of thin films

We investigate the evolution of a layer of an incompressible viscous fluid flowing with velocity v on top of a curved surface Γ, under the influence of surface tension and, potentially, gravity. The liquid layer is attached to the surface at the liquid-solid interface, i.e., no-slip boundary condition (we extend this later), whereas the liquid-air surface is evolving freely. A typical scenario is illustrated in Figure 3 showing the notation for various related quantities.

Navier–Stokes equations. A common approach for modeling the evolution of thin liquid films is to consider the *Navier–Stokes equations*. These equations describe the fluid's velocity in the liquid phase (the *bulk*), the surface tension on the liquid-air interface (i.e., the *free surface*), and a suitable boundary condition for the velocity in the liquid-solid interface (i.e., on the solid surface). Formally, the fluid velocity v and the pressure p satisfy the equations:

$$\begin{aligned} \partial_t v + (v \cdot \nabla) v - \mu \Delta v + \nabla p &= 0 \text{ in the bulk} \\ \operatorname{div} v &= 0 \text{ in the bulk} \\ v &= 0 \text{ on the surface} \\ \sigma n - \gamma \mathcal{H} n &= 0 \text{ at the free surface} \end{aligned} \tag{1}$$

where $\sigma = -p\,\mathrm{id} + \mu(\nabla v + \nabla v^T)$ is the stress tensor, μ and γ are the viscosity and the capillary constants. Furthermore, the free surface itself evolves according to the kinematic condition $\partial_t \chi = v$.

Unfortunately, a straightforward discretization of these equations is challenging. In particular, to achieve the type of effects we show below, the main obstacle is due to the prohibitively small time steps which are imposed by such a method. Moreover, the spatial discretization is also challenging since Eulerian methods will require dense sampling of the domain, whereas Lagrangian techniques will involve complex tracking of the free surface. Therefore, direct discretization of equations (1) is not practical for graphics applications for this type of problems.

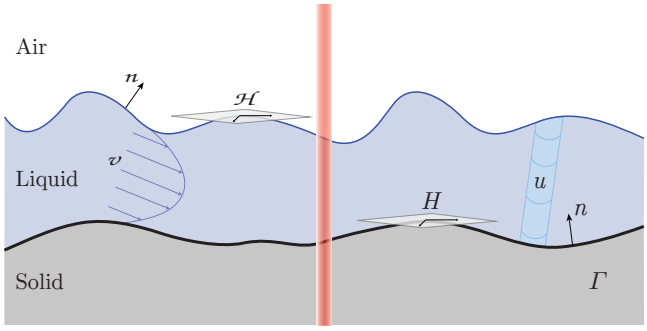

Figure 3: *A typical scenario is illustrated for the full 3D Navier–Stokes (left) compared to the reduced lubrication model (right). Notice that under the lubrication assumptions the involved quantities are computed directly on Γ, e.g., u is a scalar function.*

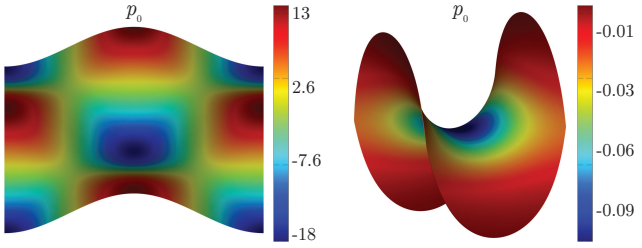

Figure 4: *By visualizing the pressure we can identify regions where the fluid is likely to accumulate. For example, for an initially uniform layer of fluid, the initial pressure p_0 indicates that fluid is expected to concentrate at the respective centers, where the pressure is lowest. See Fig. 5 for the temporal evolution of the flow.*

Lubrication approximation. Since we are interested in *thin* films, a reduction in dimensionality can be achieved by using the *lubrication approximation* model (see e.g., [Oron et al. 1997]). In this model, the dynamics of the film are governed by the evolution of a function (i.e., a scalar quantity) defined on the surface Γ.

Given a characteristic scaling of height and length, the key assumption to consider is a small height to length ratio, i.e., $\epsilon = \frac{\text{height}}{\text{length}} \ll 1$. Then, one takes into account an asymptotic expansion of the Navier–Stokes equations with respect to ϵ, where the resulting thin film equations are composed of the leading order terms. Taking this path, a derivation of a lubrication model without gravity for the *mass density* u on curved domains yields equations of the form (see [Roy et al. 2002] and [Rumpf and Vantzos 2013]):

$$\partial_t u = \operatorname{div}_\Gamma \left(M(u) \nabla_\Gamma p \right) \tag{2a}$$

$$M(u) = \frac{1}{3} u^3 \,\mathrm{id} + \frac{\epsilon}{6} u^4 (H\,\mathrm{id} - S) \tag{2b}$$

$$p = -H - \epsilon T u - \epsilon \Delta_\Gamma u \tag{2c}$$

where $M(u)$ is the mobility tensor (to be discussed later) and p can be considered as a pressure-like quantity on the surface, i.e., the fluid moves away from areas of high p. H and K are the mean and Gaussian curvatures, $T = H^2 - 2K$, and S is the shape operator.

Notice that inertia effects are neglected in this model, i.e., the Reynolds number is assumed to be small, $\mathrm{Re} \ll 1$, as expected (by simple scaling arguments) for a thin enough film. Moreover, we assume that the mass density u is a proper function. As u is closely related to the fluid's height h, that is $u = h - \frac{\epsilon}{2} H h^2$, the consequence of the former constraint is that the free surface is assumed to be representable as a height function over Γ, and hence, e.g., contact angles higher than $\pi/2$ and wave-like structures cannot be modeled with equations (2).

In addition to providing a reduced model for the Navier–Stokes equations, the thin films equations are also instrumental for analyzing the behavior of the flow. As mentioned above, the fluid flows towards low pressure areas thus visualizing p allows to evaluate the underlying dynamics of the film. Moreover, a qualitative study of the expected flow can be done by estimating the different scales of the various components in p. For instance, the dominating term in Eq. (2c) is the mean curvature and hence the dynamics on curved domains are expected to be completely different when compared to the flat case (where $H = 0$). Indeed, we demonstrate this and other effects in the following example.

In Figure 4 we show the color coding of the pressure computed for an initial uniform deposition of liquid on a bumpy plane (left) and on the Scherk surface (right). These figures suggest that the fluid is most likely to accumulate at the center of the respective

surfaces, where the pressure is low. Indeed, we show in Figure 5 (top) the color coding of the evolution of the mass density u on the bumpy plane, starting from a uniform layer of fluid. In this case, since the dominating term is H (top, left), the film flows towards the maximal mean curvature, at the center of the basin. Similarly, for a minimal surface, namely when $H = 0$, the terms that govern the dynamics are the Gaussian curvature and the Laplacian of u. In Figure 5 (bottom), we show frames of the flow on the Scherk minimal surface, starting again from a uniform layer of fluid. Here, the initial Laplacian of u is 0 thus the minimal Gaussian curvature (bottom, left) drives the fluid towards the center of the surface.

Unfortunately, the simulation of thin film flow based on a PDE of the form (2) suffers from serious drawbacks. First, explicit discretization of equation (2) requires very strong time step restrictions, and stable (semi-)implicit discretizations allowing for large time steps, are unknown. Second, qualitative properties, such as volume preservation and energy decay, are difficult to ensure. Finally, on general triangulated surfaces it is unclear how to discretize the geometric quantities in a physically consistent way.

These issues motivate a different approach—instead of directly discretizing the PDE, it is possible to model the evolution from the *variational* perspective of gradient flows, as was first suggested in [Rumpf and Vantzos 2013]. To introduce the concepts to the graphics community, and to keep the paper self contained, in the next section we first briefly describe the gradient flow model of thin films, and then discuss our modifications.

3 Gradient flow model

Background. The key insight behind the variational approach is that the quantity p can be viewed as the negative (Frechet) derivative of the *free energy functional* $\mathcal{E}^\epsilon(u) = \int_\Gamma \left\{ -Hu - \frac{\epsilon}{2} T u^2 + \frac{\epsilon}{2} |\nabla_\Gamma u|^2 \right\} \mathrm{d}x$ so that the PDE (2) is of the *gradient flow* form $\partial_t u = -G(\frac{\delta \mathcal{E}^\epsilon(u)}{\delta u})$. The evolution of u then can be understood as a "steepest" descent for the free energy $\mathcal{E}^\epsilon$, at a rate regulated by the *mobility* $M(u)$ via the function $G(\phi) = \mathrm{div}_\Gamma\,(M(u)\nabla_\Gamma \phi)$. The previous statement can be made precise by introducing the flux $f = -M(u)\nabla_\Gamma p$, so that the PDE can be written in the form of a flow equation as

$$\partial_t u = -\,\mathrm{div}_\Gamma\, f. \tag{3}$$

Then the gradient flow is equivalent to the statement that the free energy decays as $\frac{d}{dt}\mathcal{E}^\epsilon(u) = -\mathcal{D}^\epsilon_u(f,f) \le 0$, where the bilinear form $\mathcal{D}^\epsilon_u(f,f) = \int_\Gamma f \cdot M(u)^{-1} f \,\mathrm{d}x$ is known as the (viscous) *dissipation*. This in turn is equivalent to the variational requirement that the density variation $\partial_t u$ and the flux f minimize (at each time t) the so-called Rayleigh functional $\frac{1}{2}\mathcal{D}^\epsilon_u(f,f) + \frac{\delta \mathcal{E}^\epsilon(u)}{\delta u}(\partial_t u)$ under the transport constraint (3).

Intuitively, the energy is an approximation of the area of the free surface, which should be minimized due to surface tension, and the dissipation is the "price to pay" for the total shear stress due to the flow inside the film. Hence, among all the possible flows which preserve the mass of the fluid, we look for the one which optimally minimizes the area of the free surface and the stress inside the film.

Finally, following the idea of *natural time discretization* of gradient flows [Otto 2001] and *minimizing movements* [Giorgi and Ambrosio 2006], we integrate in time to arrive at a variational approximation of $u^{k+1} = u(t^k + \tau)$ given $u^k = u(t^k)$:

$$u^{k+1} = \underset{u = \mathcal{F}_\tau(u^k, f)}{\mathrm{argmin}} \left\{ \frac{1}{2\tau}\mathcal{D}^\epsilon_u(f,f) + \mathcal{E}^\epsilon(u) \right\} \tag{4}$$

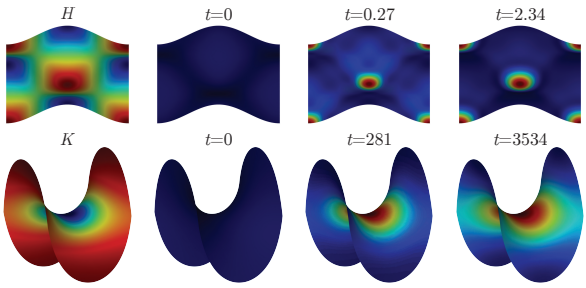

Figure 5: *(top) The motion of the film primarily depends on the mean curvature thus the fluid concentrates in the center basin, $u_0 = 0.1, \epsilon = 0.1$. (bottom) For minimal surfaces (i.e., when $H = 0$) the film is mostly influenced by the Gaussian curvature as shown for the Scherk's surface, $u_0 = 0.1, \epsilon = 1$.*

where $\mathcal{F}_\tau(u^k, f)$ denotes a suitable (approximate) solution at $t^k + \tau$ of the initial value transport problem (3) with $u(t^k) = u^k$. The constrained minimization problem (4) is equivalent to discretizing the original PDE (2) in time; instead of the PDE then, one can describe (and discretize) the thin film flow through the three components of the gradient flow: the free energy $\mathcal{E}^\epsilon$, the dissipation $\mathcal{D}$ and the flow equation (3) (or in the time-discrete setting the flow operator $\mathcal{F}_\tau$).

This approach, as shown in [Rumpf and Vantzos 2013], addresses some of the shortcomings of PDE-based solvers pointed out previously. Speficially, discrete qualitative properties are straightforward to preserve: the energy decay is built into the time discretization (4), as will be shown later, and it is also easier to set up discrete mass conservation for the flow equation than for the full PDE (2). In addition, because of the explicit control on the energy decay, the variational scheme is very stable, allowing for large time steps.

In [Rumpf and Vantzos 2013], suitable energy and dissipation functionals are derived for gravity- and surface tension-driven thin film flow on a *smooth* curved surface. The variational time discretization (4) is coupled then with a spatial discretization based on Discrete Exterior Calculus, resulting in a fully discrete scheme on triangulated surfaces that has the positive properties mentioned above. Unfortunately, directly applying that scheme for graphics purposes on *general* triangle meshes is challenging. Next we describe our modifications to the aforementioned scheme which lead to stable simulations on general triangle meshes, and compare our results.

Transport constraint. The transport constraint is difficult to discretize while ensuring discrete mass preservation. One way to achieve this (as was done in [Rumpf and Vantzos 2013]), is to work with a flux-based formulation, that lends itself naturally to a finite-volume approach such as Discrete Exterior Calculus. However, in the presence of obtuse triangles, i.e., triangles with angles larger than $\pi/2$, negative entries can arise in the diagonal matrices that the scheme uses to define inner products between discrete k-forms. This can lead to non-convexity and eventually to instability and/or non-convergence of the variational scheme. Notice that for general meshes, eliminating these obtuse triangles is highly non-trivial.

Recently, a new discretization for the transport equation was suggested in [Azencot et al. 2013], which uses a velocity-based representation, and does not suffer from the aforementioned problem. We therefore switch to a velocity-based representation, resulting in the transport equation $\partial_t u + \mathrm{div}_\Gamma(uv) = 0$, discretized in a similar spirit to [Azencot et al. 2013] (as described in Section 4).

We compared our scheme to the method of [Rumpf and Vantzos 2013] on the bunny model which has obtuse triangles. Specifically, we computed the difference in energy and the minimal u in the first iteration for different time step sizes. In Figure 6 (left) we show

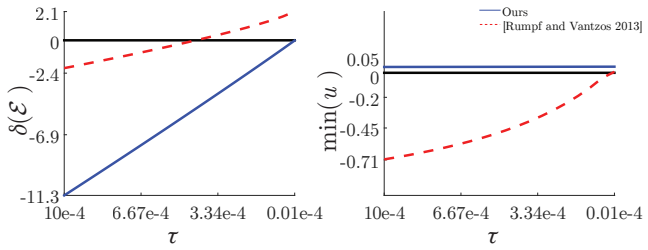

Figure 6: *Comparison with [Rumpf and Vantzos 2013]. (left) The present scheme consistently decreases the energy, i.e., $\delta(\mathcal{E}) \leq 0$ (even for meshes with obtuse triangles), whereas the other method is non-converging—it increases the energy for decreasing time step size, contrary to the desired behavior. (right) Here the present method preserves the initial value of u while the other method yields negative values.*

that our method is consistently decreasing the energy, whereas the method of [Rumpf and Vantzos 2013] actually increases the energy for small time steps. In addition, we show in Figure 6 (right) that their method yields negative values for u even for very small time steps, whereas ours preserves the initial value of the precursor layer.

Approximate normal fields. As we have previously seen in Figure 5, the film dynamics are heavily dependent on the curvature operators, H, K and S. In their work [Rumpf and Vantzos 2013] presented one dimensional applications and simulations on two dimensional surfaces where the curvatures are easy to compute analytically (such as surfaces of revolution and graphs). One could, of course, extend their method to triangulated surfaces by choosing a set of discrete curvature operators from the many available in the literature (see e.g., [Gatzke and Grimm 2006]). We chose instead to go back to fluid mechanics and look for a definition of the energy and dissipation functionals that could be applied on continuous but non-smooth surfaces, such as a triangulated mesh.

Our main observation is that if Γ is equipped with a continuous vector field n that is *approximately*[1] normal, one can follow similar derivations as in [Rumpf and Vantzos 2013], and arrive at energy and dissipation functionals given by (up to an $\mathrm{O}(\epsilon^2)$ error):

$$\mathcal{E}^\epsilon(u) = \int_\Gamma (\mathfrak{b}z - H)u + \frac{\epsilon}{2}(\mathfrak{b}\cos\theta - T)u^2 + \frac{\epsilon}{2}|\nabla_\Gamma u|^2 \,\mathrm{d}a \quad (5)$$

and

$$\mathcal{D}^\epsilon_u(v,v) = \int_\Gamma v \cdot M(u)^{-1} v \,\mathrm{d}a \quad (6)$$

$$M(u) = \left(\beta + \frac{u}{3}\right)\mathrm{id} + \epsilon\frac{u^2}{12}\left(7H\,\mathrm{id} - 3S - 5\bar{S}\right) \quad (7)$$

respectively. In (5) we included the gravity terms that involve the Bond number $\mathfrak{b}$, which measures the relative strength of gravity vs. surface tension, the altitude z, and the angle θ of the surface normal with the vertical direction. Moreover, we incorporated in (7) a constant β which allows for various slip conditions.

Note that (unlike in [Rumpf and Vantzos 2013]) the curvature quantities in these equations are now given in terms of the approximate normal field n, as follows. The discrete shape operator S is defined, per triangle, as the tangential gradient of n (albeit symmetrized and projected to the tangent plane):

$$S := -\frac{1}{2}P(\nabla_\Gamma n + (\nabla_\Gamma n)^T)P\,, \quad (8)$$

[1] On a triangular mesh with face normals ν and average length edge δx, we require that $n \cdot \nu = \mathrm{O}(\delta x)$ and that $\nabla_\Gamma n$ be tangential and symmetric up to order $\mathrm{O}(\delta x)$.

where $P = \mathrm{id} - \nu\nu^T$ is the projection onto the tangent space. From this shape operator we extract a discrete mean curvature $H = \mathrm{Tr}(S)$ and a discrete Gaussian curvature $K = \frac{1}{2}\left(\mathrm{Tr}(S)^2 - \mathrm{Tr}(S^2)\right)$. Surprisingly, these definitions are exactly analogous to the continuous case. Finally, the *rotated shape operator* $\bar{S} = -[\nu]_\times S[\nu]_\times$ is defined via the skew-symmetric matrix $[\nu]_\times$, where $[\nu]_\times \cdot x = \nu \times x$ for any vector x.

In the case of triangular meshes, the thin film can be visualized as a union of prism-like volumes, each one attached to a triangle, with sides parallel to the vector field n (see Figure 7). The discrete shape operator S captures the shape of each prism, with H and K describing whether the sides of the prism converge or diverge away from its base. The rotated shape operator $\bar{S}$ on the other hand is related to the effect that crossing from one prism to the other has on the fluid flow.

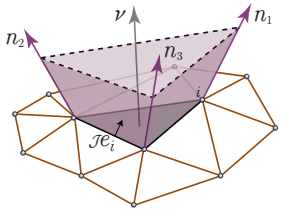

Figure 7: *Prismatic layer of fluid over a triangle.*

Together with the transport constraint

$$\partial_t u + \mathrm{div}_\Gamma(uv) = 0\,, \quad (9)$$

the energy functional (5) and the dissipation functional (6) completely describe the gradient flow model on which our scheme is based. In the next section we address the remaining required ingredients, namely the discretizations of the function u, the vector field v, the differential operators and the transport constraint.

4 Spatial discretization

The main challenge here is to define a stable discretization of the transport equation (9) such that various properties (e.g., energy decay and mass preservation) will hold on general triangle meshes. While many of the operators we use are standard in geometry processing, we highlight the properties these operators should possess such that the resulting optimization scheme would indeed be stable.

Notation. We consider a triangle mesh and denote by $\mathcal{V}$ its vertex set and by $\mathcal{F}$ its face set. We use bold faced symbols to denote the spatial discrete analogues of continuous quantities (e.g., $\boldsymbol{u}$ is the discrete mass density). When required, we use the subscripts $\mathcal{V}$ and $\mathcal{F}$ to denote quantities on the vertices and the faces, respectively. The bracket $[\cdot]$ operator is used to convert vectors in $\mathbb{R}^{|\mathcal{V}|}$ and $\mathbb{R}^{|\mathcal{F}|}$ to block diagonal matrices in $\mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ and $\mathbb{R}^{3|\mathcal{F}|\times 3|\mathcal{F}|}$ respectively (replicating each entry 3 times for the latter).

Functions, vector fields and inner products. We use a typical setup, i.e., piecewise-linear functions and piecewise-constant vector fields, with corresponding inner products. Specifically, we represent real-valued functions as scalars on the vertices of the mesh, i.e., $\boldsymbol{u} \in \mathbb{R}^{|\mathcal{V}|}$, and extend them to the whole mesh using piecewise linear hat basis functions. Similarly, vector fields are treated as piecewise-constant on the faces of the mesh, i.e., $\boldsymbol{v} \in \mathbb{R}^{3|\mathcal{F}|}$.

For defining discrete inner products we require vertex and face areas, denoted by $\boldsymbol{A}_\mathcal{V} \in \mathbb{R}^{|\mathcal{V}|}$ and $\boldsymbol{A}_\mathcal{F} \in \mathbb{R}^{|\mathcal{F}|}$, respectively. For the vertex area we use $1/3$ of the total area of its adjacent triangles, and we define an interpolating matrix $\boldsymbol{I}^\mathcal{F}_\mathcal{V} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{F}|}$ which interpolates quantities from faces to the vertices, i.e., $\boldsymbol{I}^\mathcal{F}_\mathcal{V}(i,j) = \frac{\boldsymbol{A}_\mathcal{F}(j)}{3\boldsymbol{A}_\mathcal{V}(i)}$, iff vertex i belongs to face j and 0 otherwise. This choice implies that $\boldsymbol{A}_\mathcal{F} = (\boldsymbol{I}^\mathcal{F}_\mathcal{V})^T\boldsymbol{A}_\mathcal{V}$, which will be important for consistency

later. Now, discrete inner products are defined by:

$$\int_\Gamma \boldsymbol{u_1}\boldsymbol{u_2}\mathrm{d}a = \boldsymbol{u}_1^T \boldsymbol{G}_\mathcal{V} \boldsymbol{u_2}, \quad \int_\Gamma \langle \boldsymbol{v_1}, \boldsymbol{v_2} \rangle \mathrm{d}a = \boldsymbol{v}_1^T \boldsymbol{G}_\mathcal{F} \boldsymbol{v_2},$$

where $\boldsymbol{G}_\mathcal{V} = [\boldsymbol{A}_\mathcal{V}] \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ and $\boldsymbol{G}_\mathcal{F} = [\boldsymbol{A}_\mathcal{F}] \in \mathbb{R}^{3|\mathcal{F}|\times 3|\mathcal{F}|}$ denote the diagonal mass matrix of the vertices and the faces.

Differential Operators. Equations (5) and (9) require discrete gradient and divergence operators. In the smooth case, these operators fulfill integration by parts, namely on a surface without boundary we have: $\int_\Gamma \langle v, \nabla_\Gamma u \rangle \, \mathrm{d}a + \int_\Gamma u \cdot \mathrm{div}_\Gamma \, v \, \mathrm{d}a = 0$. In order to maintain discrete preservation of mass (see appendix A), we need the operators $\mathbf{grad}_\Gamma \in \mathbb{R}^{3|\mathcal{F}|\times|\mathcal{V}|}$ and $\mathbf{div}_\Gamma \in \mathbb{R}^{|\mathcal{V}|\times 3|\mathcal{F}|}$ to fulfill this discretely, namely:

$$\boldsymbol{v}^T \boldsymbol{G}_\mathcal{F} (\mathbf{grad}_\Gamma \, \boldsymbol{u}) + (\mathbf{div}_\Gamma \, \boldsymbol{v})^T \boldsymbol{G}_\mathcal{V} \boldsymbol{u} = 0,$$

for arbitrary $\boldsymbol{v}$ and $\boldsymbol{u}$. Interestingly, the standard operators (e.g., as defined in [Botsch et al. 2010, Chapter 3]) fulfill this property.

Approximate normal field, curvature and gravity. As described in the previous section, all of the required curvature quantities can be computed once a suitable approximate normal field is given. In practice, we use the area-weighted averages of triangle normals [Botsch et al. 2010, pg. 42] as vertex normals. By applying the discrete gradient operator defined previously, the tangential gradient of the discrete normal field per face j is:

$$(\nabla_\Gamma \boldsymbol{n})_j = \frac{1}{2\boldsymbol{A}_\mathcal{F}(j)} \left(\sum_{i=1}^{3} \boldsymbol{n}_{j_i} (\mathcal{J}\boldsymbol{e}_{j_i})^T \right)$$

where the sum runs over the three vertex normals $\boldsymbol{n}_{j_i}$ of the face and $\mathcal{J}\boldsymbol{e}_{j_i}$ is the rotated (by $\pi/2$) edge opposite to vertex i in the triangle j (see Figure 7). The gravity quantities can be computed as follows: $\boldsymbol{z}$ is the vertical coordinate function and $\mathbf{cos}\,\boldsymbol{\theta}$ is the vertical component function of $\boldsymbol{n}$.

Mobility. The discrete mobility $\boldsymbol{M}(\boldsymbol{u})$ is a $3|\mathcal{F}| \times 3|\mathcal{F}|$ *diagonal* matrix, where for each face the associated quantities can be computed using Eq. (7), the curvature operators, and the interpolated mass density $\boldsymbol{u}_\mathcal{F}$ on the faces ($\boldsymbol{u}$ is defined on vertices).

Transport operator. In the continuous case, equation (9) guarantees that the integral of $\partial_t u$ vanishes on a closed surface (since the divergence of any vector field integrates to 0). However, once we discretize u and v then $\mathbf{div}_\Gamma(\boldsymbol{uv})$ is no longer well defined using our discrete operators, since $\boldsymbol{uv}$ is not a piecewise constant vector field. To avoid this issue, we first apply the product rule to (9) and reformulate the constraint as $\partial_t u = -(v \cdot \nabla_\Gamma u + u \, \mathrm{div}_\Gamma \, v)$. We then follow [Azencot et al. 2013] and define a directional derivative $\boldsymbol{D}(\boldsymbol{v})$ such that $\mathbf{1}_\mathcal{V}^T \boldsymbol{G}_\mathcal{V} \left(\boldsymbol{D}(\boldsymbol{v}) + [\mathbf{div}_\Gamma \, \boldsymbol{v}] \right) \boldsymbol{u} = 0$ for any $\boldsymbol{u}$ and $\boldsymbol{v}$ (see the Appendix for the proof). Specifically, the directional derivative is given as $\boldsymbol{D}(\boldsymbol{v}) \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ by $\boldsymbol{D}(\boldsymbol{v}) = \boldsymbol{I}_\mathcal{V}^\mathcal{F} [\boldsymbol{v}]_\bullet^T \mathbf{grad}_\Gamma$, where $[\cdot]_\bullet \in \mathbb{R}^{3|\mathcal{F}|\times|\mathcal{F}|}$ converts vector fields to block diagonal matrices.

The main advantage of this point of view is that in the discrete case the transport equation turns into a system of ODEs of the form $\partial_t \boldsymbol{u} + \boldsymbol{A}\boldsymbol{u} = 0$, for a constant matrix $\boldsymbol{A}$, which can be solved using a matrix exponential [Hochbruck and Ostermann 2010]. Thus, for a velocity $\boldsymbol{v}$ constant in time, the discrete transport equation can be solved in the time interval $[t^k, t^k + \tau]$ to yield the solution

$$\boldsymbol{u} = \exp\left(-\tau \boldsymbol{D}(\boldsymbol{v}) - \tau[\mathbf{div}_\Gamma \, \boldsymbol{v}]\right) \boldsymbol{u}^k \tag{10}$$

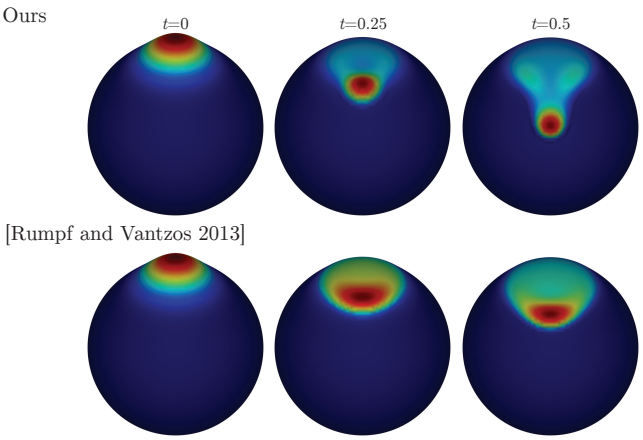

Figure 8: *Starting from the same initial conditions and physical parameters, our transport scheme (top) achieves a better resolved finger compared to the result (bottom) generated with the more diffusive scheme suggested in [Rumpf and Vantzos 2013].*

at $t = t^k + \tau$, where τ is the time step. In the case of evaporation, we have an additional term $-\tau[\boldsymbol{u}^k + c_e]^{-2}$ in the exponential.

The above transport mechanism is more appropriate to the flows we are interested in than the one suggested by [Rumpf and Vantzos 2013]. In particular, droplet formation and fingering instabilities are transport-dominated effects. Thus, a natural requirement from a transport mechanism is to exhibit minimum diffusion, allowing to capture better resolved fingers on relatively coarse meshes as we demonstrate. We show in Figure 8 that starting from the same initial conditions, our scheme is qualitatively less diffusive compared to the method of [Rumpf and Vantzos 2013].

5 Fully discrete model

Given the above discrete operators and quantities, we can write the fully-discrete optimization problem for computing $\boldsymbol{u}, \boldsymbol{v}$ given $\boldsymbol{u^k}$:

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{v}} \left\{ \frac{1}{2\tau} \mathcal{D}^\epsilon_{\boldsymbol{u}^k}(\boldsymbol{v}, \boldsymbol{v}) + \mathcal{E}^\epsilon(\boldsymbol{u}) \right\}, \\ &\text{subject to } \boldsymbol{u} = \exp\left(-\tau \boldsymbol{D}(\boldsymbol{v}) - \tau[\mathbf{div}_\Gamma \, \boldsymbol{v}]\right) \boldsymbol{u}^k. \end{aligned} \tag{11}$$

Then, the fully-discrete energy and dissipation are given by:

$$\mathcal{E}^\epsilon(\boldsymbol{u}) = \boldsymbol{a}^T \boldsymbol{G}_\mathcal{V} \boldsymbol{u} + \frac{\epsilon}{2} \boldsymbol{u}^T (\boldsymbol{G}_\mathcal{V} \boldsymbol{B} + \boldsymbol{L}) \boldsymbol{u},$$
$$\mathcal{D}^\epsilon_{\boldsymbol{u}^k}(\boldsymbol{v}, \boldsymbol{v}) = \boldsymbol{v}^T \boldsymbol{G}_\mathcal{F} \boldsymbol{M}(\boldsymbol{u}^k)^{-1} \boldsymbol{v},$$

where $\boldsymbol{a} = \mathfrak{b}\boldsymbol{z} - \boldsymbol{H}$, $\boldsymbol{B} = \mathfrak{b}\mathbf{cos}\,\boldsymbol{\theta} - \boldsymbol{H^2} + 2\boldsymbol{K}$, and the stiffness matrix $\boldsymbol{L} = -\boldsymbol{G}_\mathcal{V} \, \mathbf{div}_\Gamma \, \mathbf{grad}_\Gamma$.

5.1 Properties

Discrete energy. *The discrete energy $\mathcal{E}^\epsilon(\boldsymbol{u}^k)$ is non increasing.*

Proof: Noticing that $\boldsymbol{u} = \boldsymbol{u}^k$ and $\boldsymbol{v} = 0$ is an admissible pair for the minimization problem (11) since they satisfy the constraint, we have immediately that:

$$\begin{aligned} \frac{1}{2\tau} \mathcal{D}^\epsilon_{\boldsymbol{u}^k}(\boldsymbol{v}^{k+1}, \boldsymbol{v}^{k+1}) + \mathcal{E}^\epsilon(\boldsymbol{u}^{k+1}) &\leq \frac{1}{2\tau} \mathcal{D}^\epsilon_{\boldsymbol{u}^k}(0,0) + \mathcal{E}^\epsilon(\boldsymbol{u}^k) \\ &\Rightarrow \mathcal{E}^\epsilon(\boldsymbol{u}^{k+1}) \leq \mathcal{E}^\epsilon(\boldsymbol{u}^k) \end{aligned}$$

since $\mathcal{D}^\epsilon_{\boldsymbol{u}^k}(\boldsymbol{v}^{k+1}, \boldsymbol{v}^{k+1}) \geq 0$ and $\mathcal{D}^\epsilon_{\boldsymbol{u}^k}(0,0) = 0$.

Intuitively, since $\mathcal{D}$ is non-negative, if the fluid moved and "paid" with dissipation, then it found a smaller energy solution (otherwise it will have remained at the previous state, with the same energy).

Discrete mass. *The total discrete mass* $m(\boldsymbol{u}) = \int_\Gamma \boldsymbol{u}\ \mathrm{d}a = \mathbf{1}_\mathcal{V}^T \boldsymbol{G}_\mathcal{V} \boldsymbol{u}$ *is exactly preserved.*

Proof: The transport equation (10) can be written as $\boldsymbol{u} = \exp(-\tau \boldsymbol{A})\boldsymbol{u}^k$, where $\boldsymbol{A} = \boldsymbol{D}(\boldsymbol{v}) + [\mathbf{div}_\Gamma \boldsymbol{v}]$. In the appendix we show $\mathbf{1}_\mathcal{V}^T \boldsymbol{G}_\mathcal{V} \boldsymbol{A} = 0$ for any velocity $\boldsymbol{v}$. Hence, we have $m(\boldsymbol{u}^k) - m(\boldsymbol{u}) = \mathbf{1}_\mathcal{V}^T \boldsymbol{G}_\mathcal{V} \{\mathrm{id} - \exp(-\tau \boldsymbol{A})\} \boldsymbol{u}^k = \mathbf{1}_\mathcal{V}^T \boldsymbol{G}_\mathcal{V} \left\{\tau \boldsymbol{A} - \frac{\tau^2}{2}\boldsymbol{A}^2 + \ldots\right\} \boldsymbol{u}^k = 0.$

5.2 Optimization

To solve the discrete variational model (11) we use the first order approximation $\exp(-\tau \boldsymbol{A}) \approx \mathrm{id} - \tau \boldsymbol{A}$ of the matrix exponential, so that the linear equation:

$$\boldsymbol{u} = \boldsymbol{u}^k - \tau(\boldsymbol{D}(\boldsymbol{v}) + [\mathbf{div}_\Gamma \boldsymbol{v}])\boldsymbol{u}^k \tag{12}$$

replaces the non-linear constraint (10). Hence, at every time step we solve *a quadratic problem with a linear constraint*, which is convex for a small enough τ (see §5.3 "Dynamic Time-stepping"). As we will show next, this can be done very efficiently, by solving a single linear system for $\boldsymbol{u}$. Note that it is straightforward to check that the results of §5.1 hold for the linearized constraint as well, hence we gain efficiency yet do not lose stability.

The linear system. Using the method of Lagrange multipliers we obtain the *first order necessary conditions*:

$$\begin{aligned} \boldsymbol{G}_\mathcal{F} \boldsymbol{M}(\boldsymbol{u}^k)^{-1} \boldsymbol{v} - \left(\overline{\boldsymbol{D}}(\boldsymbol{u}^k) + [\boldsymbol{u}^k]\,\mathbf{div}_\Gamma\right)^T \boldsymbol{G}_\mathcal{V} \boldsymbol{p} &= 0 \\ \boldsymbol{G}_\mathcal{V}\left(\boldsymbol{a} + \epsilon \boldsymbol{B}\boldsymbol{u}\right) + \epsilon \boldsymbol{L}\boldsymbol{u} - \boldsymbol{G}_\mathcal{V}\boldsymbol{p} &= 0 \\ \boldsymbol{G}_\mathcal{V}\left(\boldsymbol{u} - \boldsymbol{u}^k + \tau(\boldsymbol{D}(\boldsymbol{v}) + [\mathbf{div}_\Gamma \boldsymbol{v}])\boldsymbol{u}^k\right) &= 0, \end{aligned} \tag{13}$$

where $\boldsymbol{p}$ is the dual variable.

A key ingredient to deriving (13) is the *dual operator* $\overline{\boldsymbol{D}}(\boldsymbol{u})$, defined such that $\boldsymbol{D}(\boldsymbol{v})\boldsymbol{u} = \overline{\boldsymbol{D}}(\boldsymbol{u})\boldsymbol{v}$, as it allows us to take derivatives with respect to $\boldsymbol{v}$. This operator is: $\overline{\boldsymbol{D}}(\boldsymbol{u}) = \boldsymbol{I}_\mathcal{V}^\mathcal{F}[\mathbf{grad}_\Gamma \boldsymbol{u}]_\bullet^T$. Similarly, it holds that $([\boldsymbol{u}]\,\mathbf{div}_\Gamma)\,\boldsymbol{v} = [\mathbf{div}_\Gamma \boldsymbol{v}]\boldsymbol{u}$.

Finally, eliminating $\boldsymbol{v}$ and $\boldsymbol{p}$, we arrive at the following *reduced linear system* for $\boldsymbol{u}$:

$$\left(\mathrm{id} + \tau\epsilon\, \boldsymbol{R}(\boldsymbol{u}^k, \boldsymbol{u}_e^k)(\boldsymbol{G}_\mathcal{V}\boldsymbol{B} + \boldsymbol{L})\right)\boldsymbol{u} = \boldsymbol{u}_e^k - \tau \boldsymbol{R}(\boldsymbol{u}^k, \boldsymbol{u}_e^k)\boldsymbol{G}_\mathcal{V}\boldsymbol{a} \tag{14}$$

where $\boldsymbol{R}(\boldsymbol{u}^k, \boldsymbol{u}_e^k) = \boldsymbol{F}(\boldsymbol{u}_e^k)\boldsymbol{M}(\boldsymbol{u}^k)\boldsymbol{G}_\mathcal{F}^{-1}\boldsymbol{F}(\boldsymbol{u}_e^k)^T$ and $\boldsymbol{F}(\boldsymbol{u}_e^k) = \overline{\boldsymbol{D}}(\boldsymbol{u}_e^k) + [\boldsymbol{u}_e^k]\,\mathbf{div}_\Gamma$ and $\boldsymbol{u}_e^k = \exp(-\tau[\boldsymbol{u}^k + c_e]^{-2})\boldsymbol{u}^k$ if evaporation is included and $\boldsymbol{u}_e^k = \boldsymbol{u}^k$ otherwise.

Thus, we obtain a fully discrete scheme where given an initial mass density $\boldsymbol{u}_0$, we evolve it in time using the above update rule.

We implemented our method in MATLAB using standard linear solvers for Eq. (14). In all our experiments, the method was very stable allowing for large time steps (on the scale of $\mathrm{O}(\epsilon + \delta x)$, which is excellent for 4th order problems) depending on the initial conditions and the underlying mesh. The experiments were performed on an Intel(R) Xeon(R) processor with 32 GB RAM, and we show in Table 1 the statistics for the different simulations.

Figure	$\lvert\mathcal{V}\rvert$	Avg. per step	#steps	Total time
Fig. 2, Bunny*	38306	0.484	1999	967.8
Fig. 5, Bumpy plane	40401	0.683	4996	3410.4
Fig. 5, Scherk surface	40401	0.627	1997	1252.4
Fig. 10, Rounded cube*	19728	0.142	4991	709.5
Fig. 11, Sphere	40962	1.645	300	493.5
Fig. 13, Moomoo*	16710	0.080	1981	158.4
Fig. 14, Torus	40000	1.079	456	491.8
Fig. 16, Moai	89126	3.106	314	975.3
Fig. 15, Rain	10242	0.198	18001	3570.1
Fig. 17, Pensatore	27732	0.818	991	810.3
Fig. 18, Wine glass*	38976	0.708	496	351.1

Table 1: *Timing statistics (in seconds). Asterisk denotes simulations where an iterative solver was used, whereas for the rest, we used a direct non-iterative solver.*

5.3 Limitations

Dynamic Time-Stepping. Given that the stiffness matrix $\boldsymbol{L}$ is positive semi-definite, the system (14) is invertible as long as $\tau_1\, \epsilon \|\boldsymbol{R}(\boldsymbol{u}^k)\|_2 \|\boldsymbol{G}_\mathcal{F}\|_2 \underline{\boldsymbol{B}} \leq 1$, where $\underline{\boldsymbol{B}}$ is the absolute taken on the minimum value of $\boldsymbol{B}$ and it is a measure of how strongly negative the quantity $\mathfrak{b}\cos\theta - T$ is on the surface. Moreover, we employ a CFL-type condition depending on the maximum velocity of the film $\underline{\boldsymbol{v}}$, and grid size, i.e., we require that $\tau_2 \underline{\boldsymbol{v}} \leq \delta x$. Finally, we take the time step to be $\tau = \min\{\tau_1, \tau_2\}$.

Positivity Preservation. Unfortunately, even if we start from a strictly positive $\boldsymbol{u}_0$, the evolution of the film $\boldsymbol{u}^k$ is not guaranteed to stay positive [Rumpf and Vantzos 2013]. Aside from being non-physical, in the case of negative values, droplets might rupture. In practice, all of our simulations remain positive, excluding the evaporation example. Nevertheless, the evaporation term has a stabilizing effect, indeed, negative mass concentrations are also evaporated. Intuitively, positivity is difficult to maintain due to the jump in pressure along the triple line (the interface where air, solid and liquid meet). Moreover, the so-called capillary ridge is formed, due to the competition between surface tension and other forcing effects, e.g., gravity, see Figure 9 and 10. Thus, right where the film is at its thinnest, the resulting velocity is high, implying instability along the direction of motion. We leave further investigation of the issue of positivity preservation for future work.

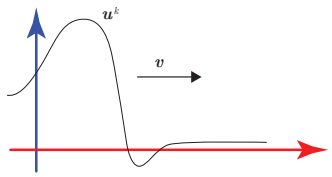

Figure 9: *Capillarity ridge with high velocity and undershooting.*

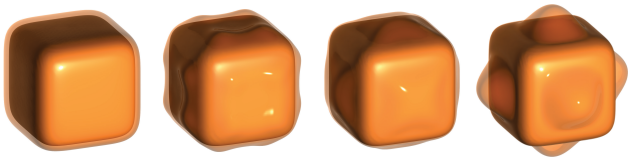

Figure 10: *In the absence of gravity, the fluid departs areas where the mean curvature is strongly negative and capillary ridges form. Later, surface tension balances the fluid on top of every face, cf. [Roy et al. 2002]* ($u_0 = 0.1, \mathfrak{b} = 0, \epsilon = 0.1, \beta = 0$).

Meshes with creases. In general, the model we developed in Section 3 has a strong dependency on the consistency of the vertex normals. In practice, general meshes might have creases, or small dihedral angles, which will cause H to be arbitrarily nega-

tively large and non-smooth. This can have a detrimental effect on the simulation, as the fluid will be drawn towards these singular locations. There are two possible remedies for this situation: we can either refine the mesh (possibly non-uniformly), however that would require additional pre-processing before one can apply our scheme to an arbitrary model. Alternatively, we can add a regularizer to the energy so that it is easier to control the simulation. We opted for the second option, as it makes our method easier to use, and can allow the artist some freedom to control the simulation in a non-physical way. Hence, for meshes with creases (see e.g., Fig. 17), we multiply the stiffness matrix $\boldsymbol{L}$ defined in Section 5 by a constant $1 \leq \boldsymbol{r} \leq 100$. This effectively adds some numerical diffusion, allowing for more smooth solutions. Note that discrete conservation of mass is not affected by this modification.

Detachment of fluid. As the fluid is "tied" to the surface, droplets cannot detach when they become too large. In these cases, the droplets grow narrower and taller until equilibrium is reached and the approximate lubrication solution is stable, although the full 3D flow is not. Note, that in this case one could potentially switch to a full 3D simulation, which will allow the droplet to separate from the surface. This is an interesting direction for future research.

6 Experimental results

Parameter exploration. We begin by exploring the effect of various parameter choices on the simulation of the thin film. For this example, we choose a sphere as a simple geometric model with limited curvature effects on the flow. The basic experiment includes placing a concentration of fluid at the top of the sphere, with slightly perturbed initial conditions to avoid perfect symmetry. Due to gravity the fluid flows downward, and the initial perturbations give rise to fingering instabilities, (see [Takagi and Huppert 2010] for an experimental demonstration of fingering on a sphere). The result for the parameters $\epsilon = 0.05, \mathfrak{b} = 50, \beta = 0$ is shown in Figure 11 (f), demonstrating the emergence of a secondary finger in the center (see also Fig. 18, showing multiple fingers in a wine glass).

We refer to this setup as the *reference configuration*, and now modify in every column of the figure a single parameter to isolate its effect on the simulation, for which we show a snapshot at time $t = 10$. Left: varying $\mathfrak{b}$ changes the speed with which the film flows downward, without strongly affecting the shape of the fingers. Specifically, for a lower $\mathfrak{b}$ value (a), the secondary finger does not emerge yet, whereas for a higher $\mathfrak{b}$ value (b) it is more pronounced than in the reference configuration. Middle: changing ϵ affects the surface tension component, and therefore the shape of the fingers. Reducing ϵ yields thinner fingers (c), whereas increasing it (d) makes more viscous thick fingers, and eliminates the secondary finger. Right: increasing β considerably speeds up the fluid (e), allowing it to flow more freely in all directions (as opposed to increasing $\mathfrak{b}$ which causes faster flow in the direction of gravity).

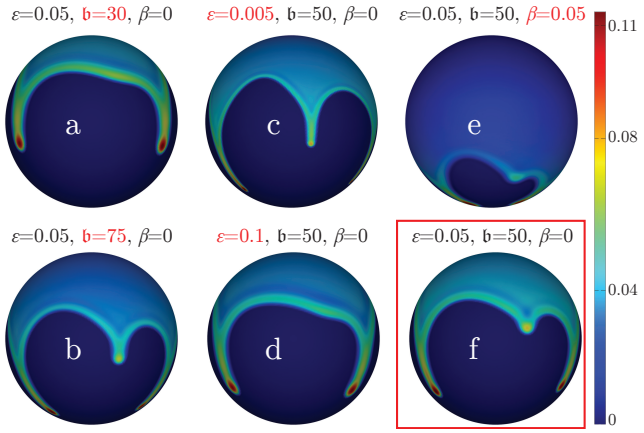

Figure 11: *Fingering behavior for varying parameters, at $t = 10$. In every column, one parameter is modified from the reference configuration (f). See the text for details.*

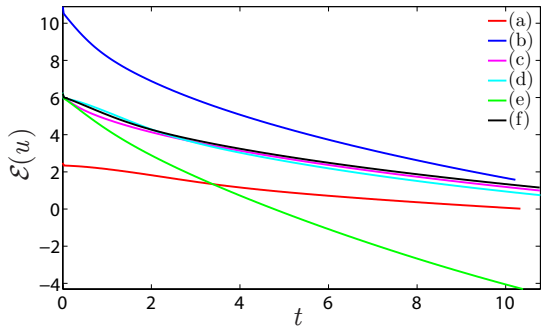

Figure 12: *The energy $\mathcal{E}(\boldsymbol{u})$ for the simulations in Figure 11.*

Energy reduction. The numerical scheme we use is guaranteed by construction to reduce the energy $\mathcal{E}(\boldsymbol{u})$ at every time step. Figure 12 shows the energy decay in time, for the different simulations in Figure 11. We observe that the slip parameter β affects the speed with which the energy is reduced, the gravity parameter $\mathfrak{b}$ also affects the initial value of the energy, and the parameter ϵ has a minor impact on the energy, as it is dominated by the leading order term.

Thin films interaction. Figure 13 demonstrates the flow and interaction of thin films on the moomoo model. The higher bulk of fluid accumulates beneath the horns of the model, followed by a faster motion when it comes in contact with the lower bulk of fluid (see also Figure 16). Then, the motion is mostly determined by the two main fingers flowing on the sides of the model. In Fig. 15 we show the interaction of many droplets viewed from four sides of the unit sphere. We repeatedly pour new droplets at the top of the sphere at a fixed rate and drain the liquid from the bottom.

Droplet formation. A thin film concentrating *beneath* a flat surface develops an instability called *droplet formation* (cf. [Sharma and Khanna 1998]). In Figure 14, we start with a uniform layer of

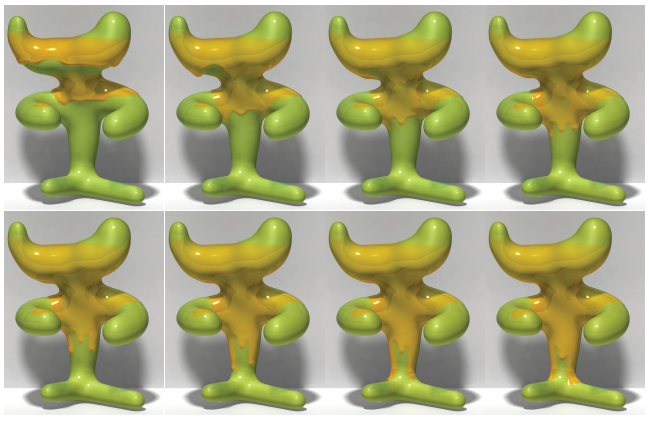

Figure 13: *Flow on the moomoo model ($\mathfrak{b} = 20, \epsilon = 0.1, \beta = 0$). Note how the upper and lower films interact: the larger mass density of the upper film causes it to catch up with the lower front leading to the formation of quickly propagating fingers.*

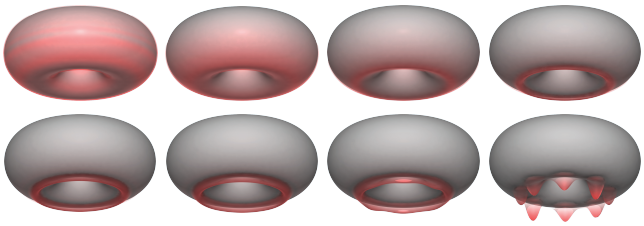

Figure 14: *Starting from a perturbed uniform layer of fluid, the fluid flows downwards, accumulates and finally forms droplets.*

fluid on the torus with small perturbations, and allow it to drop beneath the torus due to gravity. As the fluid accumulates around the circular set of lowest points, droplets form.

Evaporation. Figure 16 shows how evaporation ($c_e = 0.01$) and the precursor layer affect the motion of the film. We deposit precursor layers of different heights on the two halves of the Moai model and place a similar bulk of fluid near the eyes. Due to the initially thicker precursor layer, even though it evaporates quickly, the film on the left part of the model flows to a greater distance compared to the film on the right. Eventually, all the film evaporates.

7 Conclusion

We presented a novel method for simulating viscous thin film flow on triangulated meshes. Our approach is based on a variational time discretization and is therefore stable and allows for large time steps. Furthermore, we guarantee by construction that the discrete total mass is preserved and that the discrete energy is non-increasing. The algorithm is based on a single sparse linear solve per iteration, and is therefore very efficient. We demonstrated various intricate film motions, such as viscous fingering and droplet interaction.

There are many potential extensions to our model. First, we can add non-linear energy terms to model additional effects and to avoid negative solutions. Furthermore, it might be possible to extend the model to handle effects due to surface tension gradient. Finally, our discretization of the mass transport constraint might be potentially useful in additional applications.

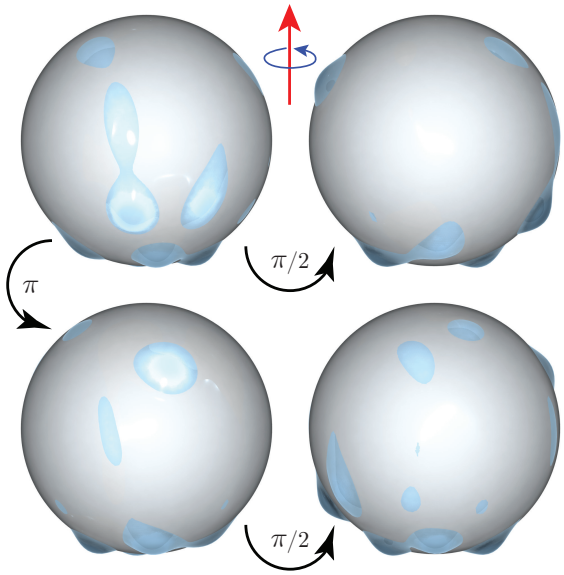

Figure 15: *Rain of droplets lead to their interesting interaction over the sphere (see the video for the full simulation). The sphere is shown from its four sides, where the axis of rotation is shown above.*

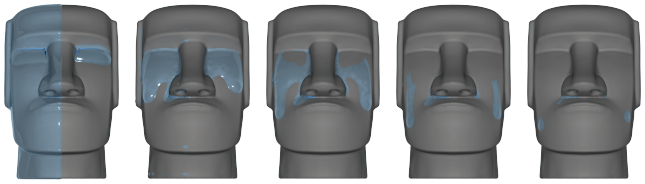

Figure 16: *Evaporation effect on the evolution of the film.*

A Discrete conservation of mass

To prove that the discrete mass $m(\boldsymbol{u}) = \mathbf{1}_{\mathcal{V}}^T \boldsymbol{G}_{\mathcal{V}} \boldsymbol{u}$, with $\mathbf{1}_{\mathcal{V}}$ a vector of ones of length $|\mathcal{V}|$, is strictly preserved, we recall the first order necessary conditions (13). For the linearized transport equation $\boldsymbol{u} = \boldsymbol{u}^k - \tau \left(\boldsymbol{D}(\boldsymbol{v}) + [\mathbf{div}_\Gamma \boldsymbol{v}] \right) \boldsymbol{u}^k$, we multiply the third equation by $\mathbf{1}_{\mathcal{V}}^T$ and, using the duality of $\boldsymbol{D}$ and $\overline{\boldsymbol{D}}$, the discrete product rule, and the fact that $\boldsymbol{A}_{\mathcal{F}} = (\boldsymbol{I}_{\mathcal{V}}^{\mathcal{F}})^T \boldsymbol{A}_{\mathcal{V}}$ and $[x]_\bullet y_{\mathcal{F}} = [y_{\mathcal{F}}]x$ for any vector field x and face function $y_{\mathcal{F}}$, we obtain:

$$
\begin{aligned}
& m(\boldsymbol{u}^{k+1}) - m(\boldsymbol{u}^k) = -\tau\, \mathbf{1}_{\mathcal{V}}^T \boldsymbol{G}_{\mathcal{V}} \left(\boldsymbol{D}(\boldsymbol{v}) + [\mathbf{div}_\Gamma \boldsymbol{v}] \right) \boldsymbol{u}^k \\
&= -\tau\, \mathbf{1}_{\mathcal{V}}^T \boldsymbol{G}_{\mathcal{V}} \left(\overline{\boldsymbol{D}}(\boldsymbol{u}^k) + [\boldsymbol{u}^k] \, \mathbf{div}_\Gamma \right) \boldsymbol{v} \\
&= -\tau\, \boldsymbol{v}^T \left(\overline{\boldsymbol{D}}(\boldsymbol{u}^k) + [\boldsymbol{u}^k] \, \mathbf{div}_\Gamma \right)^T \boldsymbol{G}_{\mathcal{V}} \mathbf{1}_{\mathcal{V}} \\
&= -\tau\, \boldsymbol{v}^T \left([\mathbf{grad}_\Gamma \boldsymbol{u}^k]_\bullet (\boldsymbol{I}_{\mathcal{V}}^{\mathcal{F}})^T \boldsymbol{G}_{\mathcal{V}} \mathbf{1}_{\mathcal{V}} + \mathbf{div}_\Gamma{}^T [\boldsymbol{u}^k] \boldsymbol{G}_{\mathcal{V}} \mathbf{1}_{\mathcal{V}} \right) \\
&= -\tau\, \boldsymbol{v}^T \left([(\boldsymbol{I}_{\mathcal{V}}^{\mathcal{F}})^T \boldsymbol{G}_{\mathcal{V}} \mathbf{1}_{\mathcal{V}}] \, \mathbf{grad}_\Gamma \boldsymbol{u}^k + \mathbf{div}_\Gamma{}^T \boldsymbol{G}_{\mathcal{V}} [\boldsymbol{u}^k] \mathbf{1}_{\mathcal{V}} \right) \\
&= -\tau\, \boldsymbol{v}^T \left([\boldsymbol{A}_{\mathcal{F}}] \, \mathbf{grad}_\Gamma \boldsymbol{u}^k - \boldsymbol{G}_{\mathcal{F}} \, \mathbf{grad}_\Gamma [\boldsymbol{u}^k] \mathbf{1}_{\mathcal{V}} \right) \\
&= -\tau\, \boldsymbol{v}^T \boldsymbol{G}_{\mathcal{F}} \left(\mathbf{grad}_\Gamma \boldsymbol{u}^k - \mathbf{grad}_\Gamma \boldsymbol{u}^k \right) = 0 \,.
\end{aligned}
$$

Acknowledgements

This work was supported by the Volkswagen Foundation, Hannover, Germany, ISF grant 699/12, Marie Curie CIG 303511, Jacobs–Qualcomm fellowship, the Hausdorff Center for Mathematics, the state of Lower-Saxony, BMBF project MuSiKa, and the Niedersachsen-Israel grant "Spectral Methods in Geometry Processing: Theory and Applications". Meshes are provided by the Stanford Computer Graphics Laboratory and the AIM@SHAPE Repository. The authors thank Peter Shipkov for his support with the rendering and his SOuP plugin and the anonymous reviewers.

References

AZENCOT, O., BEN-CHEN, M., CHAZAL, F., AND OVSJANIKOV, M. 2013. An operator approach to tangent vector field processing. In *CGF*, vol. 32, Wiley Online Library, 73–82.

BATTY, C., URIBE, A., AUDOLY, B., AND GRINSPUN, E. 2012. Discrete viscous sheets. *ACM Trans. Graph. 31*, 4, 113.

BOTSCH, M., KOBBELT, L., PAULY, M., ALLIEZ, P., AND LÉVY, B. 2010. *Polygon mesh processing*. CRC press.

BRAUN, R., USHA, R., MCFADDEN, G., DRISCOLL, T., COOK, L., AND KING-SMITH, P. 2012. Thin film dynamics on a prolate spheroid with application to the cornea. *J. of Eng. Math. 73*, 1, 121–138.

CARLSON, M., MUCHA, P. J., VAN HORN III, R. B., AND TURK, G. 2002. Melting and flowing. In *Proc. of SCA 2002*, 167–174.

CLAUSEN, P., WICKE, M., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2013. Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Trans. Graph. 32*, 2, 17.

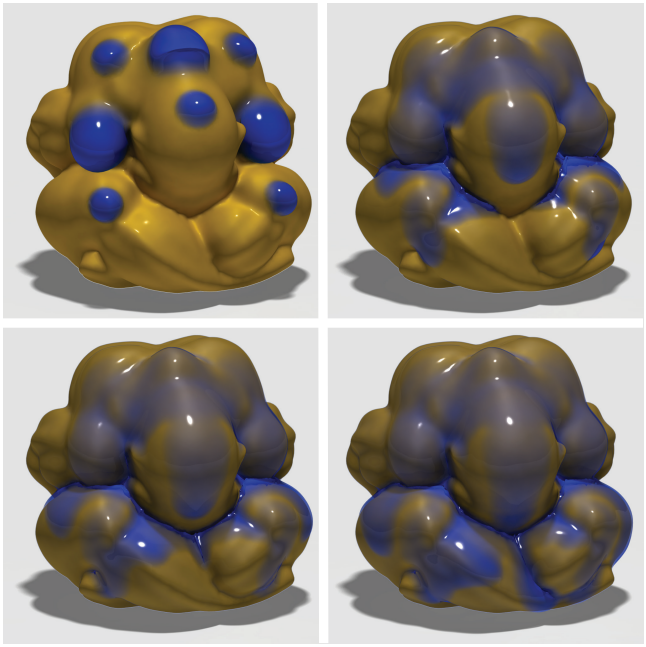

Figure 17: *Thin film flow on a geometrically complicated model. Note how starting from a few blobs of fluid, the film naturally follows the creases of the object, merges and splits accordingly.*

CRASTER, R., AND MATAR, O. 2009. Dynamics and stability of thin liquid films. *Reviews of modern physics 81*, 3, 1131.

GATZKE, T. D., AND GRIMM, C. M. 2006. Estimating curvature on triangular meshes. *Int. J. of shape modeling 12*, 01, 1–28.

GIACOMELLI, L., AND OTTO, F. 2003. Rigorous lubrication approximation. *Interfaces and Free boundaries 5*, 4, 483–530.

GIORGI, E., AND AMBROSIO, L. 2006. New problems on minimizing movements. In *Ennio De Giorgi selected papers*. Springer, 699–714.

GREER, J. B., BERTOZZI, A. L., AND SAPIRO, G. 2006. Fourth order partial differential equations on general geometries. *J. of Comp. Physics 216*, 1, 216–246.

GRIFFITHS, R. 2000. The dynamics of lava flows. *Ann. Rev. of Fluid Mechanics 32*, 1, 477–518.

GRÜN, G., AND RUMPF, M. 2000. Nonnegativity preserving convergent schemes for the thin film equation. *Numerische Mathematik 87*, 1, 113–152.

HIRANI, A. N. 2003. *Discrete exterior calculus*. PhD thesis, California Institute of Technology.

HOCHBRUCK, M., AND OSTERMANN, A. 2010. Exponential integrators. *Acta Numerica 19*, 209–286.

MULLEN, P., CRANE, K., PAVLOV, D., TONG, Y., AND DESBRUN, M. 2009. Energy-preserving integrators for fluid animation. *ACM Trans. Graph. 28*, 3, 38.

MYERS, T. G., AND CHARPIN, J. P. 2004. A mathematical model for atmospheric ice accretion and water flow on a cold surface. *Int. J. of Heat and Mass Transfer 47*, 25, 5483–5500.

ORON, A., DAVIS, S. H., AND BANKOFF, S. G. 1997. Long-scale evolution of thin liquid films. *Rev. of modern physics 69*, 3, 931.

OTTO, F. 2001. The geometry of dissipative evolution equations: the porous medium equation. *Comm. in Partial Differential Equations 26*, 1-2, 101–174.

REYNOLDS, O. 1886. On the theory of lubrication and its application to Mr. Beauchamp tower's experiments, including an experimental determination of the viscosity of olive oil. *Proc. of the Royal Society of London 177*, 157–234.

ROY, R. V., ROBERTS, A. J., AND SIMPSON, M. 2002. A lubrication model of coating flows over a curved substrate in space. *J. of Fluid Mechanics 454*, 235–261.

RUMPF, M., AND VANTZOS, O. 2013. Numerical gradient flow discretization of viscous thin films on curved geometries. *Math. Models and Methods in Applied Sciences 23*, 05, 917–947.

SHARMA, A., AND KHANNA, R. 1998. Pattern formation in unstable thin liquid films. *Phys. Rev. Lett. 81* (Oct), 3463–3466.

TAKAGI, D., AND HUPPERT, H. E. 2010. Flow and instability of thin films on a cylinder and sphere. *J. of Fluid Mech. 647*, 221.

VANTZOS, O. 2014. *Thin viscous films on curved geometries*. PhD thesis, Universitäts-und Landesbibliothek Bonn.

WANG, H., MUCHA, P. J., AND TURK, G. 2005. Water drops on surfaces. *ACM Trans. Graph. 24*, 3, 921–929.

WANG, H., MILLER, G., AND TURK, G. 2007. Solving general shallow wave equations on surfaces. In *Proc. of SCA 2007*, 229–238.

WOJTAN, C., MÜLLER-FISCHER, M., AND BROCHU, T. 2011. Liquid simulation with mesh-based surface tracking. In *ACM SIGGRAPH 2011 Courses*, ACM.

ZHANG, Y., WANG, H., WANG, S., TONG, Y., AND ZHOU, K. 2012. A deformable surface model for real-time water drop animation. *Vis. and Comp. Graph., IEEE Trans. on 18*, 8.

ZHORNITSKAYA, L., AND BERTOZZI, A. L. 1999. Positivity-preserving numerical schemes for lubrication-type equations. *SIAM J. on Numerical Analysis 37*, 2, 523–555.

ZHU, B., QUIGLEY, E., CONG, M., SOLOMON, J., AND FEDKIW, R. 2014. Codimensional surface tension flow on simplicial complexes. *ACM Trans. Graph. 33*, 4, 111.

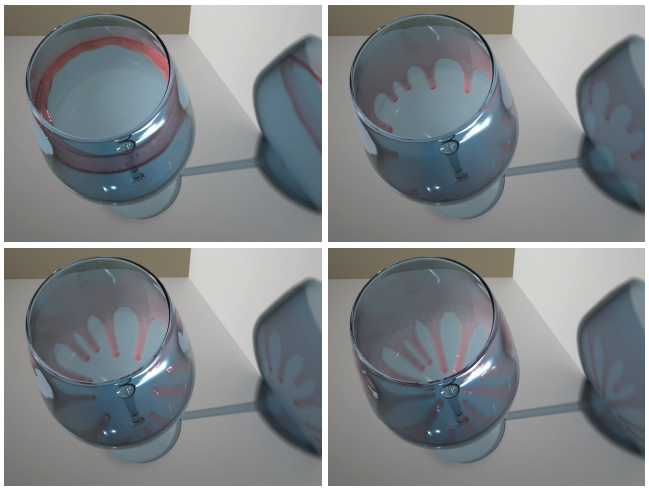

Figure 18: *Multiple fingers on the inside of a glass of wine* ($\mathfrak{b} = 500, \epsilon = 0.0001, \beta = 0$).