# Lawrence Berkeley National Laboratory
## LBL Publications

**Title**
Multiscale modelling of evolving foams

**Permalink**
https://escholarship.org/uc/item/1wp0b13k
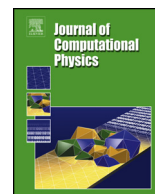
**Authors**
Saye, RI
Sethian, JA

**Publication Date**
2016-06-01

**DOI**
10.1016/j.jcp.2016.02.077

Peer reviewed

# Multiscale modelling of evolving foams

R.I. Saye, J.A. Sethian *

*Lawrence Berkeley National Laboratory and Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720, United States*

A B S T R A C T

We present a set of multi-scale interlinked algorithms to model the dynamics of evolving foams. These algorithms couple the key effects of macroscopic bubble rearrangement, thin film drainage, and membrane rupture. For each of the mechanisms, we construct consistent and accurate algorithms, and couple them together to work across the wide range of space and time scales that occur in foam dynamics. These algorithms include second order finite difference projection methods for computing incompressible fluid flow on the macroscale, second order finite element methods to solve thin film drainage equations in the lamellae and Plateau borders, multiphase Voronoi Implicit Interface Methods to track interconnected membrane boundaries and capture topological changes, and Lagrangian particle methods for conservative liquid redistribution during rearrangement and rupture. We derive a full set of numerical approximations that are coupled via interface jump conditions and flux boundary conditions, and show convergence for the individual mechanisms. We demonstrate our approach by computing a variety of foam dynamics, including coupled evolution of three-dimensional bubble clusters attached to an anchored membrane and collapse of a foam cluster.

## 1. Introduction

### 1.1. Physics of foams

In this paper, we introduce, implement, and test a suite of numerical algorithms for computing solutions to a mathematical model of multi-scale foam dynamics.

Foams have a wide variety of applications in industry and materials design. For example, liquid foams, characterized by fluid-filled membranes separating gaseous regions, include soapy detergents and substances to separate out hydrophobic molecules; solid foams, formed by solidifying liquid foams, include lightweight materials such as metallic and plastic foams. Understanding the dynamics of foam evolution is a key step in controlling the structure and properties of foam-like materials. Deriving models to quantitatively predict foam evolution is challenging since the underlying physics takes place over vastly different time and space scales.

As a model of foam, consider common soap bubbles. A single isolated bubble consists of a thin membrane of fluid, known as the lamella, separating the inside gas from the outside. In a cluster of such bubbles, multiple lamellae meet at junctions known as Plateau borders, forming a network of interconnected thin-film membranes and borders. The dynamics of foam is intricate [57], and depends on a complex interaction between microscale fluid flow inside the lamellae and Plateau borders,

---

* Corresponding author.
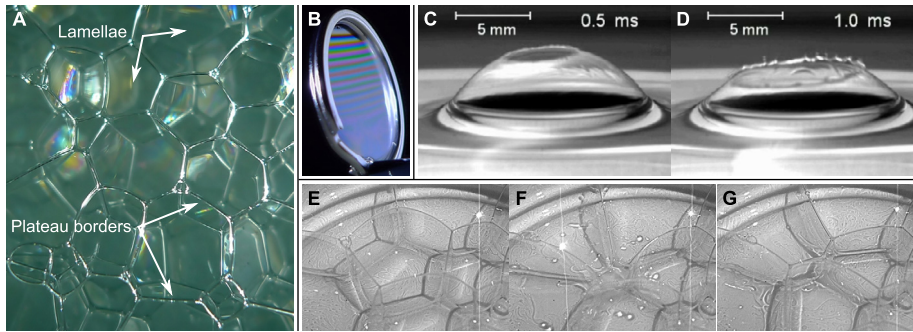  *E-mail address:* sethian@math.berkeley.edu (J.A. Sethian).

**Fig. 1.** (Reproduced from [44] and reprinted with permission from AAAS.) (A) A foam of soap bubbles made with common washing detergent. (B) Drainage and thin-film interference: A keyring suspended in soap solution makes a film, which then drains due to gravity. The subsequent variations in film thickness create interference patterns when lit with white light. (C, D) Rupture of a lamella; reproduced from [8] by permission from Macmillan Publishers Ltd., Nature, copyright 2010. (E, F, G) Rearrangement: A lamella (center of (E)) bursts, leading to macroscopic rearrangement of a foam.
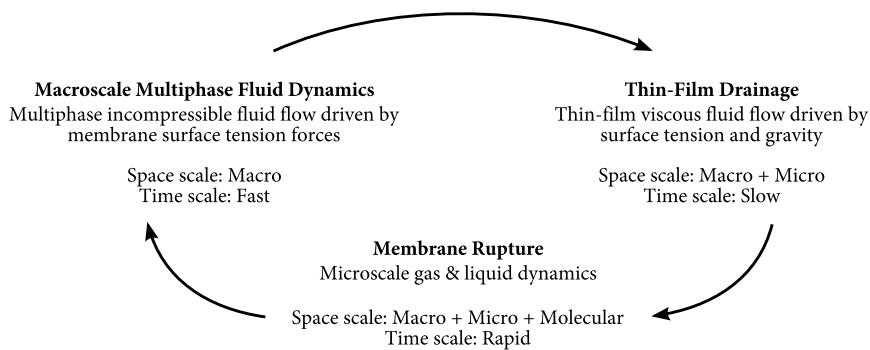


**Fig. 2.** Scale separated multiscale model of foam dynamics cycling between three stages: rearrangement, drainage, and rupture.

and the macroscale motion of the gas inside the bubbles. To illustrate, consider a foam whose macroscopic configuration appears to be in equilibrium, such as the foam in Fig. 1A. While seemingly stable, liquid inside the films drains over time, due to effects of gravity and surfactant. When one of the membranes becomes too thin, it ruptures and its liquid contents is redistributed, destroying the macroscopic equilibrium of the remaining membranes. Driven by macroscale gas dynamics and surface tension, these other membranes, as well as their film thicknesses, further change as they contort, stretch, and settle into a new equilibrium, setting the stage for continued fluid drainage.

These processes take place over six orders of magnitude in space and time. The liquid in the thin films, while only micrometers thick, drains over tens or hundreds of seconds (Fig. 1B) until a membrane ruptures (Figs. 1C, D). Membranes burst at hundreds of centimeters per second [8], causing macroscopic rearrangement of bubble geometry through surface and fluid forces occurring over less than a second (Figs. 1E, F, G).

This wide range of time and space scales leads to considerable algorithmic complexity. The resolution required to capture the microphysics associated with rupture becomes fundamentally impractical for resolving the large macroscale bubble evolution and rearrangement, even with today's advanced computing hardware and the most optimistic proposed future architectures.

## 1.2. A multiphysics scale separation approach

Fortunately, details at one space or time scale are not necessarily important at another scale. Exploiting this "scale-separation" can lead to models in which algorithms compute physics at different resolutions, and allow these different components to communicate across the scales.

In [44], we presented a scale-separated model for computing foam dynamics. The multi-scale model separates foam dynamics into a cycle of three distinct stages acting over different space and time scales (see Fig. 2): (i) a rearrangement phase, in which a foam out of macroscopic balance undergoes rearrangement due to surface tension and gas dynamics, leading to an equilibrium; (ii) a liquid drainage phase, in which the foam is essentially in macroscopic equilibrium, and the microscopic flow of liquid in the interconnected membranes is modelled until a lamella becomes too thin; and then (iii) a rupture phase, in which a lamella ruptures, sending the foam out of macroscopic balance, after which step (i) is invoked and the process repeated. Together, the dynamics of each phase affect the next, leading to a multi-scale model which captures the key effects of foam rearrangement, liquid drainage, and rupture.

### 1.3. Current work

In this paper, we provide the algorithmic and numerical details of the approach presented in [44], as well as a collection of new simulations. In particular, new aspects of this paper include:

- Second order finite difference projection methods for incompressible fluid flow, coupled to multiphase interface evolution using the Voronoi Implicit Interface Method;
- Derivation and implementation of a particle-based Lagrangian method to track and conserve fluid within the lamellae and Plateau borders, driven by the macroscale velocity field produced by the incompressible Navier–Stokes solver;
- Algorithms for generating high-quality finite element meshes of curved lamellae and Plateau borders;
- Finite element methodologies for thin-film drainage in the lamellae;
- Finite element methodologies for drainage in the Plateau borders;
- Numerical treatment of flux boundary conditions coupling the lamellae and Plateau borders;
- Numerical details of algorithms for thin-film membrane rupture.

We also discuss parallelization issues in performing different types of domain decomposition, and perform several convergence tests of various numerical methods. In addition, our results include

- Tests of individual mechanisms (macroscale rearrangement of membranes and thin film drainage) with physically appropriate parameters, and comparison with experiment;
- Modelling of a coupled multiscale model for a cluster of bubbles attached to an anchoring membrane, showing experimentally well-studied bubble cascades;
- Evolution of a large (27 bubble) foam bubble cluster coupling all physical effects, characterized through additional calculation of thin film optical interference based on lamellae thickness.

### 1.4. Background

The modeling of foams has been a topic of substantial interest for decades. Considerable mathematical analyses as well as numerical and experimental studies have focused on individual components. These include studying the geometry of stable foams, for example in Plateau's laws [37], which state that lamellae meeting at Plateau borders make 120° angles, as well as computational methods to find minimal surfaces [12,38]. Evolutionary laws for foam geometry have also been examined, such as the two-dimensional von Neumann–Mullins law [55,31], its extension to three dimensions [29], and statistical variations [20]. Statistics on the frequency and distribution of rupture events in two-dimensional and three-dimensional foams have been experimentally studied [53,54,39], as have been some of the key mechanisms driving topological changes [16]. Mathematical theory for small-scale capillary-driven oscillations of soap bubbles have been developed [30], and these have been compared to experiment using high-speed cameras [27]. Meanwhile, thin-film equations have been derived to model drainage in stationary films [34,32] as well as in moving curved surfaces [21] and drainage equations have been formulated for stationary networks of Plateau borders [56,33,26]. In addition, computational tools aimed at specific aspects of macroscopic rearrangement include numerical studies of multiphase fluid flow separated by massless and vanishingly thin interfaces that do not drain or rupture [42,43]; foam studies based on two-dimensional hydrodynamics [23,24]; and contributions made by the Surface Evolver software [10] in computing minimal energy states of complex configurations.

### 1.5. Outline

The outline of this paper is as follows.

- Section 2 – Scale-separated model and equations of motion
- Section 3 – Algorithms and numerical schemes
  - Algorithms for multiphase interface evolution
  - Algorithms for multiphase incompressible fluid flow
    - Second order fixed grid finite difference projection methods for incompressible flow
    - Lagrangian particle transport methods for membrane fluid advection
  - Algorithms for thin-film drainage
    - Mesh generation
    - Numerical schemes for lamellae thin film equations
    - Numerical schemes for Plateau border thin film equations
    - Treatment of quadruple point and flux boundary conditions
  - Approximation and algorithms for membrane rupture
  - Parallelization
- Section 4 – Convergence studies
- Section 5 – Results

## 2. Model and equations of motion

Our scale-separated model assumes that the gas and liquid flow are incompressible within the time/space scales under consideration, that liquid evaporation in the lamellae occurs during a longer time scale than rearrangement, drainage, and rupture, and that the liquid–gas interface has a no-slip boundary condition with a uniformly constant surface tension. We describe the multi-scale model in the following parts:

- *Part A: Multiphase interface evolution*. Mathematical formulation and description of multiphase interface evolution.
- *Part B: Rearrangement phase*. Incompressible fluid flow driven by surface tension exerted by the network of lamellae and Plateau borders drives the system toward an equilibrium configuration. Throughout the process, Plateau's laws are satisfied (i.e., lamellae meet at Plateau borders at 120° angles). Additionally, fluid in the lamellae and Plateau borders is conservatively redistributed according to the deformation of the membranes.
- *Part C: Thin-film drainage phase*. Liquid inside the (now stationary) lamellae and Plateau borders drains due to effects of surface tension, gravity, and suction of liquid from lamellae into Plateau borders.
- *Part D: Rupture phase*. A lamella ruptures due to its film thickness decreasing below a critical threshold; consequently its liquid contents is redistributed and the membrane is removed from the bulk network of interfaces, with large-scale macroscopic motion ensuing.

In our model, each lamella is idealized as infinitely thin, but carries with it a film thickness function, denoted by $\eta$, which varies along the extent of the lamella and in time. These thickness functions are evolved by different mechanisms according to which of the above phases is being executed, e.g., conservative transport, or solution of certain thin-film (lubrication) equations. Similarly, each Plateau border has an associated "thickness" function, $\lambda$, which defines its cross-sectional area as a function of time and space.

Before discussing numerical algorithms and approximations, we first provide the relevant equations of motion, repeating some of the discussion in [44].

### 2.1. Part A: characterization and evolution of interconnected membranes

The position and motion of the lamellae play a critical role in the model formulation. Identifying each bubble as a different "phase", this is a multiphase problem in which a large number of different phases meet and create complex triple junction structures.

Because of topological changes in the evolving foam, we need to rely on a mathematical formulation of multiphase interface evolution. Here we make use of the Voronoi Implicit Interface Method, first introduced in [42] and analyzed in [43]. The Voronoi Implicit Interface Method has several key features:

- The formulation uses a single function, plus an indicator, to describe the entire multiphase system.
- It handles multiple junctions, topological change, and works in any number of dimensions.
- It provides a consistent formulation in which no vacuums or overlaps can develop.
- The formulation leads to an initial value problem with a time component that couples naturally with associated physics.
- For surface tension driven motion, the formulation correctly produces Plateau's laws for the junction triple-point angle conditions.

Here, we briefly discuss this formulation.

#### 2.1.1. Components of the Voronoi Implicit Interface formulation

As background, the level set formulation [35] describes an interface using a function $\phi : \mathbb{R}^n \times [0, T) \to \mathbb{R}$, whose zero level set is an $(n - 1)$-dimensional hypersurface $\Gamma$ corresponding to the location of the interface between two phases at time $t = 0$. One way to construct this implicit function $\phi$ is through a signed distance function $\phi(x, t = 0) = \pm d(x)$, where $d(x)$ is the distance from $x$ to the interface $\Gamma$, with the sign chosen to be positive in one region and negative inside the other. Together with a speed function $F$ or velocity $\mathbf{u}$ defined on the interface,[1] and then extended away from the interface into the domain $\mathbb{R}^n$, the associated equation of motion is an initial value PDE which evolves $\phi$ in time in such a way that the zero level set always corresponds to the evolving interface, namely

$$\phi_t + F|\nabla\phi| = 0 \quad \text{or} \quad \phi_t + \mathbf{u} \cdot \nabla\phi = 0. \tag{1}$$

The equation on the left is the update equation if $F$ is given as the velocity normal to the interface, whereas the equation on the right corresponds to the update equation with underlying velocity field $\mathbf{u}$. In either case, the velocity most typically depends on a host of factors, including local geometry (normal, curvature, etc.) and global interface properties (length,

---

[1]  In the present work, the solution of the incompressible Navier–Stokes equations determines the velocity of the interface motion.
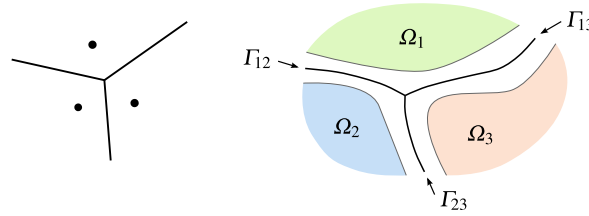
**Fig. 3.** (Left) Voronoi diagram of three points in the plane. (Right) The Voronoi interface $\Gamma_V = \Gamma_{12} \cup \Gamma_{13} \cup \Gamma_{23}$ corresponding to three given regions $\Omega_1$, $\Omega_2$, and $\Omega_3$. Here, $\mathbb{R}^2$ is divided into three cells (i.e. phases) separated by $\Gamma_V$.

enclosed areas, etc.), as well as the underlying physics in the domain, which may depend on boundary conditions, force terms, etc., supplied by the interface itself.

This formulation embeds the interface in a higher-dimensional function $\phi$ defined throughout the entire domain and, hence, adds unnecessary computational labor. More sophisticated versions employ the Narrow Band Level Set Method introduced in [1], which limits the signed distance function to a small neighborhood around the evolving front, and hence reduces the computational complexity to roughly the number of elements on the front. For details, including numerical methods and applications of the level set method, see [47].

### 2.1.2. The Voronoi Implicit Interface formulation

Recall now that the Voronoi diagram of a set of nodes in $\mathbb{R}^n$ is a tessellation of $\mathbb{R}^n$ into Voronoi cells, defined so that all the points in each cell are closer to one particular node than to any other node. The boundary between these cells is therefore the set of points that are equidistant to at least two nodes, and no closer to any other node. In more generality, instead of nodes we may have a collection of non-overlapping regions in $\mathbb{R}^n$. A similar subdivision of $\mathbb{R}^n$ exists in this case as well, based on proximity to the nearest region: letting $\{\Omega_i\}_{i=1}^m$ denote the $m$ non-overlapping regions, we can define $m$ cells such that cell $i$ consists of all points $x$ such that $x \in \Omega_i$ or else $d(x, \Omega_i) < d(x, \Omega_j)$ for all $j \neq i$. The *Voronoi interface*, $\Gamma_V$ (see Fig. 3) is defined to the boundary of these cells and corresponds to the set of points equidistant to two regions and no closer to any other – it can be characterized as $\Gamma_V = \bigcup_{i,j} \Gamma_{ij}$, where $\Gamma_{ij}$ is the set of points $x$ such that $d(x, \Omega_i) = d(x, \Omega_j) \leq d(x, \Omega_k)$ for all $k \neq i, j$.

The key idea of the Voronoi Implicit Interface formulation is to exploit the fact that the motion of a level set (including the zero level set) is bracketed by the motion of the surrounding neighboring level sets. This property is an immediate consequence of the comparison theorem under suitable restrictions on the speed function that moves the level sets, and these restrictions are often satisfied through the construction of extension velocities developed in [3].

The Voronoi Implicit Interface Method utilizes this property by letting the evolution of a multiphase system be determined by hypersurfaces that are nearby the interface. These hypersurfaces are obtained as the set of points that are a small but fixed distance away from the interface, and form a collection of individual surfaces each existing solely in one phase. While an interface in a multiphase system may have intricate complexity where multiple phases touch (such as at triple points, triple lines, etc.), neighboring hypersurfaces will be well behaved as the system evolves. The motion of the interface between the evolving phases is constructed from the position of these nearby hypersurfaces, using the Voronoi interface of these hypersurfaces, as follows.

### 2.1.3. The Voronoi Implicit Interface formulation: equations of motion

To begin, at each point of the computational domain, we define the unsigned distance function $\phi(x, t = 0)$, which is the distance from the point $x$ to the closest interface. We also assign an indicator function $\chi(x, t = 0)$ which specifies which phase $x$ is located in. If we evolve the initial value level set PDE for short period of time, the zero level set of $\phi$ will generally not remain a codimension-one surface, however nearby level sets (with value $\epsilon > 0$, say) will remain codimension-one surfaces. One can then reconstruct the interface after time $\Delta t$ as the Voronoi interface of these nearby level sets of $\phi$ with value $\epsilon$.

Thus, we can now define the evolution of a multiphase system in any number of dimensions. We shall do so by temporarily considering a fractional step approach, and defining the solution to the evolution equation as the converged limit of the fractional step process. Let $V_\epsilon(\phi)$ be the operator that reconstructs the unsigned distance function from the $\epsilon$-level sets of $\phi$ using the Voronoi interface. Let $E_{\Delta t}(\phi)$ be the evolution operator which evolves a given level set function $\phi$ for a time step $\Delta t$. Fix a particular final time $T > 0$ and let $n$ be the number of time steps required to reach that time, so that $\Delta t = T/n$. For some $\epsilon > 0$, we apply $n$ time steps, each step consisting of an evolution and a reconstruction. In the limit as $n$ goes to infinity, this defines a $\epsilon$-*smoothed solution* $\phi_\epsilon$, given by

$$\phi_\epsilon = \lim_{\Delta t \to 0, n \to \infty} (V_\epsilon \circ E_{\Delta t})^n (\phi(t = 0)),$$

where $\phi(t = 0)$ is the initial condition. We point out that this construction defines $\phi_\epsilon$ at intermediate times $0 \leq t \leq T$ as well as the final time $T$.

We can now take the limit of these $\epsilon$-smoothed solutions as $\epsilon \to 0$ from above, to obtain a solution given by

$$\phi_{\epsilon=0^+} = \lim_{\epsilon \to 0^+} \phi_\epsilon.$$

This is the formal definition of multiphase interface evolution, as defined by the VIIM algorithm. For complete details, see [42,43].

### 2.2. Part B: macroscopic rearrangement and fluid flow

#### 2.2.1. Multiphase Navier–Stokes incompressible flow

In the macroscopic rearrangement phase, the foam structure is out of equilibrium. Surface tension exerted by the network of lamellae and Plateau borders drives fluid motion in the gas pockets, rearranging the system of bubbles until a macroscopic equilibrium is reached. During this process, liquid contained in the thin films and Plateau borders is conserved and transported during this readjustment.

At this macroscopic level, we idealize the membranes as massless and vanishingly thin, and thus inertial effects are taken as negligible. This is known as the "idealized dry-foam limit", and leads to the incompressible Navier–Stokes equations for the gas phase, with continuity of the velocity field across the liquid–gas interface $\Gamma$, and an effective surface tension of $2\sigma$ (i.e., twice the coefficient of a single liquid–gas interface). The interface is thus advected by the velocity field $\mathbf{u}$ of the gas, satisfying

$$\rho_g(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \mu_g \Delta \mathbf{u} - 2\sigma \kappa \mathbf{n} \delta(\Gamma),$$
$$\nabla \cdot \mathbf{u} = 0,$$

(2)

where $\mu_g$ is the viscosity of the gas, $\rho_g$ is its density, $\kappa$ is the curvature of the interface, $p$ is the pressure, $\mathbf{n}$ is the normal to the interface, and $\delta$ is a delta function with support on the interface $\Gamma$. Later, we discuss how to solve these equations using a finite difference projection method.

#### 2.2.2. Transport of fluid in membranes

Although the interfaces are taken as massless and mathematically of zero thickness in terms of the incompressible Navier–Stokes equations, we assign a fluid thickness to each point, and require that this liquid is conserved and passively transported during rearrangement. This leads to a local conservation equation for a scalar function defining the film thickness, defined on each lamella and each Plateau border. For the lamella, its half-thickness is defined as $\eta$, and for the Plateau border, we define $\lambda$ as the cross-sectional area at any particular location in space. For liquid contained in the lamellae, conservative transport is modelled by requiring that

$$\frac{d}{dt} \int_{S(t)} \eta = 0,$$

where $S(t)$ is any surface patch on $\Gamma(t)$ passively advected by the velocity field $\mathbf{u}$. This model conserves the mass of liquid in the lamellae by measuring the amount of stretching in the interface, and it allows surface currents at the interface to move the liquid tangentially. Liquid in the Plateau borders is conserved with an analogous conservation law.

### 2.3. Part C: thin-film drainage

During drainage, fluid within the lamellae drains into Plateau borders in response to pressure differentials. This drainage takes place over the entire course of the foam dynamics. Producing corresponding equations of motion would entail deriving thin film equations over a moving collection of lamellae and Plateau borders, which are codimension-one and codimension-two objects within the computational domain.

However, the vast majority of drainage occurs after a macroscopic equilibrium has been reached, at which time the surface area is locally minimal, and hence each individual lamella is a surface with constant mean curvature. By capitalizing on the inherent scales involved, and following the construction of "thin-film approximations" [34,32] which describe the evolving membrane thickness in a single lamella, in [44] we built thin-film approximations for drainage in the curved lamellae, as well as the Plateau borders, and devised interrelated boundary conditions which couple the regions together. Here, we repeat these equations, before introducing our methodology for their numerical solution.

#### 2.3.1. Lamella drainage

For each lamella with surface $\Gamma_\ell$, the lamella thin-film equation is

$$\eta_t + \nabla_s \cdot \mathbf{Q} = 0 \text{ on } \Gamma_\ell, \text{ where}$$
$$\mathbf{Q} = \frac{1}{3\mu} \left( \sigma \eta^3 \nabla_s \big( (k_1^2 + k_2^2)\eta + \Delta_s \eta \big) + \rho \mathbf{g}_s \eta^3 \right),$$

(3)

where $\mu$ is the viscosity of the liquid, $\rho$ is its density, and $\mathbf{g}_s$ is the component of gravity tangential to the surface. Here, $\nabla_s$ is the surface gradient, $\nabla_s \cdot$ is the surface divergence and $\Delta_s$ is the surface Laplacian on the curved surface of the lamella,

while $k_1$ and $k_2$ are its principal curvatures. In this model, we are including effects of surface tension, the curved surface of the lamella, and gravity.

This is a fourth order PDE and needs two boundary conditions on the boundary of the lamella. One is a zero Neumann condition, relating to the model of the cross-sectional shape of Plateau borders and the thin-film approximations which couple together the different length scales of the lamellae and Plateau borders, see [44]. The other is provided by a flux boundary condition [11,22] that implements suction of liquid into the Plateau borders of thickness $\lambda$ at its boundary. Thus, we have for boundary conditions that

$$\frac{\partial \eta}{\partial \nu} = 0 \quad \text{and} \quad \mathbf{Q} \cdot \nu = \tfrac{1}{2} Q\,(\eta, \lambda) \quad \text{on } \partial \Gamma_\ell, \tag{4}$$

where $\nu$ is the outwards-pointing unit vector orthogonal to $\partial \Gamma_\ell$ and tangential to $\Gamma_\ell$. Here, $\mathbf{Q}$ is the vector flux of liquid in the lamella, defined in (3), while $Q$ is a function that determines the flux of liquid between a lamella and its adjoining Plateau border. The amount of flux is determined by matching the thickness of the lamella to the cross-sectional curvature of the Plateau border under a local Stokes flow argument. The value of $\lambda$ in (4) thus changes along each piece and between pieces on the boundary of $\Gamma_\ell$.

### 2.3.2. Plateau border drainage

A similar PDE is derived for the thickness of a Plateau border:

$$\lambda_t + \frac{C_\Delta}{\mu} \frac{\partial}{\partial \ell} \left( -\frac{1}{2} (\sqrt{3} - \frac{\pi}{2})^{1/2} \sigma \lambda^{1/2} \partial_\ell \lambda + \lambda^2 \rho g_\tau \right) = S, \tag{5}$$

where $C_\Delta$ is a constant associated with the cross-sectional shape of the Plateau border, $g_\tau$ is the tangential component of gravity, $\ell$ is a coordinate measuring length along the curved border, and $S$ is a source term representing the incoming liquid from the three lamellae connected to the Plateau border, determined by the sum of fluxes according to

$$S = \sum_{i=1}^{3} Q\,(\eta_i, \lambda),$$

where $\eta_i$ is the thickness of lamella $i$.

The set of Plateau borders forms a network of fibres. Linking the flow of liquid in this network requires boundary conditions where Plateau borders meet at quadruple junctions,[2] and these are provided by conservation of liquid mass and quasi-static pressure balance. Therefore, (5) is supplemented by a quadruple point boundary condition where four Plateau borders meet. With $\lambda_i$, $i = 1, \ldots, 4$, denoting the four Plateau border thickness functions, at each quadruple point we have

$$\begin{cases} \lambda_1 = \lambda_2 = \lambda_3 = \lambda_4, \quad \text{and} \\ \sum_{i=1}^{4} \frac{1}{2} (\sqrt{3} - \frac{\pi}{2})^{\frac{1}{2}} \sigma \lambda_i^{\frac{1}{2}} \partial_\ell \lambda_i - \lambda_i^2 \rho g_{\tau_i} = 0. \end{cases} \tag{6}$$

For each quadruple point, this is a nonlinear system of five equations in five unknowns (the common value of $\lambda$ and the four fluxes).

### 2.4. Part D: rupture

Rupture within a lamella occurs when drainage causes the fluid thickness to decrease below a critical level. A small tear appears and rapidly opens up as surface tension causes the membrane to retract. This rapid process occurs on a much smaller time scale (and with much larger velocities) than seen in the rest of foam mechanics. Depending on surface coatings and geometry, the fluid in this torn membrane can retract into the Plateau borders or disperse into the ambient gas [8].

Our model assumes an instantaneous rupture, which is handled in the above Voronoi Implicit Interface formulation. We note that a more intricate model could be developed within this implicit framework:

- On each membrane, calculate the rupture location where the membrane thickness has fallen below the prescribed critical level.
- Initiate a small hole, mathematically characterized as the zero level set of a function implicitly defined on the entire surface.
- Solve the resulting equations of motion on that stationary surface for the evolving interface, moving according to a normal speed law that retracts from the hole.

Further details of the approach used here are given later in the discussion of numerical methods.

---

[2] In this work, foams are considered in isolation, i.e., contact with walls or other boundaries is not considered. Consequently, if a Plateau border terminates, it does so at a junction consisting of four Plateau borders, see Fig. 1A. If a foam is in contact with a wall, then a Plateau border may terminate at the wall without forming a junction; a different drainage model and set of boundary conditions would be required in this case.

*2.5. Comments on our model*

Our scale-separation model contains several idealizations. First, we assume that drainage occurs only after macroscopic equilibrium is reached and the configuration is stable. Second, we are not accounting for the inertial effects of fluid mass in the lamellae during the macroscopic motion. Third, we assume rupture takes place instantaneously.

We also note that additional forces, scales and regimes, beyond those included here, play an important role in foam dynamics, although they can be added to this framework. Diffusive coarsening, which results from gas exchange between bubbles separated by permeable membranes, is important over very long time scales (minutes to hours) [57]; although our macroscale Navier–Stokes fluid solver easily allows such permeability effects (see [43]), our thin-film equations are derived assuming a static equilibrium, and thus we do not include coarsening effects. At the liquid–gas interface, we idealize that Marangoni forces, which act to equilibrate surfactant concentration, occur quickly enough to produce a uniformly constant surface tension, while the no-slip boundary condition assumption ignores some effects of surface rheology which can be important for surfactant solutions exhibiting mobile boundary conditions. Finally, we focus on dry foams, i.e. foams with liquid occupying less than approximately 10% of the total volume [57], though the methodology below has extensions to wet foam modelling as well.

## 3. Algorithmic approach: numerical approximation and schemes

Thus, our equations of motion are as follows:

- Multiphase interconnected interface evolution:

$$\phi_{\epsilon=0^+} = \lim_{\epsilon \to 0^+} \left[ \lim_{\Delta t \to 0, n \to \infty} (V_\epsilon \circ E_{\Delta t})^n \big( \phi(t=0) \big) \right].$$

- Navier–Stokes incompressible fluid flow:

$$\rho_g (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \mu_g \Delta \mathbf{u} - 2\sigma \kappa \mathbf{n} \delta(\Gamma), \quad \nabla \cdot \mathbf{u} = 0.$$

- Liquid transport in membranes:

$$\frac{d}{dt} \int_{S(t)} \eta = 0 \quad \text{and} \quad \frac{d}{dt} \int_{L(t)} \lambda = 0.$$

- Drainage in lamellae and Plateau borders:

$$\eta_t + \frac{1}{3\mu} \nabla_s \cdot \left( \sigma \eta^3 \nabla_s \big( (k_1^2 + k_2^2)\eta + \Delta_s \eta \big) + \rho \mathbf{g}_s \eta^3 \right) = 0,$$

$$\lambda_t + \frac{C_\Delta}{\mu} \frac{\partial}{\partial \ell} \left( -\frac{1}{2} (\sqrt{3} - \frac{\pi}{2})^{1/2} \sigma \lambda^{1/2} \partial_\ell \lambda + \lambda^2 \rho g_\tau \right) = S.$$

In this section, we discuss our algorithmic approach and numerical schemes for each of the above equations.

Before doing so, we take a high level perspective as to the steps in the algorithm (see Fig. 4). We imagine an initial multiphase configuration of gas pockets separated by fluid-filled lamellae linked together through a network of Plateau borders. The initial velocities of the gases and membranes, the thickness of the fluid at each point of each lamella and Plateau border, and the geometric configuration of the multiphase system, are all input parameters. We do not assume that the initial system is in macroscopic equilibrium. The algorithmic sequence of events is then as follows:

- Step 1: We evolve the system under the incompressible Navier–Stokes equations, including transport of fluid in the connective membranes, until a near equilibrium is reached, established by examining the change in geometric structure from one time step to the next.
- Step 2: Given this near-equilibrium, we freeze the macroscale motion and solve the thin film equations in the membranes, using the positions and film thickness at the end of Step 1 as initial conditions.
- Step 3: When the film thickness within any lamella, as computed under Step 2, falls below a user-supplied critical value, that membrane ruptures: the membrane is removed, and the accompanying fluid is deposited in the nearby borders. The positions of and fluid in the connective membranes are then supplied as input to the macroscopic dynamics of Step 1, and the algorithm returns to Step 1.

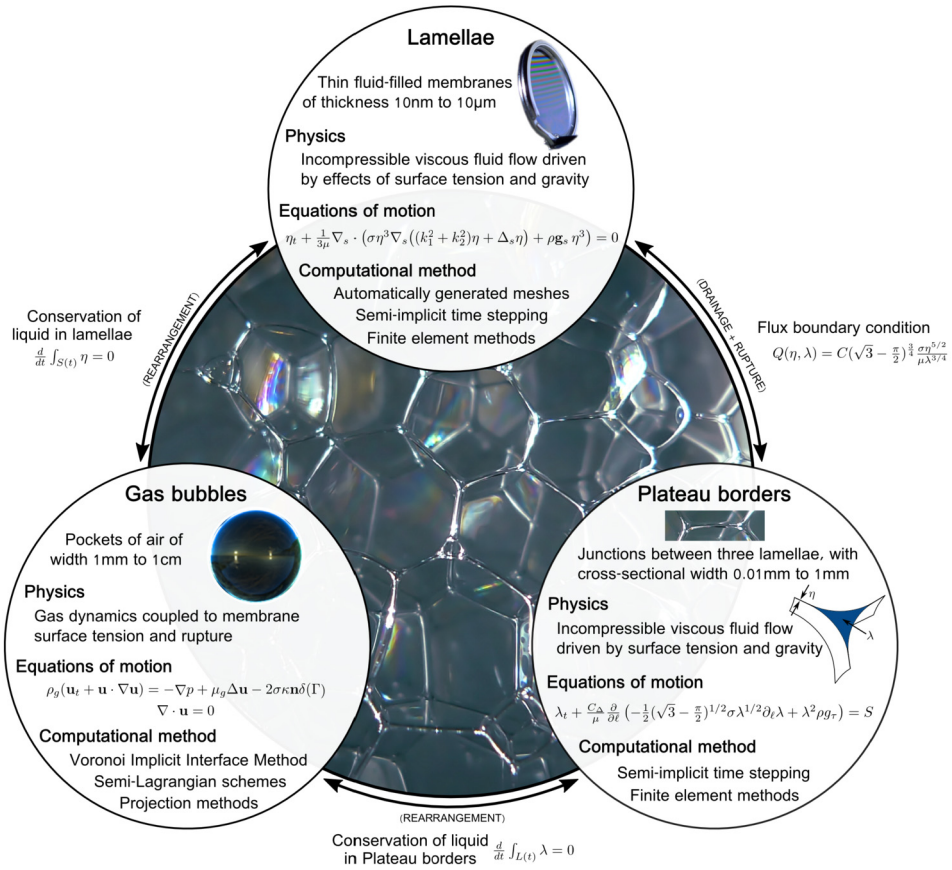The individual approximation schemes are as follows.

**Fig. 4.** A schematic of the coupled multiscale model and equations of motion. The model cycles between three stages: rearrangement, drainage, and rupture. During rearrangement, gas dynamics and surface tension drives rearrangement of lamellae and Plateau borders, affecting their film thicknesses in the process. During drainage, these thicknesses, characterized by the functions $\eta$ and $\lambda$ defined on individual membranes and junctions, change according to thin-film evolution equations. Upon rupture, liquid inside the ruptured film is conservatively redistributed, and the cycle repeats.

### 3.1. Multiphase interface evolution

We use the Voronoi Implicit Interface Method (VIIM) to track the motion of the network of interfaces (i.e., the lamellae and Plateau borders) in the foam. Here, the bubbles and the exterior region are considered separate regions ("phases") in the multiphase interface evolution. We will use the velocity field **u** of the gas obtained from the Navier–Stokes solution (see below) to evolve the interface.

The essential operation of the numerical method is as follows (for details, see [42,43]). We begin by initializing a multiphase problem using the unsigned distance function. That is, at every grid point $x_i$ in the domain, we assign a non-negative number $\phi(x_i)$ and an integer flag $\chi(x_i)$. Here, $\phi(x_i)$ is the distance from $x_i$ to the interface, and $\chi(x_i)$ denotes which phase contains the point $x_i$. Thus, the zero level set of this unsigned distance function $\phi$ is the interface, which may include triple junctions, quadruple points, etc.

Let us assume for now that the physics, described by the underlying velocity **u**, is known on the interface. The level set update equation given by (1) requires that the velocity be defined away from the interface. Here, we use the extension velocity approach given in [3] to build values away from the interface, using a method based on pre-sorted values of the distance function obtained through a fast marching strategy.

The key is to now appropriately update and evolve the unsigned distance function. First, we advance $\phi$ for one time step according to the underlying physics, i.e., the extended speed function. For a short time, and away from the zero level set, this is a well-posed update. The key observation, introduced in [42], is that the Voronoi interface of the nearby $\epsilon$-level sets of $\phi$ approximate the new position of the interface, and this in turn can be used to establish a new updated unsigned distance function. Finally, through judicious use of a fast Eikonal solver, see [46,13], one can construct the new updated unsigned distance function without ever explicitly finding the interface (see also [41]).

Thus, for a general multiphase interface evolution problem, we can summarize the algorithm as follows:

- *Initialization:* Choose $\Delta t$ and a fixed regular Cartesian grid with spacing $h$. Also choose $k_{\text{reinit}}$, which is the number of time steps taken before rebuilding the unsigned distance function using the Voronoi reconstruction. Then:

- Given a multiphase problem, defined by regions in space belonging to different phases, compute the unsigned distance function $\phi$ at each grid point.
- At each grid point, also assign a value for the indicator function $\chi$ reporting the phase for that point.
- *Time stepping:*
  - Use the position of the Voronoi interface to compute an extension velocity $\mathbf{u}$, depending on geometry and physics.
  - Advance the level set function $\phi$ one time step by solving the update equation $\phi_t + \mathbf{u} \cdot \nabla \phi = 0$.
  - After $k_{\text{reinit}}$ time steps, use the Voronoi reconstruction to rebuild the unsigned distance function from the Voronoi interface of the $\epsilon$-level sets.
  - Loop to top.

For our particular application, a few comments are in order.

- Since the viscosity of a typical gas species is relatively low, the velocity field $\mathbf{u}$ computed by Navier–Stokes is often not smooth. Consequently, rather than use the fluid velocity itself, we use the extension velocity methodology [3] discussed above. This significantly reduces the effect of currents that exist nearby the interface, and leads to minimal mass loss.[3] For each grid point in a small initial band surrounding the Voronoi interface, an extension velocity is calculated at the grid point by interpolating the velocity field at the corresponding closest point on the mesh (using trilinear interpolation), see [13,41]. This procedure determines a velocity field $\mathbf{u}_{\text{ext}}$, which is then used in the level set update $\phi_t + \mathbf{u}_{\text{ext}} \cdot \nabla \phi = 0$. We use high-order accurate upwinding finite difference schemes for the advection.
- We can use one-sided differences to make sure that finite difference stencils used in the level set update stay completely in one phase and do not reach across the interface. Alternatively, we can build suitable extensions of the unsigned distance function across the interface. In this fashion, finite difference stencils can reach across the interface without seeing discontinuities in the gradient of the unsigned distance function. With additional programming complexity, $\epsilon$ can effectively be taken as zero. Regardless of which approach is taken, we are essentially computing $\phi_{\epsilon=0^+}$, i.e., the formal definition of multiphase interface evolution discussed in §2.1.
- For complete details, including convergence studies of the effect of $\epsilon$ on accuracy, see [42,43].

### 3.2. Multiphase incompressible flow

We now turn to the algorithms and approximations required to model the macroscale fluid mechanics of the multiphase system.

#### 3.2.1. Second order fixed grid finite difference projection methods for incompressible flow

We wish to approximate the multi-phase incompressible Navier–Stokes equations given by (2). To do so, we employed a second order approximate projection method [4], which is based on Chorin's projection method [14]. We used a Godunov scheme for the advection term to better treat the low-viscosity gas dynamics, and Crank–Nicolson for the diffusion term. Surface tension is computed as a body force with a regularized three-dimensional Dirac delta function.

In more detail, the second order approximate projection method finds an intermediate velocity $\mathbf{u}^*$ such that

$$\rho_g \left( \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + (\mathbf{u} \cdot \nabla \mathbf{u})^{n+\frac{1}{2}} \right) = -\nabla p^{n-\frac{1}{2}} + \mu_g \frac{\Delta \mathbf{u}^* + \Delta \mathbf{u}^n}{2} + \mathbf{st}_\varsigma. \tag{7}$$

Here, the advection term $(\mathbf{u} \cdot \nabla \mathbf{u})^{n+\frac{1}{2}}$ is implemented with a second order unsplit Godunov scheme (see [51]), while the diffusion term is implemented with a standard Crank–Nicolson scheme.

Surface tension is computed as a body force with a regularized three-dimensional Dirac delta function, such that

$$\mathbf{st}_\varsigma(x) = \sigma \sum_i \int_{\Gamma_i} (\kappa \mathbf{n})(y) \delta_\varsigma(x - y) \, dS,$$

using the method described in [43]. Here, $\Gamma_i$ is the surface of bubble $i$ and $\kappa \mathbf{n}$ is the mean curvature times the unit normal of the surface.

This method of calculating surface tension results in a smoothed body force that is similar to other approaches [9,52,49]. However, it differs in that we formulate the force in terms of a convolution over the surface of each bubble and leads to better treatment of the force at junctions (such as Plateau borders and quadruple points), as discussed further in [43]. This method of calculating surface tension is physically consistent, since in the case of a dry foam, each bubble is separated

---

[3] Since the incompressible Navier–Stokes equations determine the interface evolution and there is no gas diffusion across bubble membranes, the volume of each bubble is conserved in the theoretical model. On a discrete level, exact volume conservation is obtained in the limit of mesh refinement. For coarse grids, we found that conservation was considerably improved through the use of extension velocities. Experiments indicated that mass loss was only of concern for bubbles a handful of grid points in diameter. In these cases, we ensured volume conservation by using the volume-constrained time stepping methods for the VIIM described in [43].

from the others by a thin membrane. Moreover, the method automatically recovers Plateau's laws, i.e. lamellae make 120° angles at Plateau borders and quadruple points have symmetric angle conditions. To complete the projection step, once the intermediate velocity in (7) is found, the velocity at the next time step is determined by performing a projection

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \mathcal{P}\Big(\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t}\Big). \tag{8}$$

Here, $\mathcal{P}$ is a projection operator which takes a velocity field and projects it onto the space of divergence-free velocity fields, and in so doing, determines the pressure $p^{n+\frac{1}{2}}$ at the next time interval; for more details, including a discussion on the merits of projecting $\mathbf{u}_t$ (as is done in (8)) rather than projecting $\mathbf{u}^*$, see [51].

### 3.2.2. Lagrangian particle transport methods for membrane fluid advection

The fluid within the lamellae and Plateau borders is transported by the motion of the gas produced from the solution of the above incompressible Navier–Stokes solver. As discussed above, this results in a local conservation equation for evolving the lamellae film thickness $\eta$, namely

$$\frac{d}{dt}\int_{S(t)} \eta\, dS = 0$$

for any surface patch $S(t)$ on a lamella $\Gamma$ that is passively advected by the velocity field $\mathbf{u}$. In [2,48], this is written in strong form as a PDE by extending $\eta$ off the interface $\Gamma$ to produce an equation in all of space. Their resulting conservation law can be shown to be equivalent to (see also [50])

$$\eta_t + \mathbf{u} \cdot \nabla \eta = \Big(\mathbf{n} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \nabla \cdot \mathbf{u}\Big)\eta.$$

Here, $\mathbf{n}$ is the normal vector field for the interface $\Gamma$, while the term in parentheses measures the local stretching and compression of the interface in the directions tangential to the surface. In [2,48], this PDE is solved on a background grid.

In our case, we have an interconnected network of lamellae with individually-defined $\eta$ functions, and this makes an implicit formulation at the junctions somewhat difficult. Instead, we take a Lagrangian particle approach, based on the method of characteristics, to determine fluid transport. Consider a specific lamella with surface $\Gamma$. On $\Gamma$, a set of particles is uniformly seeded with positions $x_i$, $i = 1, \ldots, N$, and each particle carries a thickness $\eta_i$. A specific particle's coordinates and thickness values are evolved in time by solving the ODE system

$$\frac{d}{dt}x_i = \mathbf{u}, \qquad \frac{d}{dt}\eta_i = \Big(\mathbf{n} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \nabla \cdot \mathbf{u}\Big)\eta_i.$$

Note that $\nabla \cdot \mathbf{u} = 0$ due to incompressibility. A simple forward Euler scheme for each particle yields:

$$\frac{x_i^{n+1} - x_i^n}{\Delta t} = \mathbf{u}^n(x_i^n), \qquad \frac{\eta_i^{n+1} - \eta_i^n}{\Delta t} = \hat{n} \cdot \frac{\mathbf{u}^n\big(x_i^n + h\hat{n}\big) - \mathbf{u}^n\big(x_i^n - h\hat{n}\big)}{2h}.$$

Here, $\mathbf{u}^n$ is evaluated from a trilinear interpolation of the values defined on the grid, and the normal of the interface at $x_i^n$, denoted by $\hat{n}$ in the above, is evaluated by a finite difference approximation based on the level set function used in the VIIM.

A similar algorithm is used to track the thickness of the Plateau borders. In this case, the local conservation law states that

$$\frac{d}{dt}\int_{L(t)} \lambda\, ds = 0$$

for any line segment $L(t)$ on the Plateau border that is passively advected by the velocity field. Particles on a specific Plateau border are seeded with positions $x_i$ and carry thickness values $\lambda_i$. By considering infinitesimal line segments, it is straightforward to derive the corresponding differential equations for the particles, which are given by

$$\frac{d}{dt}x_i = \mathbf{u}, \qquad \frac{d}{dt}\lambda_i = -\Big(\tau \cdot \frac{\partial \mathbf{u}}{\partial \tau}\Big)\lambda_i.$$

Here, $\tau = \tau(x_i)$ is a unit tangent vector to the Plateau border, and in analogy with the case of the lamella, the term in parentheses measures the local stretching and compression of the Plateau border. A similar forward Euler and finite difference scheme is used to update the $\lambda$ thickness values for particles on the Plateau border.

Depending on the interface dynamics, over time, particles on the surface of the lamellae and Plateau borders can become dispersed in some areas and concentrated in others. This then requires intermittent reseeding through interpolation. Briefly, our reseeding procedure: (i) uses the contouring procedure described in [43] to extract the multiphase interface

as a triangulated mesh, (ii) samples particles on this mesh approximately uniformly in space, and (iii) calculates thickness values with an algorithm that bins existing particles into grid cells, yielding efficient local lookup, combined with a linear interpolation procedure. The interpolation algorithm is second order, so the bulk of the error in our numerical methods for conservatively advecting lamellae and Plateau border film thickness is contained in the first order forward Euler scheme. For further details, see [45].

### 3.3. Thin-film drainage

Next, we discuss the algorithms required to approximate the thin film drainage equations in both the lamellae and the Plateau borders. Abstractly, we need to solve a PDE system on a set of interconnected curved surfaces. There are essentially two approaches for solving such equations:

- *Explicit representation:* Here, the surfaces are tessellated, typically producing a triangulated mesh, and the resulting equations of motion are solved using a finite element method (see, for example, [17]).
- *Implicit representation:* Here, the surfaces are represented as zero level sets of a higher dimensional function, following the typical level set approach, transforming a two-dimensional surface PDE into a three-dimensional PDE extended off the surface, and solved with associated equations that capture the 2D dynamics on the curved surface, see, for example, [2,48,6,18].

Because we have an interconnected network of surfaces, an explicitly meshed tessellation of the lamellae and adjoining Plateau borders seems to be the most natural approach.

Once a discretized representation of the interface structures is chosen, we must solve the degenerate non-linear fourth order PDEs associated with thin film evolution. In the case of flat surfaces, there are a variety of approaches, see, for example [19,5,58], that describe finite element, finite volume and finite difference-based schemes, respectively. Because these equations are fourth order, explicit time-step schemes usually entail restrictive time steps for stability. As an alternative, in [7], semi-implicit methods and convexity-splitting schemes are given. In particular, convergence is proved for a semi-implicit biharmonic-modified forward time stepping scheme, based on the Laplace-modified forward time stepping scheme introduced in [15]. For a specific class of fourth-order nonlinear parabolic PDEs, it is possible to prove that this biharmonic-modified time-stepping scheme is unconditionally stable. This method is straightforward to implement, and has been adopted here to time-step the lamella thin-film equation.

We also need to address the boundary conditions that link the lamellae and Plateau borders together. Implementing the boundary conditions implicitly with the equations describing each individual lamella and Plateau border leads to a very large non-linear system requiring some form of iterative procedure. Instead, we adopt an essentially explicit approach that aligns the various components together at the beginning of each time step.

From an overall perspective, the sequence of events for solving the thin film drainage step are as follows:

- Given a near-equilibrium configuration of the complete system, generate a mesh of the geometric configuration, and interpolate the fluid film thicknesses from the connective membranes to this mesh.
- Solve the drainage equations by advancing in time the following iteration:
  1. Solve the Plateau border conditions at each quadruple point to obtain Neumann boundary conditions for each Plateau border.
  2. Calculate the flux $Q$, used as a boundary condition for the lamellae and as a source for the Plateau borders.
  3. Using a finite element discretization, solve for $\eta^{n+1}$ and $\lambda^{n+1}$ for each lamella and Plateau border.
  4. If a lamella becomes too thin, simulate rupture and terminate loop, otherwise continue time stepping.
- Use the final thickness functions of the remaining lamellae and Plateau borders as input to the macroscopic rearrangement phase.

We now develop approximations for each of these steps.

### 3.3.1. Mesh generation

First, we wish to discretize the interconnected lamellae into triangles that are approximately uniform in size and as close to equilateral as possible, and discretize each Plateau borders into a set of connected line segments such that each line segment is an edge of three triangles corresponding to different lamellae. This is important for consistency in the finite element formulation, as the "communication" between adjoining lamellae and Plateau borders only occurs through the Plateau borders. Each Plateau border should be made of smooth curves, though the borders may meet at quadruple junctions.

The algorithm we use was introduced in [40], and is partially based on the DistMesh algorithm [36] which was designed to generate unstructured meshes from implicitly defined geometries using force-based smoothing of mesh elements, together with a Delaunay tessellation algorithm. DistMesh takes as input a mesh with arbitrary quality and topology as might be generated from, for example, a standard Delaunay algorithm from a set of randomly chosen vertices, or from a standard contouring algorithm such as Marching Cubes for extracting level sets. Using a user-defined forcing function,
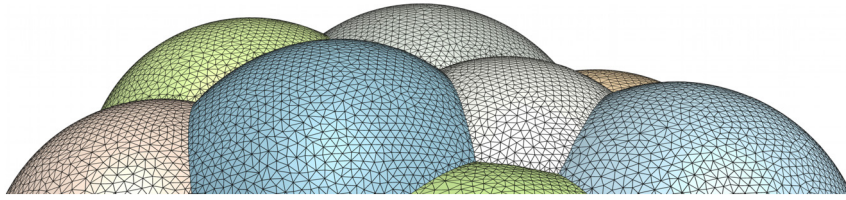
**Fig. 5.** An example of a triangular mesh generated automatically from the network of connected lamellae. Individual lamellae are colored differently and triangles meet consistently at junctions to form segmented line curves at Plateau borders.

DistMesh moves the vertices of the mesh, based on the current topology of the mesh, to spread vertices apart. This process is intertwined with a Delaunay algorithm, which is applied iteratively to update the topology (through a series of "edge flipping" operations), favoring good quality triangles. The result is a simple algorithm that converges relatively efficiently to a mesh with high quality elements.

DistMesh was originally applied to a single implicitly defined surface without junctions. Here, we briefly describe the approach detailed in [40] that extends some of the ideas in DistMesh to produce a mesh for multiply-connected lamellae with our desired properties. We begin by extracting an initial mesh from the implicit representation given as input from the VIIM algorithm, as was discussed in [43]. These initial mesh vertices are moved according to force laws, similar to as is done in DistMesh, but with one type of force law near junctions and another away from junctions. This is combined with a locally adaptive time step approach, and in practice generates a good quality mesh in as few as 20 iterations, each iteration having approximately the same computational cost as creating the initial mesh. As an example, Fig. 5 illustrates a typical mesh obtained by this algorithm. For details, see [40].

*3.3.2. Numerical schemes for lamellae thin film equations*

Next, we discuss the finite element method for the thin film equations in the set of connected lamellae. We begin with the scheme for an individual specific lamella $\Gamma$. We first consider the time discretization, followed by the variational form of the governing PDE and its discrete counterpart.

**Time discretization**

Since (3) is a fourth order, nonlinear parabolic PDE, explicit time stepping schemes typically entail severe time step constraints for stability (e.g., $\Delta t = \mathcal{O}(h^4)$). Instead, we use a biharmonic-modified forward time stepping scheme, derived by applying forward Euler and adding a term of the form $\alpha \Delta_s^2 (\eta^{n+1} - \eta^n)$, where $\alpha$ is a constant, and for (3), corresponds to the scheme

$$\frac{\eta^{n+1} - \eta^n}{\Delta t} + \alpha \frac{\sigma}{3\mu} \Delta_s^2 (\eta^{n+1} - \eta^n) + \frac{1}{3\mu} \nabla_s \cdot \left( \sigma (\eta^n)^3 \nabla_s \big( (k_1^2 + k_2^2) \eta^n + \Delta_s \eta^n \big) + \rho \mathbf{g}_s (\eta^n)^3 \right) = 0. \tag{9}$$

Since $\Delta_s^2 (\eta^{n+1} - \eta^n)$ is $\mathcal{O}(\Delta t)$ in magnitude, this term does not alter the first order convergence rate of the forward Euler time scheme, and the error depends on the size of $\alpha$. For any $\alpha > 0$, the scheme becomes implicit and leads to a symmetric positive definite system of equations to solve at each time step, the matrix of which is a discretization of $I + \Delta t \alpha \frac{\sigma}{3\mu} \Delta_s^2$. Choosing $\alpha$ large enough, we have shown that the resulting scheme is stable (see [7] for a proof for a similar PDE). For the scheme (9), $\alpha$ needs to bound $\eta^3$; in this work, at each time step $\alpha$ is set to be $\alpha = 2 \max_\Gamma \eta^3$.

In addition, (9) is supplemented by boundary conditions (i.e., (4)) such that $\nabla_s \eta^{n+1} \cdot \nu = 0$ and

$$\frac{2}{3\mu} \left( \alpha \sigma \nabla_s \Delta_s (\eta^{n+1} - \eta^n) + \sigma (\eta^n)^3 \nabla_s \big( (k_1^2 + k_2^2) \eta^n + \Delta_s \eta^n \big) + \rho \mathbf{g}_s (\eta^n)^3 \right) \cdot \nu = Q (\eta^n, \lambda^n) \quad \text{on} \quad \partial \Gamma,$$

where $\nu$ is the outwards-pointing unit vector orthogonal to $\partial \Gamma$ and tangential to $\Gamma$. Note that the flux boundary condition is a mixture of implicit and explicit terms that takes the same form as the biharmonic-modified time stepping scheme; for the most part the flux boundary condition is evaluated explicitly (at the current time step) with the $\alpha$ modification making it semi-implicit. This form of the boundary condition is suitable for the finite element formulation used below.

**Finite element method for spatial terms**

The finite element method used here is based on the formulation and theory presented in [17] for parabolic PDEs on surfaces.

**Weak form.** On a smooth surface, there exist analogues of the usual integration-by-parts formulas, including $\int_\Gamma \nabla_s f \cdot \nabla_s g = -\int_\Gamma f \Delta_s g + \int_{\partial \Gamma} f \nabla_s g \cdot \nu$, and, if $\mathbf{f}$ is everywhere tangential to the surface, $\int_\Gamma g \nabla_s \cdot \mathbf{f} = -\int_\Gamma \mathbf{f} \cdot \nabla_s g + \int_{\partial \Gamma} g \mathbf{f} \cdot \nu$. Using these formulas, a suitable variational form of the governing PDE (3) is to find two functions $\eta$ and $p$ satisfying

$$\int_\Gamma \eta_t \phi = \frac{1}{3\mu} \int_\Gamma \left( \sigma \eta^3 \nabla_s \big( (k_1^2 + k_2^2) \eta + p \big) + \rho \mathbf{g}_s \eta^3 \right) \cdot \nabla_s \phi \ - \int_{\partial \Gamma} \phi \mathbf{Q} \cdot \nu, \quad \text{and}$$

$$\int_\Gamma p\,\psi = -\int_\Gamma \nabla_s \eta \cdot \nabla_s \psi,$$

for all test functions $\phi \in C([0, T]; H^1(\Gamma))$, $\psi \in H^1(\Gamma)$. This formulation introduces a second function $p$ satisfying $p = \Delta_s \eta$ in the weak sense and creates a coupled system of equations to solve for both $\eta$ and $p$. This weak form requires less regularity on $\eta$, and this carries over to the discrete setting, since then linear finite elements can be used for both $\eta$ and $p$.

**Discrete variational form.** A finite element scheme using linear elements is as follows. The smooth surface of the lamella is approximated by a triangulation $\mathcal{T}_h$ such that $\Gamma_h = \bigcup_{t \in \mathcal{T}_h} t$. The vertices $(x_i)_{i=1}^n$ are assumed to lie on $\Gamma$ so that $\Gamma_h$ is a Lipschitz continuous surface interpolating $\Gamma$. Here, $h$ denotes the maximum circumdiameter of the triangles in $\mathcal{T}_h$. The error due to approximating the surface $\Gamma$ by the triangulation $\Gamma_h$ introduces an error that is no worse than second order as $h \to 0$ [17]. Let $V_h$ be the linear finite element space consisting of all continuous functions defined on $\Gamma_h$ that are linear when restricted to an element of $\mathcal{T}_h$. Let $\{\chi_j\}$ be the associated standard Lagrange basis functions so that if $v \in V_h$, then $v = \sum_i v(x_i)\chi_i$.

Using this finite element, together with the forward time stepping scheme, the discrete variational form is as follows. Given $\eta^n, p^n \in V_h$ at the current time step, find $\eta^{n+1}, p^{n+1} \in V_h$ at the next time step, such that

$$\begin{aligned}
\int_{\Gamma_h} \frac{\eta^{n+1} - \eta^n}{\Delta t}\phi &= \alpha \frac{\sigma}{3\mu}\int_{\Gamma_h} \nabla_s(p^{n+1} - p^n) \cdot \nabla_s\phi \\
&\quad + \frac{1}{3\mu}\int_{\Gamma_h} \left[ \sigma(\eta^n)^3 \nabla_s\big(k\eta^n + p^n\big) + \rho\mathbf{g}_s(\eta^n)^3 \right] \cdot \nabla_s\phi \\
&\quad - \frac{1}{2}\int_{\partial\Gamma_h} Q\,(\eta^n, \lambda^n)\phi,
\end{aligned} \tag{10}$$

for all $\phi \in V_h$, and

$$\int_{\Gamma_h} p^{n+1}\psi = -\int_{\Gamma_h} \nabla_s\eta^{n+1} \cdot \nabla_s\psi,$$

for all $\psi \in V_h$. Here, $\mathbf{g}_s$ and $Q$ are defined to be the piecewise linear interpolant of their values defined on mesh vertices (in the case of $Q$, they are given by (16)), while $k$ is defined to be the piecewise linear function whose value on mesh vertices is $k_1^2 + k_2^2$; see later for additional remarks. This choice of interpolation allows the integral quantities to be calculated with straightforward quadrature schemes. In particular, many of the integrals reduce to the form of $\int_T f^3 g$ where $T$ is a triangle and $f$ and $g$ are linear functions on $T$. This in turn reduces to a function of the three values of $f$ and $g$ on the vertices of the triangle, a closed form expression of which can be found by, e.g., symbolic computer algebra.

The variational form leads to a linear solve for $\eta^{n+1}$ in the usual way, as follows. Let $\eta_i^{n+1}$ denote the value of $\eta^{n+1}$ at vertex $i$, let $M$ be the mass matrix defined by $M_{ij} = \int_{\Gamma_h} \chi_i\,\chi_j$, and let $K = \int_{\Gamma_h} \nabla_s\chi_i \cdot \nabla_s\chi_j$ be the usual stiffness matrix. Abusing notation and considering $\eta^{n+1}$ and $p^{n+1}$ as a vector, we thus require that

$$\begin{cases}
\frac{M\eta^{n+1} - M\eta^n}{\Delta t} = \alpha\frac{\sigma}{3\mu}Kp^{n+1} + f, \quad \text{and} \\
Mp^{n+1} = -K\eta^{n+1},
\end{cases} \tag{11}$$

where $f = f(\eta^n)$ contains the explicit terms, i.e.

$$f_i = -\alpha\frac{\sigma}{3\mu}\int_{\Gamma_h} \nabla_s p^n \cdot \nabla_s\phi_i + \frac{1}{3\mu}\int_{\Gamma_h} \left[ \sigma(\eta^n)^3 \nabla_s\big(k\eta^n + p^n\big) + \rho\mathbf{g}_s(\eta^n)^3 \right] \cdot \nabla_s\phi_i - \frac{1}{2}\int_{\partial\Gamma_h} Q\,(\eta^n, \lambda^n)\phi_i.$$

According to (11), it follows that $p^{n+1} = -M^{-1}K\eta^{n+1}$, however this requires inversion of the full mass matrix. It is simpler and more efficient to make an approximation using instead the lumped mass matrix, i.e. $\tilde{M} = \text{diag}(M_{ii})$, and write $p^{n+1} = -\tilde{M}^{-1}K\eta^{n+1}$. Experiments indicated this approximation did not alter the overall convergence rate of the finite element scheme. Eliminating $p^{n+1}$ from (11), we obtain

$$\left(M + \alpha\Delta t\frac{\sigma}{3\mu}K\tilde{M}^{-1}K\right)\eta^{n+1} = M\eta^n + \Delta t f(\eta^n).$$

This is a symmetric positive definite system for the unknown $\eta^{n+1}$, whose matrix (left-multiplied by $M^{-1}$) approximates the operator $I + \alpha\Delta t\frac{\sigma}{3\mu}\Delta_s^2$. To solve this equation, we have used a simple Conjugate Gradient method, although more sophisticated methods such as multigrid would lead to more efficient solvers.

**Additional comments**

The curvature function $k$ containing the principal curvatures $k_1$ and $k_2$ in (10) is defined to interpolate $k_1^2 + k_2^2$ at mesh vertices. To calculate the vertex values, we must calculate the principal curvatures of the interface $\Gamma$ given implicitly by the function $\phi$ in the VIIM. We can use the VIIM indicator function $\chi$ to construct a signed level set function $\psi$, and then calculate $k$ with simple finite differences, as follows. Since $\mathbf{n} = \nabla\psi/|\nabla\psi|$ is a normal vector field of $\Gamma$, its gradient has eigenvalues $\{0, k_1, k_2\}$ and is related to the shape operator on $\Gamma$. The mean curvature of $\Gamma$ can be calculated with $\kappa = k_1 + k_2 = \nabla \cdot \mathbf{n} = \text{tr}(\nabla\mathbf{n})$. A similar relation can be used to calculate $k$ through $k = k_1^2 + k_2^2 = \text{tr}((\nabla\mathbf{n})^2)$. This quantity is calculated at grid points using standard second order finite differences, after which the grid point values are trilinearly interpolated onto the vertices of the triangulation $\Gamma_h$, thereby yielding $k$ at mesh vertices.

Similarly, the tangential component of gravity $\mathbf{g}_s$ in (10) is also defined to be the piecewise linear interpolant of $\mathbf{g}_s$ defined at mesh vertices. Through a similar procedure as above, we can construct $\mathbf{g}_s = \mathbf{g} - (\mathbf{g} \cdot \mathbf{n})\mathbf{n}$ at grid points, and then use trilinear interpolation to define $\mathbf{g}_s$ on mesh vertices.

*3.3.3. Numerical schemes for Plateau border thin film equations*

We solve the Plateau border thin-film equation (5) in a manner similar to that used for the lamella. An analogue of the biharmonic-modified forward time stepping scheme is adopted, however, the forward Euler step is now modified by a term of the form $\alpha\partial_{\ell\ell}$. Provided $\alpha$ is large enough, this time stepping scheme is expected to be a stable, first order accurate time stepping scheme for the second order nonlinear parabolic PDE. In addition, a finite element-based spatial discretization has been used, but since there is only one spatial dimension, this can also be viewed as a conservative finite difference approximation.

**Numerical discretization**

Let $\Gamma$ denote the curve of a Plateau border. In the triangulated mesh, $\Gamma$ is approximated by a set of connected line segments $\mathcal{S}_h$ such that $\Gamma_h = \bigcup_{e\in\mathcal{S}_h} e$. The vertices $(x_i)_{i=1}^n$ are assumed to lie on $\Gamma$ so that $\Gamma_h$ is a Lipschitz continuous curve interpolating $\Gamma$. Let $V_h$ be the linear finite element space consisting of all continuous functions defined on $\Gamma_h$ that are linear when restricted to an element of $\mathcal{S}_h$. Let $\{\chi_j\}$ be the standard Lagrange basis functions so that if $v \in V_h$, then $v = \sum_i v(x_i)\chi_i$.

Using the analogue of a Laplace-modified forward time stepping scheme, the finite element method is as follows. Given $\lambda^n \in V_h$ at the current time step, find $\lambda^{n+1} \in V_h$ at the next time step, such that

$$\int_{\Gamma_h} \frac{\lambda^{n+1} - \lambda^n}{\Delta t}\phi = \frac{C_\Delta}{\mu}\int_{\Gamma_h}\left[-\tfrac{1}{2}\sigma(\sqrt{3} - \tfrac{\pi}{2})^{\frac{1}{2}}\left(\sqrt{\lambda^n}\partial_\ell\lambda^n + \alpha\partial_\ell(\lambda^{n+1} - \lambda^n)\right) + (\lambda^n)^2\rho g_\tau\right]\partial_\ell\phi$$
$$+ \int_{\partial\Gamma_h} W\phi + \int_{\Gamma_h} S^n\phi, \tag{12}$$

for all $\phi \in V_h$. Here, $W$ is the Plateau border flux boundary condition that is derived from the quadruple point boundary condition (as given later by (15)), while $S^n$ is defined to be the piecewise linear interpolant of the source term defined at the vertices given by (17). The tangential component of gravity is calculated on a per-edge basis, such that $g_\tau = \mathbf{g} \cdot \tau$ is piecewise constant. Finally, the stabilization factor used in the Laplace-modified forward time stepping scheme has been set to $\alpha = 2\max_{\Gamma_h}\sqrt{\lambda^n}$, evaluated at the beginning of each time step.

Similar to the case of the lamella, the discrete variational form (12) leads to a symmetric positive definite system for $\lambda^{n+1}$. In fact, the associated matrix can be made tridiagonal if the Lagrange basis functions are ordered by position along the Plateau border. However, despite the possibility of using an efficient tridiagonal solver, a simple Conjugate Gradient method has again been used to solve this system.

*3.3.4. Quadruple point and flux boundary conditions*

Finally, we need to address the boundary conditions that link the structures together.

**Discretization of Plateau border boundary conditions**

The boundary conditions at quadruple points (6) are a set of nonlinear simultaneous equations combining Dirichlet and Neumann boundary condition types. This boundary condition has been implemented in an explicit fashion: viewed as a type of domain decomposition, the approach decouples the Plateau borders for a short amount of time at each time step. It was found that it is possible to maintain first order accuracy (in time), as well as conservation of mass, by combining both Dirichlet and Neumann boundary conditions in an alternating fashion, as follows.

Consider a specific quadruple point. On the mesh, the quadruple point is a single vertex from which emanates four Plateau borders, which are four curves in space made up of line segments; let $i = 1, 2, 3, 4$ index these Plateau borders. Let $\lambda_{i,0}$ denote the value of the thickness function for Plateau border $i$ at the quadruple point, let $\lambda_{i,1}$ denote its value at the position at precisely one edge length of size $h_{i,0}$ away from the quadruple point, and let $\lambda_{i,2}$ denote its value two edge lengths away, the second edge having length $h_{i,1}$; see Fig. 6. Also let $x_{i,j}$ denote the position of the vertices, so that $x_{1,0} = \cdots = x_{4,0} =: x_0$. The strategy is to:
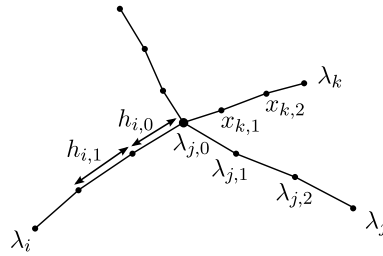
**Fig. 6.** Schematic of a quadruple point and the four Plateau borders emanating from it. Here, $h_{i,0}$ and $h_{i,1}$ are edge lengths and $\lambda_{i,j}$ are thickness values at mesh vertices $x_{i,j}$.

1. Use the values of $\lambda_{i,1}$ and $\lambda_{i,2}$, $i = 1, \ldots, 4$, together with the quadruple point boundary condition, to obtain a common value of $\lambda_0$ at the quadruple point.
2. Use $\lambda_0$ to calculate fluxes $\lambda_i'$, used as Neumann boundary conditions for the Plateau border time stepping scheme.
3. Time step each Plateau border according to the finite element method described above. In general, due to the decoupling, this yields different values of $\lambda_{i,0}$ at the quadruple point at the next time step.
4. Project these values onto a common value, in such a way that the total mass is conserved.

In essence, this strategy uses the quadruple point boundary condition to obtain flux boundary conditions for each Plateau border in such a way as to conserve the total mass in the system, at the expense of obtaining discontinuities in $\lambda$ at each quadruple point after each time step. At the end of each time step, this discrepancy is resolved via a projection procedure that conserves mass. Altogether, the procedure leads to a first order accurate time stepping scheme that conserves the total mass of liquid in the network of Plateau borders.

In more detail, by approximating the derivatives in (6) with second order finite differences, the quadruple point boundary condition, with common value $\lambda_0$, leads to

$$\sum_{i=1}^{4} \tfrac{1}{2}(\sqrt{3} - \tfrac{\pi}{2})^{\frac{1}{2}} \sigma \lambda_0^{\frac{1}{2}} \left[ \frac{\lambda_{i,1} - \lambda_0}{h_{i,0}} + \frac{h_{i,1}(\lambda_{i,1} - \lambda_0) + h_{i,0}(\lambda_{i,1} - \lambda_{i,2})}{h_{i,1}(h_{i,0} + h_{i,1})} \right] - \lambda_0^2 \rho g_i = 0. \tag{13}$$

Here, $g_i$ is a second order approximation of the tangential component of gravity evaluated at the quadruple point for Plateau border $i$, computed as[4]

$$g_i = g\hat{\mathbf{g}} \cdot \left( \frac{x_{i,1} - x_0}{h_{i,0}} + \frac{h_{i,1}(x_{i,1} - x_{i,0}) + h_{i,0}(x_{i,1} - x_{i,2})}{h_{i,1}(h_{i,0} + h_{i,1})} \right).$$

Equation (13) reduces to a cubic equation, as follows. We seek a physical solution and so may assume that $\lambda_0 > 0$. Letting $z := \sqrt{\lambda_0}$, the summation in (13) leads to the equation

$$C(a - bz^2) - dz^3 = 0, \tag{14}$$

where $C = \tfrac{1}{2}(\sqrt{3} - \tfrac{\pi}{2})^{\frac{1}{2}} \sigma$, $d = \sum_{i=1}^{4} \rho g_i$, and

$$a = \sum_{i=1}^{4} \frac{\lambda_{i,1}}{h_{i,0}} + \frac{h_{i,1}\lambda_{i,1} + h_{i,0}(\lambda_{i,1} - \lambda_{i,2})}{h_{i,1}(h_{i,0} + h_{i,1})}, \qquad b = \sum_{i=1}^{4} \frac{1}{h_{i,0}} + \frac{1}{h_{i,0} + h_{i,1}}.$$

If $d = 0$, there is exactly one positive solution of (14). If $d \neq 0$, it can be shown that if $h_{i,j} = \mathcal{O}(h)$ and $h$ is sufficiently small, then there is exactly one positive solution of the same order as $\lambda_{i,1}^{1/2}$. In practice, it was observed that an appropriate solution of the cubic equation could always be found, after which the sought after solution to the quadruple point boundary condition is simply $\lambda_0 = z^2$.

Equipped with this value of $\lambda_0$, flux values for each Plateau border can be calculated as the summands in (13):

$$W_i := \tfrac{1}{2}(\sqrt{3} - \tfrac{\pi}{2})^{\frac{1}{2}} \sigma \lambda_0^{\frac{1}{2}} \left[ \frac{\lambda_{i,1} - \lambda_0}{h_{i,0}} + \frac{h_{i,1}(\lambda_{i,1} - \lambda_0) + h_{i,0}(\lambda_{i,1} - \lambda_{i,2})}{h_{i,1}(h_{i,0} + h_{i,1})} \right] - \lambda_0^2 \rho g_i. \tag{15}$$

These values are used in the finite element method for advancing in time each Plateau border discussed above, and since they satisfy $\sum_i W_i = 0$, it follows that the (discrete) mass in the network of Plateau borders is conserved (ignoring the source term $S$). After one time step of the finite element method for each Plateau border, four not necessarily equal thickness values $\tilde{\lambda}_{i,0}^{n+1}$ are obtained at the quadruple point at time step $n + 1$. They are replaced via a simple projection:

---

[4] The equation for $g_i$ may be viewed as calculating $\hat{\mathbf{g}} \cdot \tau = \hat{\mathbf{g}} \cdot \frac{d}{d\ell}\mathbf{x}(\ell)$ where $\ell$ is the arc length and $\mathbf{x}(\ell)$ parameterizes the Plateau border.
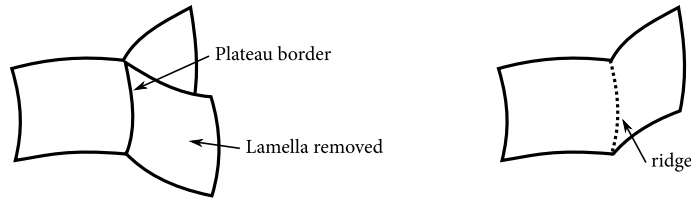
Fig. 7. (Left) Three lamellae meet at a Plateau border. (Right) After removal of the indicated lamella, the associated Plateau border ceases to exist as well and what remains is a sharp ridge that will quickly smooth due to macroscopic effects of surface tension.

$$\lambda_{i,0}^{n+1} = \frac{\sum_{j=1}^{4} \tilde{\lambda}_{j,0}^{n+1} h_{j,0}}{\sum_{j=1}^{4} h_{j,0}}.$$

This projection conserves mass since $\sum_i \lambda_{i,0}^{n+1} h_{i,0} = \sum_i \tilde{\lambda}_{i,0}^{n+1} h_{i,0}$.

**Flux boundary condition**

The flux boundary condition is calculated at the beginning of each time step, and used as a boundary condition for the lamella thin-film equation, and as a source term for the Plateau border thin-film equation. Consider a specific Plateau border, and let $i = 1, 2, 3$ index the three lamellae connected to the Plateau border. At a mesh vertex $x_j$ belonging to the Plateau border, let $\eta_{i,j}$ and $\lambda_j$ be the value of the thickness functions at that particular mesh vertex. Given these film thicknesses, a per-lamella flux value is calculated with (see [44])

$$Q_{i,j} = \tfrac{1}{2}(\sqrt{3} - \tfrac{\pi}{2})^{\frac{3}{4}} \frac{\sigma \eta_{i,j}^{5/2}}{\mu \lambda_j^{3/4}}. \tag{16}$$

The sum of the three flux values defines the source term for the Plateau border:

$$S_j = \sum_{i=1}^{3} Q_{i,j}. \tag{17}$$

Once calculated, the three flux values $Q_{i,j}$, $i = 1, 2, 3$, and the source term $S_j$ are used in the finite element methods above.

### 3.4. Coupling drainage and rearrangement via rupture

In the drainage phase, the coupled system of thin-film equations are evolved until the minimum lamellae thickness falls below a critical threshold $\eta < \eta_{\text{threshold}}$. Rupture is then modelled as a three-step process that couples the results of the drainage phase with reinitializing the rearrangement phase. In the first step, the membrane which had minimal thickness is removed and the liquid mass it contained is uniformly distributed around the Plateau borders to which it was once connected. However, once it is removed, the associated Plateau borders are no longer the intersection of three lamellae and instead become ridges in adjacent lamellae, as shown in Fig. 7. Hence, in the second step, the liquid mass in these Plateau borders is locally "injected" into the adjacent lamellae, and this is accomplished in the same process as reinitializing the rearrangement phase. In more detail:

(i) In the first step, liquid mass in the ruptured lamella is temporarily reassigned to the Plateau borders. This is accomplished with a simple update of the appropriate Plateau border thickness functions: let $\Gamma_L$ denote the lamella which has ruptured, and let $\{\Gamma_{\text{PB}_i}\}_{i=1}^N$ denote the set of Plateau borders (with thickness functions $\lambda_i$) that were on the boundary of $\Gamma_L$. An adjustment factor $\delta$ is computed with the formula

$$\delta = \frac{2 \int_{\Gamma_L} \eta}{\sum_{i=1}^{N} \int_{\Gamma_{\text{PB}_i}} 1}$$

and is used to update $\lambda_i \leftarrow \lambda_i + \delta$. This update conserves liquid mass.

(ii) In the second step, the Lagrangian particle scheme described in §3.2.2 is reinitialized with the updated thickness functions $\eta$ and $\lambda$ obtained from the final step of the drainage phase. This uses an identical procedure to that used to re-seed the Lagrangian particle scheme: the mesh elements used in the finite element formulation are used as locations to seed $\{\eta_i\}$ and $\{\lambda_i\}$ particles with thickness values obtained from the finite element solver. Such particles are generated for all lamellae and Plateau borders which have not been removed by the rupture event. Fictitious $\{\lambda_i\}$ particles are also generated for the removed Plateau borders $\{\Gamma_{\text{PB}_i}\}$; however these are immediately injected onto the adjacent lamellae. This is done by imagining each of these particles to be a "ball of liquid" of finite radius, and collapsing this ball onto the adjacent lamellae in such a way that mass is conserved. In this work, these balls were chosen to

have radius $h$ where $h$ is the grid cell size, such that they have a radially symmetric density that decays to zero at the boundary of the ball.

(iii) Finally, to complete the coupling between the drainage phase and rearrangement phase, the VIIM is notified of the removal of a lamella by merging the two bubbles on either side of the lamella. This is done with a simple update of the indicator function $\chi$ used in the VIIM, which re-identifies the two bubbles on either side of the lamella as now being the same, now larger, bubble. The rearrangement phase can now be re-executed with new system of interconnected interfaces and film thicknesses.

### 3.5. Parallelization

#### 3.5.1. Macroscopic evolution

The macroscopic incompressible Navier–Stokes solver has been parallelized using a simple domain decomposition approach, in which the background rectangular Cartesian grid is subdivided into smaller grids that are assigned to individual processors in an MPI implementation. Synchronization of grid-based data on the subdomains is performed using ghost layers of sufficient size, while the Lagrangian particles are assigned ownership to individual processors according to the subdomain in which they are located, and their ownership is transferred whenever particles cross processor boundaries. The Crank–Nicolson step in the Navier–Stokes solver has been implemented with a simple parallelized Conjugate Gradient algorithm, while the pressure Poisson problem (for the projection step) is solved with a parallelized multigrid algorithm. Our parallelization approach scales well up to thousands of processors. Further comments about parallelizing the VIIM are given in [43].

#### 3.5.2. Thin film drainage

To parallelize the finite element code, the inherent domain decomposition in the problem has been utilized: each lamella and Plateau border is assigned in its entirety to individual processors in an MPI implementation. To do this, the cost of solving the symmetric positive definite systems is estimated for each lamella and Plateau border (via a simple function of the number of mesh elements), and based on this, a simple load balancing algorithm is designed so that each processor had approximately the same amount of work to do, when possible. (For example, one processor might be responsible for a single, large lamella, while another processor may have a few small lamellae and several Plateau borders.) This greatly simplifies parallelization since the associated matrix equation solvers do not need to be parallelized. Synchronization among processors is only necessary for mesh vertices shared by multiple Plateau borders and/or lamellae, and is used to determine flux boundary conditions and quadruple point boundary conditions, and can be performed with MPI's global gather operations. Overall, the scaling efficiency of this approach depends heavily on the number and size of lamellae and Plateau borders, and how they are distributed to the processors. In the work presented here, this approach gave good parallel efficiency in most cases. Efficiency is less when the drainage phase is applied to relatively few bubbles, in which case many of the processors are idle. Usually, however, the computational cost of the macroscopic rearrangement phase dominates the cost of the drainage phase.

## 4. Convergence tests

Next, we test convergence of the various components in our multi-scale framework.

### 4.1. Convergence of multiphase incompressible Navier–Stokes algorithm

The Voronoi Implicit Interface Method and its coupling to a second order projection method for macroscopic incompressible Navier–Stokes flow have been tested extensively in [43], and we refer the reader there.

### 4.2. Convergence of Lagrangian schemes for conservative transport of membrane liquid

We have tested the numerical accuracy of the Lagrangian particle-based scheme for conservatively advecting fluid in the lamellae and Plateau borders, and confirmed that the presented algorithms are first order accurate in time and second order accurate in space. Details are omitted; see [45].

### 4.3. Convergence of thin-film drainage in membranes

Next, we discuss convergence of the thin-film drainage algorithms for the lamellae and Plateau borders.

#### 4.3.1. Lamella

Two tests are performed: one in which an exact solution is manufactured, and another in which the solution is unknown and grid convergence is used. In both cases, we use a catenoid-shaped lamella, which is a surface of revolution with zero mean curvature but nontrivial Gaussian curvature, and thus $k_1^2 + k_2^2$ varies in space. The surface is parameterized by $u \in [0, 2\pi)$ and $v \in [-\frac{5}{4}, \frac{5}{4}]$ such that $x = \cosh v \cos u$, $y = \cosh v \sin u$, $z = v$, and is shown in Fig. 8.
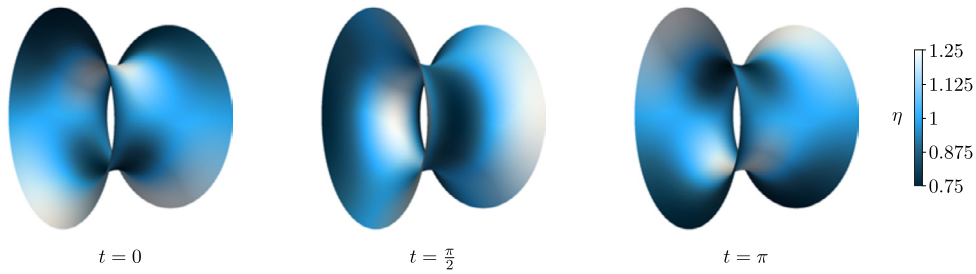
**Fig. 8.** Time evolution of the exact solution for the lamella thin-film equation convergence test on a catenoid.

**Table 1**
Results for the lamella convergence test using a known solution.

| $n$ | $h$ | $\Delta t \propto h$ | | | | $\Delta t \propto h^2$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $e_\infty$ | Order | $e_2$ | Order | $e_\infty$ | Order | $e_2$ | Order |
| 8 | 0.843 | 0.15422 | – | 0.27302 | – | 0.15422 | – | 0.27302 | – |
| 16 | 0.445 | 0.06780 | 1.3 | 0.11975 | 1.3 | 0.04255 | 2.0 | 0.07190 | 2.1 |
| 32 | 0.229 | 0.03631 | 0.9 | 0.06287 | 1.0 | 0.01139 | 2.0 | 0.01895 | 2.0 |
| 64 | 0.117 | 0.01952 | 0.9 | 0.03263 | 1.0 | 0.00284 | 2.0 | 0.00482 | 2.0 |
| 128 | 0.059 | 0.01013 | 1.0 | 0.01663 | 1.0 | 0.00071 | 2.0 | 0.00121 | 2.0 |

In the first test, we create an exact solution of the lamella thin-film equation by choosing $\eta = \eta(u, v, t) = 1 + \frac{1}{4} \sin(u + t) \cos(4\pi v/5)$ and calculating $f$ such that

$$\eta_t + \frac{1}{3\mu} \nabla_s \cdot \left( \sigma \eta^3 \nabla_s \big( (k_1^2 + k_2^2)\eta + \Delta_s \eta \big) + \rho g \hat{\mathbf{g}}_s \, \eta^3 \right) = f.$$

This is supplemented by the implied flux boundary condition $\mathbf{Q} \cdot \nu = Q$ that the exact solution satisfies. A closed-form expression for $f$ and $Q$ is most easily obtained by using computer algebra software, together with the curvilinear coordinate expressions for the surface Laplacian, etc., in the $(u, v)$ coordinate system. The result is too involved to repeat here, but poses no problem in evaluating numerically. In the following, $\hat{\mathbf{g}} = \hat{z}$, and the parameters were chosen such that $\mu = 1$, $\sigma = 0.2$ and $\rho g = 0.5$. Altogether, this choice of exact solution $\eta$ and parameters lead to a solution that is relatively smooth and has competing effects of gravity and diffusion. (Different values of parameters, solutions, and geometries were also tested, and results similar to the following were obtained.) The time evolution of this exact solution is shown in Fig. 8.

For the convergence test, we first generate a finite element mesh by dividing the interval $v \in [-\frac{5}{4}, \frac{5}{4}]$ into $n$ equal sections and $u \in [0, 2\pi]$ into $2n$ equal sections, and cut each resulting quadrilateral into two triangles. While the curvature term $k_1^2 + k_2^2$ can be shown to equal $2 \operatorname{sech}^2 v$, we instead calculate this term and the gravitational term $\hat{\mathbf{g}}_s$ by using a level set function that implicitly defines the catenoid geometry, together with the procedure outlined earlier in §3.3.2. Thus, in the following, we are testing the combined effects of the finite element method in computing $\eta$ and the finite difference method in computing $k_1^2 + k_2^2$ and $\hat{\mathbf{g}}_s$. The error in the computed solution $\eta_h$ is measured with

$$e_p = \max_{0 \leq t \leq \pi} \|\eta_h - \eta\|_{L^p}$$

for $p = 2$ and $p = \infty$. Two time stepping schemes are considered: (i) $\Delta t = \pi/4n = \mathcal{O}(h)$ and (ii) $\Delta t = 2\pi/n^2 = \mathcal{O}(h^2)$. The results are shown in Table 1 and agree with our expectation that the finite element scheme is first order in time and second order in space.

In the second test, the lamella thin-film equation is solved on the same catenoid, with $Q \equiv 0$, $f \equiv 0$, and the initial condition $\eta(t = 0) \equiv 1$. In this example, the direction of gravity is chosen to be $\hat{\mathbf{g}} = \hat{y}$, and all other parameters identical to the previous test. The resulting evolution of film thickness is shown in Fig. 9 (computed using $n = 128$), and shows that the liquid drains in the direction of gravity, but also collects in regions of high curvature. Since the solution is unknown, grid refinement is used to measure convergence: with the difference between solutions on two different grid sizes defined as

$$d_p = \max_{0 \leq t < 2} \|\eta_{2h} - \eta_h\|_{L^p},$$

the convergence rate can be estimated with ratios of $d_p$. For a time step $\Delta t = 4/n^2 = \mathcal{O}(h^2)$, the results are given in Table 2 and show second order convergence.

### 4.3.2. Plateau borders

Next, we test convergence of the scheme for the Plateau border drainage through a system of four Plateau borders meeting at two quadruple points (as shown in Fig. 10) with the aim of testing all aspects of the scheme, including coupling
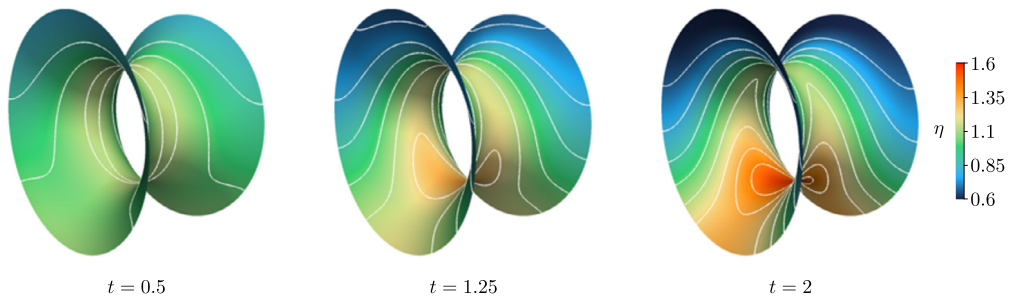
**Fig. 9.** Evolution of film thickness on a catenoid with initially uniform thickness $\eta \equiv 1$ at time $t = 0$ (gravity points directly down). The white lines are contour lines of the thickness function $\eta$.

**Table 2**
Corresponding to Fig. 9, results for the lamella convergence test with unknown solution, using grid refinement.

| $n$ | $h$ | $d_\infty$ | Order | $d_2$ | Order |
|-----|-----|------------|-------|-------|-------|
| 16 | 0.445 | 0.06782 | – | 0.07931 | – |
| 32 | 0.229 | 0.01344 | 2.5 | 0.02018 | 2.1 |
| 64 | 0.117 | 0.00434 | 1.7 | 0.00665 | 1.7 |
| 128 | 0.059 | 0.00107 | 2.1 | 0.00161 | 2.1 |
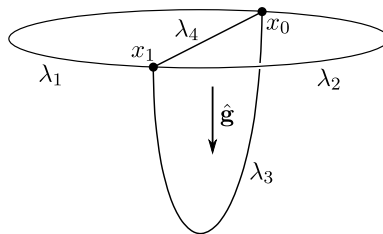


**Fig. 10.** Geometry of the Plateau border convergence test, consisting of four Plateau borders meeting at two quadruple points $x_0$ and $x_1$.

of solutions via the quadruple point boundary condition as well as surface tension and gravity. For each Plateau border we have the evolution equations

$$\partial_t \lambda_i + \frac{C_\Delta}{\mu} \frac{\partial}{\partial \ell} \left( -\frac{1}{2} (\sqrt{3} - \frac{\pi}{2})^{\frac{1}{2}} \sigma \lambda_i^{\frac{1}{2}} \partial_\ell \lambda_i + \lambda_i^2 \rho g \hat{g}_{\tau_i} \right) = S_i, \quad i = 1, 2, 3, 4, \tag{18}$$

coupled via the quadruple point boundary condition $\sum_{i=1}^{4} \frac{1}{2} (\sqrt{3} - \frac{\pi}{2})^{\frac{1}{2}} \sigma \lambda_i^{\frac{1}{2}} \partial_\ell \lambda_i - \lambda_i^2 \rho g \hat{g}_{\tau_i} = 0$ and $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4$ at the two quadruple points.

Two different tests are considered: one in which an exact solution is constructed, and another in which the solution is unknown and grid convergence is used. For the first case, an exact solution of this system is designed by substituting known expressions for $\lambda_1$, $\lambda_2$, and $\lambda_3$ into their respective evolution equations, thereby determining $S_1$, $S_2$, and $S_3$. The quadruple point boundary condition is then used to find a cubic polynomial in $\ell$ for $\lambda_4$ that has the correct value and derivative at each quadruple point, and this in turn generates $S_4$. Using the expressions for $S_i$, the numerical method is used to solve the system of PDEs, with the aim of recovering the exact solutions.

In more detail, with reference to Fig. 10, Plateau borders number 1, 2, and 3 are circular arcs of arc length $\pi$ while Plateau border number 4 is a straight line of arc length 2. The parameterization is such that $\ell = 0$ at quadruple point $x_0$, and $\ell = \pi$ or $\ell = 2$ at quadruple point $x_1$. Gravity points from the middle of Plateau border 4 to the middle of Plateau border 3. Let

$$\lambda_0(\ell, t) = 2 + \cos(3\ell + t), \quad \lambda_1(\ell, t) = 2 + \cos(3\ell + t) + 3\sin\ell, \quad \lambda_2(\ell, t) = 2 + \cos(3\ell + t) + \frac{1}{2}\sin 2\ell.$$

These were chosen to be nontrivial smooth solutions that are strictly positive for all time $t$. Using the quadruple point boundary condition, we can solve for both $\lambda_3(\ell, t)$ and $\lambda_3'(\ell, t)$ at $\ell = 0$ and $\ell = 2$. For each point in time, these four values uniquely determine a cubic polynomial in $\ell$, which is chosen to be the remaining solution $\lambda_3(\ell, t)$. This yields
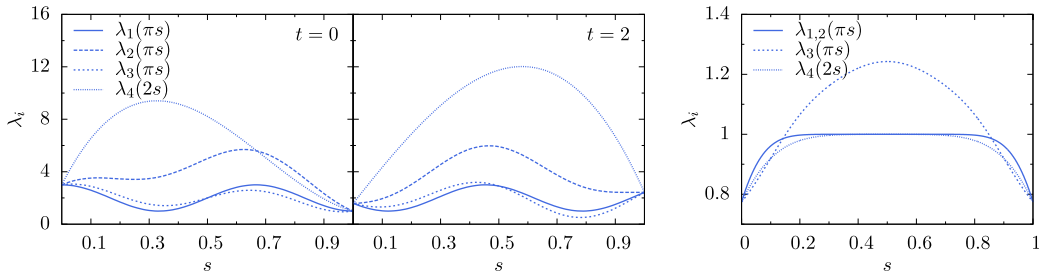
**Fig. 11.** (Left pair) Exact solution for the Plateau border convergence test at times $t = 0$ and $t = 2$. (Right) Numerical solution at time $t = 10$ for the case when $S_i \equiv 0$ and $\lambda_i(t = 0) \equiv 1$ for all $i$.

**Table 3**
Results for the Plateau border convergence test using a known solution.

| $n$ | $\Delta t \propto h$ | | | | $\Delta t \propto h^2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $e_\infty$ | Order | $e_2$ | Order | $e_\infty$ | Order | $e_2$ | Order |
| 32 | 0.06374 | – | 0.05624 | – | 0.06374 | – | 0.05624 | – |
| 64 | 0.03212 | 1.0 | 0.02846 | 1.0 | 0.01597 | 2.0 | 0.01415 | 2.0 |
| 128 | 0.01612 | 1.0 | 0.01433 | 1.0 | 0.00400 | 2.0 | 0.00354 | 2.0 |
| 256 | 0.00808 | 1.0 | 0.00720 | 1.0 | 0.00100 | 2.0 | 0.00089 | 2.0 |
| 512 | 0.00404 | 1.0 | 0.00361 | 1.0 | 0.00025 | 2.0 | 0.00022 | 2.0 |

$$\lambda_3(\ell, t) = 2 + \cos t + x \frac{f_1(t)}{C_1 \sqrt{2 + \cos t}} - \frac{1}{2} x^2 \left( 3 \cos t - \frac{f_2(t)}{C_1 \sqrt{2 - \cos t}} + \frac{2 f_1(t)}{C_1 \sqrt{2 + \cos t}} \right)$$
$$+ \frac{1}{4} x^3 \left( 2 \cos t - \frac{f_2(t)}{C_1 \sqrt{2 - \cos t}} + \frac{f_1(t)}{C_1 \sqrt{2 + \cos t}} \right)$$

where

$$C_1 = \frac{1}{2} (\sqrt{3} - \frac{\pi}{2})^{\frac{1}{2}} \sigma,$$
$$f_1(t) = \rho g (2 + \cos t)^2 + C_1 \sqrt{2 + \cos t}(9 \sin t - 4),$$
$$f_2(t) = \rho g (2 - \cos t)^2 + C_1 \sqrt{2 - \cos t}(9 \sin t - 2).$$

Substituting the expressions for $\lambda_1, \ldots, \lambda_4$ into (18) we can find closed-form expressions of $S_1, \ldots, S_4$. These are too cumbersome to repeat here, but pose no problem when evaluating numerically. Now we must choose specific values for the physical parameters, and we set $\mu = \sigma = \rho = g = 1$. These values were chosen to (i) make surface tension and gravity equally important effects for the purposes of convergence tests, and (ii) guarantees that $\lambda_4$ is strictly positive for all time $t$. Fig. 11 (left pair) illustrates the exact solutions $\lambda_1, \ldots, \lambda_4$ at a two example points in time.

For the numerical method, each Plateau border is discretized with $n$ equal sized line segments, so that $h \approx \pi/n$ for Plateau borders 1, 2, and 3, and $h = 2/n$ for Plateau border 4. The system of PDEs and the quadruple point boundary conditions are solved using the methods described above. The error of the numerical solution is defined as

$$e_p = \max_{i=1,2,3,4} \max_{t \in [0,1]} \|\lambda_i^h - \lambda_i\|_{L^p},$$

i.e., the maximum error over all (discrete) points in time of all Plateau borders, measured in the $L^p$ norm for $p = 2$ and $p = \infty$, where $\lambda_i^h$ denotes the numerical solution. To determine the order of accuracy, two time stepping schemes are considered: (i) one in which $\Delta t = 1/n \propto h$; and (ii) $\Delta t = 32/n^2 \propto h^2$. The convergence results are shown in Table 3, and confirms our expectation that the scheme is first order accurate in time and second order accurate in space.

For the second convergence test, the same Plateau border geometry and numerical discretization is used, and the case when $S_i \equiv 0$ for all time is investigated. In particular, the Plateau borders are initialized with a uniform thickness so that $\lambda_i \equiv 1$ for all $i$ at time $t = 0$. In this case, the system conserves mass, i.e. $\frac{d}{dt} \sum_i \int_{\Gamma_i} \lambda_i = 0$. Since the exact solution is unknown, grid refinement is used to study convergence by defining

$$d_p(h) = \max_{i=1,2,3,4} \max_{t \in [0,10]} \|\lambda_i^{2h} - \lambda_i^h\|_{L^p},$$

and then estimating the convergence rate by using ratios of $d_p$ with $\log_2 \big( d(2h)/d(h) \big)$. As before, two time stepping schemes are considered: (i) $\Delta t = 5/n \propto h$ and (ii) $\Delta t = 320/n^2 \propto h^2$. The results are shown in Table 4, and a plot of the solution at time $t = 10$ is shown in Fig. 11 (right). The results indicate first order accuracy in time and second order in space. It was also confirmed that mass of the discrete solution was conserved, independent of the grid size.

**Table 4**
Results for the Plateau border convergence test with unknown solution, using grid refinement.

| $n$ | $\Delta t \propto h$ | | | | $\Delta t \propto h^2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $d_\infty$ | Order | $d_2$ | Order | $d_\infty$ | Order | $d_2$ | Order |
| 64 | 0.01136 | – | 0.00308 | – | 0.01877 | – | 0.00482 | – |
| 128 | 0.00569 | 1.0 | 0.00119 | 1.4 | 0.00697 | 1.4 | 0.00141 | 1.8 |
| 256 | 0.00170 | 1.7 | 0.00038 | 1.6 | 0.00121 | 2.5 | 0.00031 | 2.2 |
| 512 | 0.00062 | 1.4 | 0.00016 | 1.2 | 0.00027 | 2.2 | 0.00008 | 2.0 |
| 1024 | 0.00028 | 1.1 | 0.00008 | 1.0 | 0.00007 | 2.0 | 0.00002 | 2.0 |

**Table 5**
Physical parameters used in the numerical simulations.

| Parameter and units | Notation | Fig. 12 | Fig. 13 | Fig. 15 | Fig. 16 | Fig. 17[a] | Fig. 18 |
|---|---|---|---|---|---|---|---|
| Surface tension ($kg\,s^{-2}$) | $\sigma$ | 0.034 | 0.034 | 0.034 | 0.034 | 0.034 | 0.034 |
| Gas density ($kg\,m^{-3}$) | $\rho_g$ | 1.15 | 1.15 | – | 1.15 | 1.15 | 1.15 |
| Gas viscosity ($kg\,m^{-1}\,s^{-1}$) | $\mu_g$ | $1.8 \times 10^{-5}$ | $3.42 \times 10^{-5}$ | – | $1.8 \times 10^{-5}$ | $5 \times 10^{-5}$ | $5 \times 10^{-5}$ |
| Liquid density ($kg\,m^{-3}$) | $\rho$ | – | – | 1000 | 1000 | 1000 | 1000 |
| Liquid viscosity ($kg\,m^{-1}\,s^{-1}$) | $\mu$ | – | – | 0.001 | 0.001 | 0.001 | 0.001 |
| Gravity ($m\,s^{-2}$) | $g$ | – | – | 9.8 | 9.8 | 9.8 | 9.8 |
| Initial lamellae thickness (µm) | $\eta\ (t=0)$ | – | – | 5 | 5 | 10 | 1 |
| Initial Plateau border ($mm^2$) cross-sectional area | $\lambda\ (t=0)$ | – | – | 0.05 | 0.05 | 0.002 | 0.05 |
| Typical bubble diameter (mm) | | 10 | 40 | 2 | 3 | 0.1–0.4 | 3 |
| Rupture threshold (nm) | | – | – | – | 10 | 40 | 10 |

[a] For the purposes of limiting bubble rearrangement to the lamellae cluster, in the simulation of Fig. 17, rupture of the background membrane was inhibited by including an additional diffusion term of the form $\Delta_s^2 \eta$ into its thin-film equation.

## 5. Results

In the previous set of sections, several numerical schemes have been developed for use in the rearrangement, drainage, and rupture phases of the multiscale model of foam dynamics. For each of these phases, the corresponding methods have been designed to accurately solve the underlying evolution equations, with the ability to couple to the other phases. Here, several results of the framework are presented. In the first set of results, individual components of the foam model are tested and verified, demonstrating various physical mechanisms of the model. The entire system is then coupled to study two problems exhibiting nontrivial foam dynamics that involve foam collapse via bubble rupture cascades. We note that some of these figures first appeared in [44].

In all of these results, the physical parameters have been chosen to represent a typical soap bubble foam: the density and viscosity of the liquid is similar to that of water, gravity is normal Earth gravity, and the gas phase represents typical ambient air at room temperature. The precise values of the physical parameters used in the following results are given in Table 5.

### 5.1. Rearrangement phase

To demonstrate rearrangement and surface area minimization, Fig. 12 shows the effect of removing a specific lamella from a cluster which is otherwise in macroscopic equilibrium. After removal, surface tension drives the cluster into a new configuration, undergoing various topological changes in the process. Fig. 12 (top) plots the total surface area as a function of time and shows that it reaches a local minimum; the bottom figures illustrate how the "hole" made by removing the lamella is filled-in, generating capillary waves as it does so, with 120° angle conditions satisfied throughout the process, ultimately leading to an equilibrium where each lamella has constant mean curvature.

To test the accuracy of the Navier–Stokes solver, in Fig. 13, numerical results are compared to that of a bubble oscillation experiment by Kornek et al. [27]. In that work, several high-speed movies were captured of bubbles colliding and merging together; once merged, the resulting larger bubbles oscillate due to effects of surface tension. One of these movies was used to determine the radii of two slightly overlapping bubbles, as shown by the $t = 0$ inset in Fig. 13. Experimental parameters quoted in [27] were then used as parameters for the Navier–Stokes solver. In particular, the density of the gas was quoted as 1.2 $kg\,m^{-3}$, however the authors were uncertain as to precisely what percentage of butane was contained in the gas mixture. It was found that better agreement between numerical results and the experiment were obtained by slightly altering the density to 1.15 $kg\,m^{-3}$. Overall, the numerical results illustrated in Fig. 13 show good qualitative agreement with the experiment. It is possible to more carefully analyze the results by measuring modes and frequencies of oscillation, as well as dampening rates, as was done in [27], however this has not been considered here.

In the next example, the effect of rearrangement on changes in film thickness is demonstrated by considering an oscillating soap bubble, as shown in Fig. 14. A bubble with an initial shape of an ellipsoid (semi-principal axes of lengths of $\frac{20}{3}$ mm, 5 mm and 4 mm) is initialized with a uniform lamella thickness of $\eta \equiv 0.3$ µm. Fig. 14 (top) shows the evolution of the
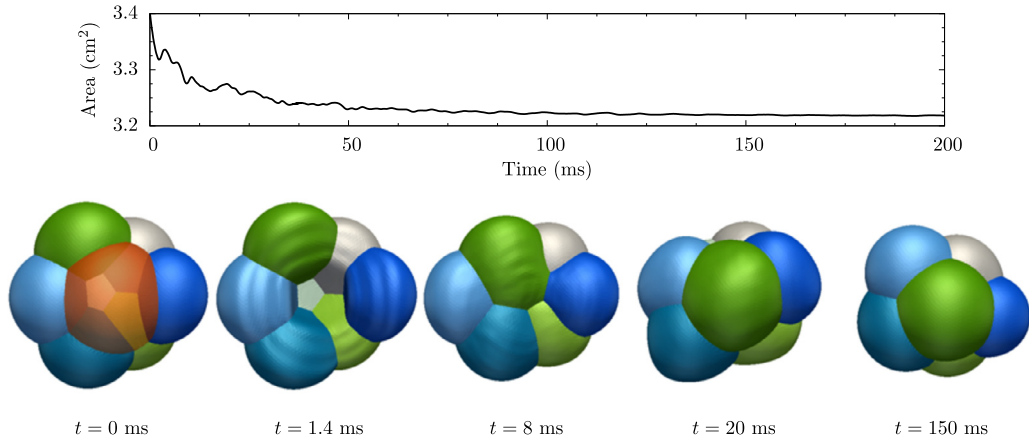
**Fig. 12.** A small cluster, initially in equilibrium, undergoes rearrangement due to the removal of a lamella (orange) at time $t = 0$. (Top) Total surface area as a function of time. (Bottom) Evolution of cluster during rearrangement. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
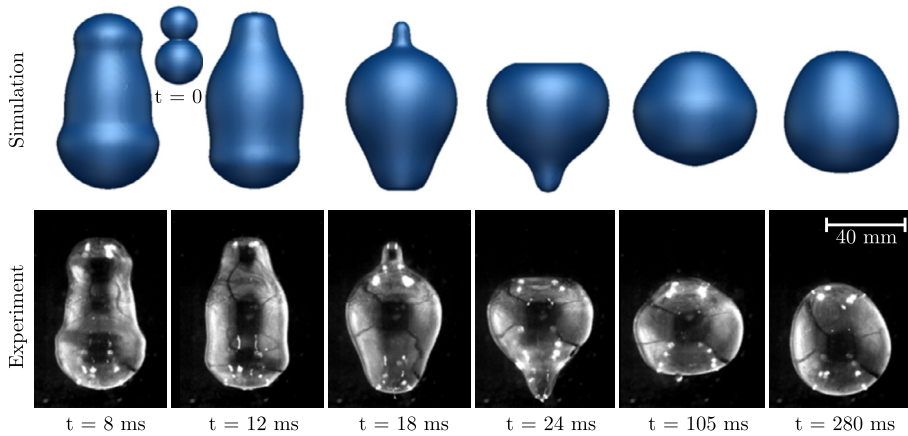


**Fig. 13.** Comparison of numerical results with experiment. Two spherical soap bubbles merge at $t = 0$, subsequently causing surface tension driven oscillations that eventually lead to a larger spherical bubble. Images of experimental results reproduced from [27] (by permission); numerical simulation uses identical physical parameters and time scale, and was computed on a $64 \times 64 \times 96$ grid.
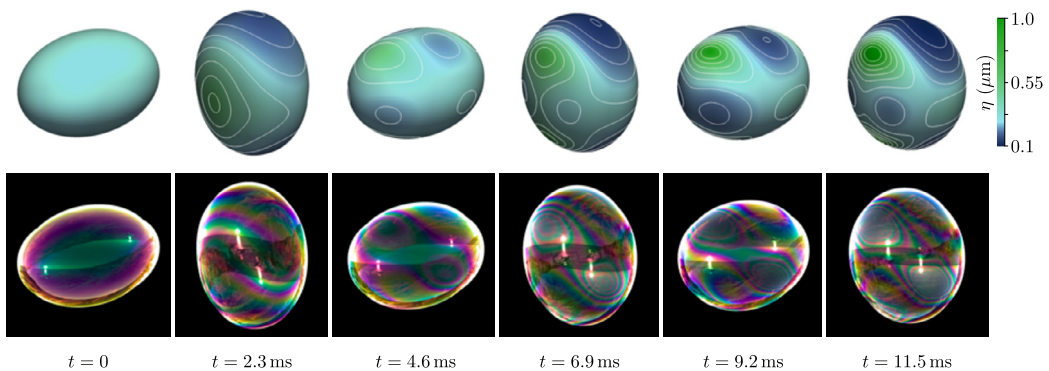


**Fig. 14.** Evolution of lamella thickness for an oscillating bubble. (Top row) Colors indicate thickness $\eta$ of the lamella and the white curves are contour lines of $\eta$. (Bottom row) The same bubble oscillation visualized with thin-film interference. Simulation computed on a $256 \times 256 \times 256$ grid in a cubic domain of side length 20 mm with periodic boundary conditions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
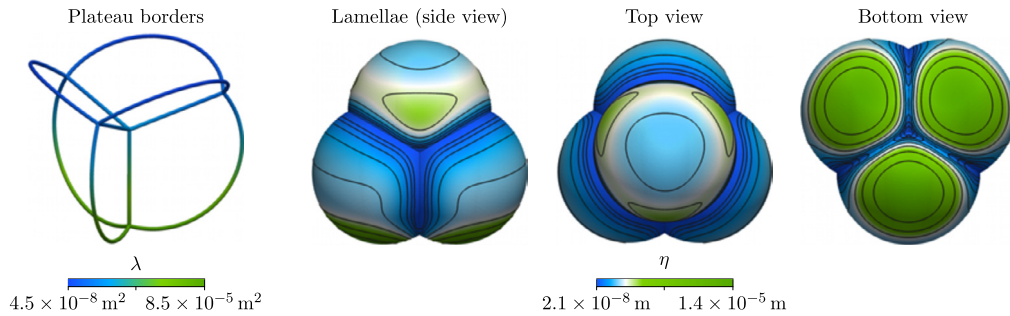
Plateau borders          Lamellae (side view)          Top view          Bottom view

$\lambda$                                               $\eta$

$4.5 \times 10^{-8}\,\mathrm{m}^2$    $8.5 \times 10^{-5}\,\mathrm{m}^2$          $2.1 \times 10^{-8}\,\mathrm{m}$    $1.4 \times 10^{-5}\,\mathrm{m}$

**Fig. 15.** Solution of the coupled lamella and Plateau border thin-film equations on a pyramid of four spherical bubbles. Colors indicate thickness $\eta$ of the lamellae and cross-sectional area $\lambda$ of the Plateau borders, the black curves are contour lines of $\eta$, and (except for the top and bottom views) gravity points down. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

subsequent variations in film thickness: areas where the bubble has "compressed" tend to increase in thickness, while the thickness decreases near points of expansion. In Fig. 14 (bottom), the same results are shown using thin-film interference: constructive and destructive interference of reflected light, together with variations in film thickness, lead to interference patterns ("rainbows") seen in everyday soap films. Here, the physically-based ray tracing rendering engine *LuxRender* [28] has been used to solve the Fresnel equations to obtain reflection, refraction, and thin-film interference effects. In particular, the bubble has been globally illuminated by a beach scene: in the reflected image, one can see two suns (from the front and back surfaces of the bubble), with the sky in the upper half. Comparison of the interference patterns with the film thickness shown in the top row reveals a correlation between the two.

### 5.2. Drainage phase

In this example, liquid drainage in a coupled lamellae and Plateau border system is demonstrated. Fig. 15 shows a pyramid of four spheres with diameters 2 mm, forming a network of six lamellae and ten Plateau borders. The lamellae are initialized at time $t = 0$ with a uniform thickness of $\eta = 5$ μm and the Plateau borders with uniform cross-sectional area $\lambda = 0.05$ mm². Fig. 15 shows the thickness after draining for 16.1 s. The effect of gravity is seen with the accumulation of liquid at the bottom of the lamellae and Plateau borders, while the effect of the flux boundary condition can be observed with the reduced thickness of the lamellae at the junctions. In the case of the Plateau borders, the thickness profile has essentially attained an equilibrium – as the liquid drains to the bottom due to gravity, the Plateau borders become thin at the top, thereby reducing the liquid pressure, which in turn leads to a pressure gradient opposing the force of gravity.

### 5.3. Rupture and redistribution of mass

To demonstrate rupture and redistribution of liquid mass, in Fig. 16, a cluster of bubbles with non-uniform thickness has been draining, and the internal lamella separating the two front facing bubbles ruptures immediately after time $t = 0$. The liquid originally contained in the lamella and the Plateau borders to which it was once connected is locally distributed to the remaining lamellae, as shown by the sudden increase in thickness. The system, driven by macroscopic rearrangement, quickly moves into a new configuration.

### 5.4. Coupled rearrangement, drainage, and rupture

By assembling the complete physical system, the multiscale model can be used to predict the evolution of foam cluster dynamics, under the combined effects of rearrangement, drainage, and rupture. This is demonstrated here with two foam collapse problems.

The first example serves to highlight how foam dynamics can crucially depend on the interaction between the three phases. In other words, while the drainage phase determines which lamellae rupture, both the rupture phase and rearrangement phase contribute significantly to this process as well, through nontrivial transport of membrane liquid. To motivate the design of this example, suppose that the lamellae start with a uniform thickness of $\eta_0$. As seen in the drainage example in Fig. 15, it is often the case that as lamellae drain over time, a boundary layer in their thickness develops near the Plateau borders, due in part to the flux boundary condition. Scaling arguments applied to the lamella thin-film equation together with the flux boundary condition suggest that the width of the boundary layer after a fixed amount of time is $\mathcal{O}(\eta_0^{1/2}\lambda_0^{1/4})$, where $\lambda_0$ is a typical cross-sectional area of the Plateau border. For typical film thicknesses and drainage times, the length predicted by the scaling is on the order of 0.1 mm, and this was confirmed by numerical tests, as is the result that Plateau borders tend to have the same order of magnitude thickness across the entire network. It follows that for a cluster of bubbles which initially have the same lamellae thickness, all lamellae drain at approximately the same rate, and thus those bubbles smaller than the boundary layer will thin more rapidly and rupture first.
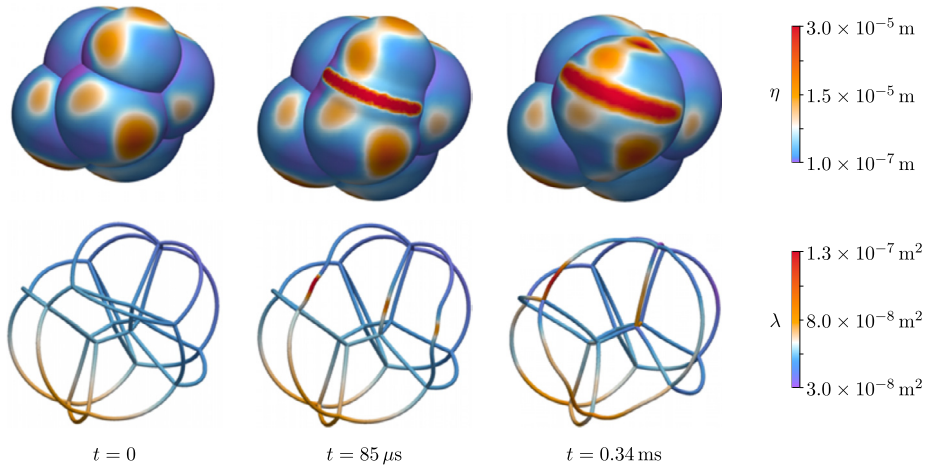
**Fig. 16.** Evolution during rupture. An internal lamella joining the two front facing bubbles ruptures and is removed, leading to rearrangement of bubbles and varying film thicknesses.
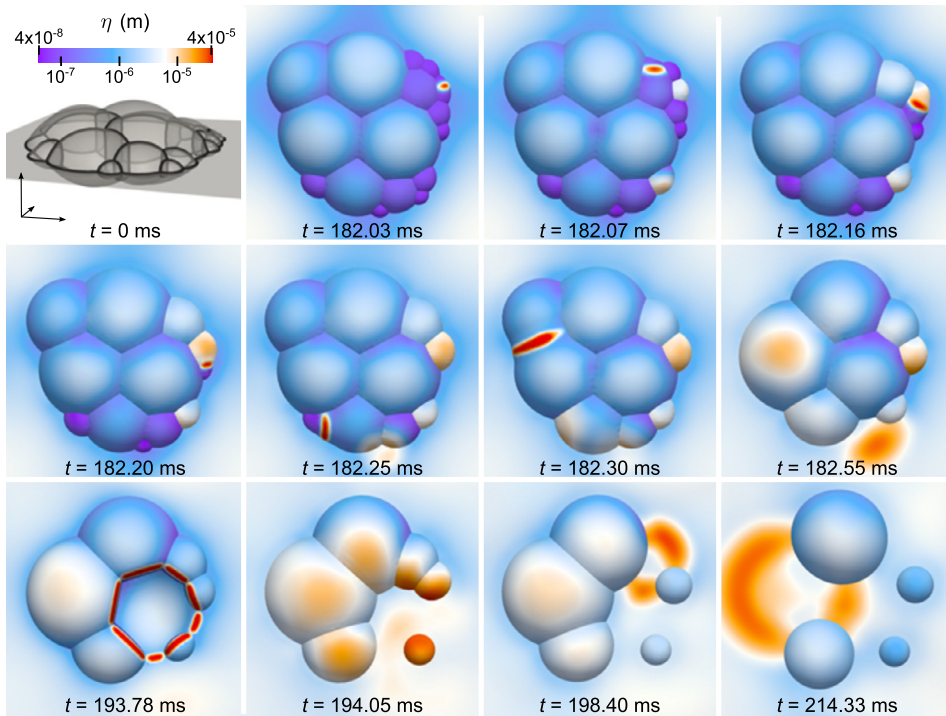


**Fig. 17.** Results of the coupled multiscale model for a cluster of bubbles attached to a membrane. In the top-left frame, a side-view of the initial configuration is shown, using semi-opaque lamellae and emphasizing the Plateau borders, to highlight the 3D structure of the results. In the rest of the frames, a top-down view is given, showing the lamellae film thickness $\eta$, corresponding to the indicated color scale. The background membrane absorbs some of the drainage, but is chosen not to rupture. As the system evolves, rupture events can be identified by the localized increases in lamellae thickness. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To demonstrate this behavior, and how it effects rearrangement of bubbles, an example is shown in Fig. 17. A cluster of 17 bubbles is suspended by a membrane, so that bubbles protrude below and above the membrane. This configuration was designed in order to make the rearrangement simpler to visualize with a top-down perspective. The cluster has a range of bubble sizes, from 0.1 to 0.5 mm in diameter, and at time $t = 0$ is initially in equilibrium, such that each lamella has a uniform thickness of 10 μm, and each Plateau border a uniform cross-sectional area of 0.002 mm$^2$. After draining for a time of 182 ms, some of the smallest lamellae rupture in quick succession. As this occurs, adjacent bubbles grow in size and increase in thickness. Initially, many of the rupture events are associated with the smaller lamellae, but because rupture affects the macroscopic dynamics of the bubbles, larger lamellae can rupture due to membrane stretching. On this small spatial scale, the rearrangement phase typically takes 0.1 ms to equilibrate, while drainage takes tens of milliseconds. The
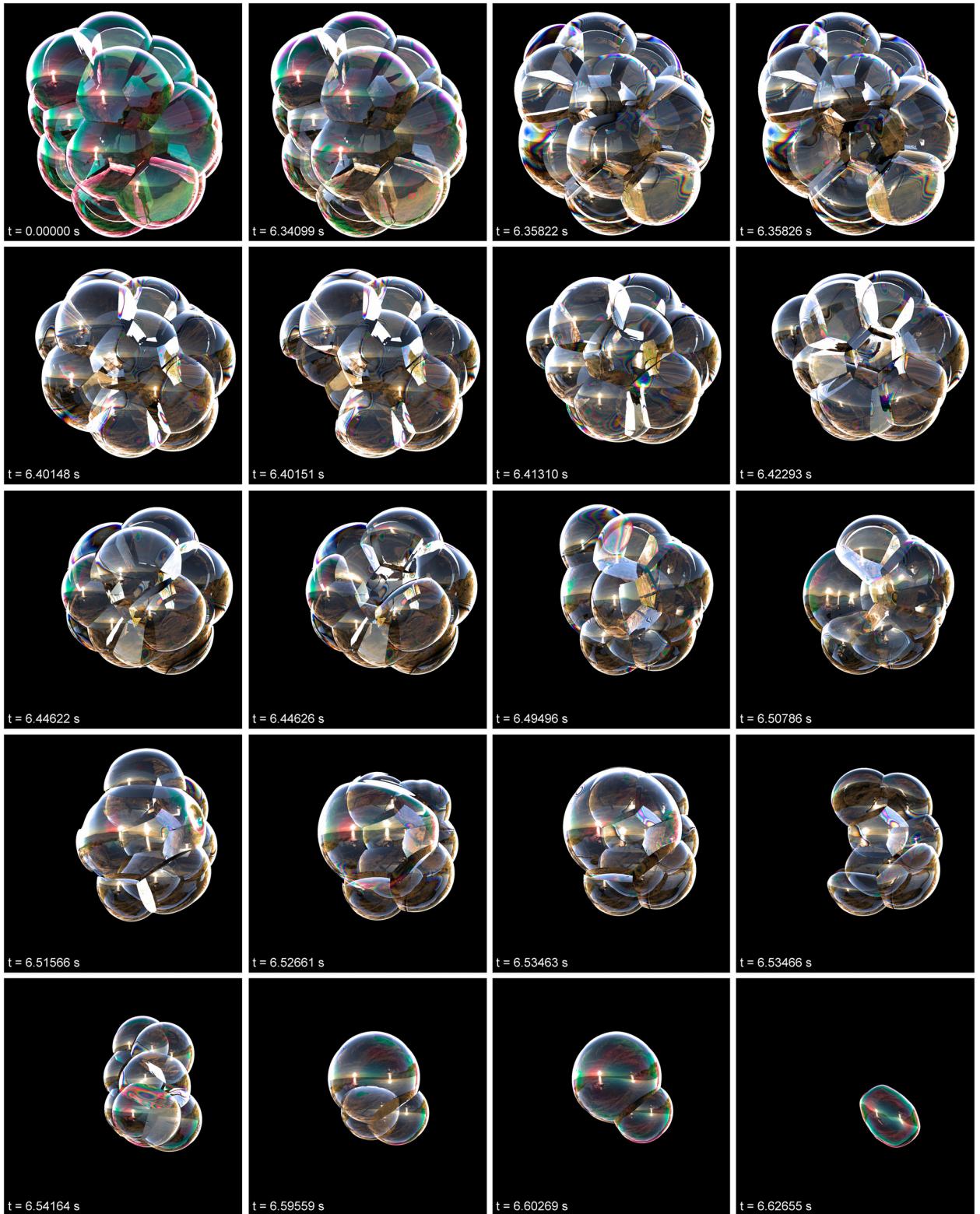
**Fig. 18.** Collapse of a foam cluster, visualized with thin-film interference. See also the supplementary material of [44] for the corresponding movie.

results show how a nontrivial sequence of rupture events is obtained, and how bubble rearrangement affects rupture events, both locally and globally, due to changes in film thickness and macroscopic hydrodynamics.

Finally, in the last example, Fig. 18 shows the results for a larger cluster of 27 bubbles with a typical bubble diameter of 3 mm. In this example, a cluster starts in macroscopic equilibrium with a uniform lamellae thickness of 1 μm, and for each Plateau border a uniform cross-sectional area of 0.05 mm$^2$. Compared to the case in Fig. 17, in this example the typical bubble size is much larger. As a result, the rupture of a single lamella can have a greater impact on the global dynamics. We see this in Fig. 18 – after draining for 6.3 s, a single lamella ruptures, causing a rapid collapse of the entire structure.

## 6. Conclusions

A multiscale model of the dynamics of a foam has been developed, permitting the study of the effects of fluid properties, topology, bubble shape, and distribution, on drainage, rupture, and rearrangement. Several numerical methods were developed to accompany this model, ranging from Lagrangian-based schemes for transporting film thickness during rearrangement, to biharmonic-modified finite element methods and techniques for treating the coupled boundary conditions in the system of thin-film equations in the drainage phase. Using two foam collapse problems, it was demonstrated how rupture, liquid drainage, and gas hydrodynamics each affect one another on both local and global scales.

Both the scale-separated model and the underlying numerical algorithms are general enough to allow extension of the physics at individual scales to include other phenomena. For example, some types of surfactant solution give rise to films with long lifetimes. In such films, disjoining pressures and van der Waals force can be important in liquid drainage and rupture initiation; these additional physics can be modelled by altering the thin-film equations. Diffusive coarsening, which can also be important for long lifetime foams, could be added by generalizing the thin-film equations to allow slow movement of the membranes, using equations similar to those derived in [21]. In other types of surfactant solution, mobile/stress-free boundary conditions at the liquid–gas interface are more appropriate than the no-slip boundary conditions used here [22, 26,25]. In principle, it is possible to derive different thin-film equations on the curved lamellae, taking into account these boundary conditions. In so doing, it is expected that this would lead to coupled PDEs for film thickness evolution: one for the film thickness, which is coupled to a second equation for the tangential velocity field of the liquid inside the membrane. Similar approaches may also allow surface viscosities, evaporation dynamics, and heating of a foam to be modelled.

Additional future work could include more detailed considerations of some of the assumptions made in this model. For example, in the macroscopic rearrangement phase, the films were idealized as massless and infinitely thin, so that the liquid was essentially passively advected by the gas dynamics. Thus, inertial and viscous effects of the liquid inside the films were neglected. Some experimental studies indicate that such effects can dampen the motion of the films. In the work of [27], several experiments were performed which indicated that the dampening rate for soap bubble oscillations is 40%–60% faster in practice, compared to small-scale perturbation theory and numerical experiments. Nevertheless, the frequency of oscillation of the experimental data matches theoretical and numerical predictions. To address the difference in dampening, it may be possible to suitably modify the surface tension force in Navier–Stokes, and/or the local viscosity of the gas, to take into account these inertial effects.

## Acknowledgements

## References

[1] D. Adalsteinsson, J.A. Sethian, A fast level set method for propagating interfaces, J. Comput. Phys. 118 (2) (1995) 269–277, http://dx.doi.org/10.1006/jcph.1995.1098.

[2] David Adalsteinsson, J.A. Sethian, Transport and diffusion of material quantities on propagating interfaces via level set methods, J. Comput. Phys. 185 (1) (2003) 271–288, http://dx.doi.org/10.1016/S0021-9991(02)00057-8.

[3] David Adalsteinsson, James A. Sethian, The fast construction of extension velocities in level set methods, J. Comput. Phys. 148 (1) (1999) 2–22, http://dx.doi.org/10.1006/jcph.1998.6090.

[4] Ann S. Almgren, John B. Bell, Phillip Colella, Louis H. Howell, Michael L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, J. Comput. Phys. 142 (1) (1998) 1–46, http://dx.doi.org/10.1006/jcph.1998.5890.

[5] Jürgen Becker, Günther Grün, Martin Lenz, Martin Rumpf, Numerical methods for fourth order nonlinear degenerate diffusion problems, Appl. Math. 47 (6) (2002) 517–543, http://dx.doi.org/10.1023/B:APOM.0000034537.55985.44.

[6] Marcelo Bertalmio, Li-Tien Cheng, Stanley Osher, Guillermo Sapiro, Variational problems and partial differential equations on implicit surfaces, J. Comput. Phys. 174 (2) (2001) 759–780, http://dx.doi.org/10.1006/jcph.2001.6937.

[7] Andrea L. Bertozzi, Ning Ju, Hsiang-Weir Lu, A biharmonic-modified forward time stepping method for fourth order nonlinear diffusion equations, Discrete Contin. Dyn. Syst. 29 (4) (2011) 1367–1391, http://dx.doi.org/10.3934/dcds.2011.29.1367.

[8] J.C. Bird, R. de Ruiter, L. Courbin, H.A. Stone, Daughter bubble cascades produced by folding of ruptured thin films, Nature 465 (2010) 759–762, http://dx.doi.org/10.1038/nature09069.

[9] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (2) (1992) 335–354, http://dx.doi.org/10.1016/0021-9991(92)90240-Y.

[10] K. Brakke, The Surface Evolver, Exp. Math. 1 (2) (1992) 141–165, http://dx.doi.org/10.1080/10586458.1992.10504253.

[11] C.J.W. Breward, P.D. Howell, The drainage of a foam lamella, J. Fluid Mech. 458 (2002) 379–406, http://dx.doi.org/10.1017/S0022112002007930.

[12] D.L. Chopp, Computing minimal surfaces via level set curvature flow, J. Comput. Phys. 106 (1993) 77–91, http://dx.doi.org/10.1006/jcph.1993.1092.

[13] D.L. Chopp, Some improvements of the Fast Marching Method, SIAM J. Sci. Comput. 23 (1) (2001) 230–244, http://dx.doi.org/10.1137/S106482750037617X.

[14] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (104) (1968) 745–762, http://dx.doi.org/10.1090/S0025-5718-1968-0242392-2.

[15] J. Douglas Jr., T. Dupont, Alternating-direction Galerkin methods on rectangles, in: Numerical Solution of Partial Differential Equations, II (SYNSPADE 1970), Proc. Sympos. Univ. of Maryland, College Park, Md., 1970, Academic Press, New York, 1971, pp. 133–214.

[16] M. Durand, H.A. Stone, Relaxation time of the topological T1 process in a two-dimensional foam, Phys. Rev. Lett. 97 (2006) 226101, http://dx.doi.org/10.1103/PhysRevLett.97.226101.

[17] G. Dziuk, C.M. Elliott, Surface finite elements for parabolic equations, J. Comput. Math. 25 (4) (2007) 385–407.

[18] John B. Greer, Andrea L. Bertozzi, Guillermo Sapiro, Fourth order partial differential equations on general geometries, J. Comput. Phys. 216 (1) (2006) 216–246, http://dx.doi.org/10.1016/j.jcp.2005.11.031.

[19] G. Grün, M. Rumpf, Nonnegativity preserving convergent schemes for the thin film equation, Numer. Math. 87 (2000) 113–152, http://dx.doi.org/10.1007/s002110000197.

[20] S. Hilgenfeldt, A.M. Kraynik, S.A. Koehler, H.A. Stone, An accurate von Neumann's law for three-dimensional foams, Phys. Rev. Lett. 86 (12) (2001) 2685–2688, http://dx.doi.org/10.1103/PhysRevLett.86.2685.

[21] P.D. Howell, Surface-tension-driven flow on a moving curved surface, J. Eng. Math. 45 (2003) 283–308, http://dx.doi.org/10.1023/A:1022685018867.

[22] P.D. Howell, H.A. Stone, On the absence of marginal pinching in thin free films, Eur. J. Appl. Math. 16 (05) (2005) 569–582, http://dx.doi.org/10.1017/S095679250500625X.

[23] Yongsam Kim, Ming-Chih Lai, Charles S. Peskin, Numerical simulations of two-dimensional foam by the Immersed Boundary Method, J. Comput. Phys. 229 (13) (2010) 5194–5207, http://dx.doi.org/10.1016/j.jcp.2010.03.035.

[24] Yongsam Kim, Yunchang Seol, Ming-Chih Lai, Charles S. Peskin, The immersed boundary method for two-dimensional foam with topological changes, Commun. Comput. Phys. 12 (2) (2012) 479–493, http://dx.doi.org/10.4208/cicp.181210.080811s.

[25] S.A. Koehler, S. Hilgenfeldt, H.A. Stone, Liquid flow through aqueous foams: the node-dominated foam drainage equation, Phys. Rev. Lett. 82 (1999) 4232–4235, http://dx.doi.org/10.1103/PhysRevLett.82.4232.

[26] S.A. Koehler, S. Hilgenfeldt, H.A. Stone, Foam drainage on the microscale: I. Modeling flow through single Plateau borders, J. Colloid Interface Sci. 276 (2) (2004) 420–438, http://dx.doi.org/10.1016/j.jcis.2003.12.061.

[27] U. Kornek, F. Müller, K. Harth, A. Hahn, S. Ganesan, L. Tobiska, R. Stannarius, Oscillations of soap bubbles, New J. Phys. 12 (2010) 073031, http://dx.doi.org/10.1088/1367-2630/12/7/073031.

[28] LuxRender, http://www.luxrender.net, 2012.

[29] R.D. MacPherson, D.J. Srolovitz, The von Neumann relation generalized to coarsening of three-dimensional microstructures, Nature 446 (2007) 1053–1055, http://dx.doi.org/10.1038/nature05745.

[30] C.A. Miller, L.E. Scriven, The oscillations of a fluid droplet immersed in another fluid, J. Fluid Mech. 32 (3) (1968) 417–435, http://dx.doi.org/10.1017/S0022112068000832.

[31] W.W. Mullins, Two-dimensional motion of idealized grain boundaries, J. Appl. Phys. 27 (8) (1956) 900–904, http://dx.doi.org/10.1063/1.1722511.

[32] T.G. Myers, Thin films with high surface tension, SIAM Rev. 40 (3) (1998) 441–462, http://dx.doi.org/10.1137/S003614459529284X.

[33] Anh V. Nguyen, Liquid drainage in single Plateau borders of foam, J. Colloid Interface Sci. 249 (2002) 194–199, http://dx.doi.org/10.1006/jcis.2001.8176.

[34] A. Oron, S.H. Davis, S.G. Bankoff, Long-scale evolution of thin liquid films, Rev. Mod. Phys. 69 (1997) 931–980, http://dx.doi.org/10.1103/RevModPhys.69.931.

[35] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49, http://dx.doi.org/10.1016/0021-9991(88)90002-2.

[36] P.-O. Persson, G. Strang, A simple mesh generator in Matlab, SIAM Rev. 46 (2) (2004) 329–345, http://dx.doi.org/10.1137/S0036144503429121.

[37] J. Plateau, Statique expérimentale et théorique des liquides soumis aux seules forces moléculaires, Gauthier-Villars, Trubner et cie, Paris, France, 1873.

[38] Konrad Polthier, Computational aspects of discrete minimal surfaces, in: D. Hoffman (Ed.), Global Theory of Minimal Surfaces, Proceedings of the Clay Mathematics Institute 2001 Summer School, American Mathematical Society, 2002.

[39] Hernán Ritacco, Flavien Kiefer, Dominique Langevin, Lifetime of bubble rafts: cooperativity and avalanches, Phys. Rev. Lett. 98 (2007) 244501, http://dx.doi.org/10.1103/PhysRevLett.98.244501.

[40] R.I. Saye, An algorithm to mesh interconnected surfaces via the Voronoi interface, Eng. Comput. 31 (1) (2013) 123–139, http://dx.doi.org/10.1007/s00366-013-0335-9.

[41] R.I. Saye, High-order methods for computing distances to implicitly defined surfaces, Commun. Appl. Math. Comput. Sci. 9 (1) (2014) 107–141, http://dx.doi.org/10.2140/camcos.2014.9.107.

[42] R.I. Saye, J.A. Sethian, The Voronoi Implicit Interface Method for computing multiphase physics, Proc. Natl. Acad. Sci. 108 (49) (2011) 19498–19503, http://dx.doi.org/10.1073/pnas.1111557108.

[43] R.I. Saye, J.A. Sethian, Analysis and applications of the Voronoi Implicit Interface Method, J. Comput. Phys. 231 (18) (2012) 6051–6085, http://dx.doi.org/10.1016/j.jcp.2012.04.004.

[44] R.I. Saye, J.A. Sethian, Multiscale modeling of membrane rearrangement, drainage, and rupture in evolving foams, Science 340 (6133) (2013) 720–724, http://dx.doi.org/10.1126/science.1230623.

[45] Robert Saye, The Voronoi Implicit Interface Method with applications to multiphase fluid flow and multiscale modelling of foam dynamics, PhD thesis, University of California, Berkeley, 2013.

[46] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Natl. Acad. Sci. 93 (1996) 1591–1595, http://dx.doi.org/10.1073/pnas.93.4.1591.

[47] J.A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences, Cambridge University Press, 1999.

[48] J.A. Sethian, Ying Shan, Solving partial differential equations on irregular domains with moving interfaces, with applications to superconformal electrodeposition in semiconductor manufacturing, J. Comput. Phys. 227 (13) (2008) 6411–6447, http://dx.doi.org/10.1016/j.jcp.2008.03.001.

[49] J.A. Sethian, Peter Smereka, Level set methods for fluid interfaces, Annu. Rev. Fluid Mech. 35 (2003) 341–372, http://dx.doi.org/10.1146/annurev.fluid.35.101101.161105.

[50] H.A. Stone, A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface, Phys. Fluids A 2 (1) (1990) 111–112, http://dx.doi.org/10.1063/1.857686.

[51] Mark Sussman, Ann S. Almgren, John B. Bell, Phillip Colella, Louis H. Howell, Michael L. Welcome, An adaptive level set approach for incompressible two-phase flows, J. Comput. Phys. 148 (1) (1999) 81–124, http://dx.doi.org/10.1006/jcph.1998.6106.
[52] Mark Sussman, Peter Smereka, Stanley Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1) (1994) 146–159, http://dx.doi.org/10.1006/jcph.1994.1155.
[53] N. Vandewalle, J.F. Lentz, Cascades of popping bubbles along air/foam interfaces, Phys. Rev. E 64 (2001) 021507, http://dx.doi.org/10.1103/PhysRevE.64.021507.
[54] N. Vandewalle, J.F. Lentz, S. Dorbolo, F. Brisbois, Avalanches of popping bubbles in collapsing foams, Phys. Rev. Lett. 86 (1) (2001), http://dx.doi.org/10.1103/PhysRevLett.86.179.
[55] J. von Neumann, in: C. Herring (Ed.), Metal Interfaces, American Society for Metals, Cleveland, 1952, pp. 108–110.
[56] Zebin Wang, Ganesan Narsimhan, Model for Plateau border drainage of power-law fluid with mobile interface and its application to foam drainage, J. Colloid Interface Sci. 300 (2006) 327–337, http://dx.doi.org/10.1016/j.jcis.2006.03.023.
[57] D.L. Weaire, S. Hutzler, The Physics of Foams, Oxford University Press, 2001.
[58] L. Zhornitskaya, A.L. Bertozzi, Positivity-preserving numerical schemes for lubrication-type equations, SIAM J. Numer. Anal. 37 (2) (2000) 523–555, http://dx.doi.org/10.1137/S0036142998335698.