Winter 2023

Simulating incompressible thin-film fluid with a Moving Eulerian-Lagrangian Particle method

Yitong Deng
Dartmouth College, thomasdeng1999@hotmail.com

Follow this and additional works at: https://digitalcommons.dartmouth.edu/masters_theses

Part of the Graphics and Human Computer Interfaces Commons

Recommended Citation
Deng, Yitong, "Simulating incompressible thin-film fluid with a Moving Eulerian-Lagrangian Particle method" (2023). Dartmouth College Master's Theses. 120.
https://digitalcommons.dartmouth.edu/masters_theses/120

This Thesis (Master's) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Master's Theses by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.
SIMULATING INCOMPRESSIBLE THIN-FILM FLUID WITH A MOVING EULERIAN-LAGRANGIAN PARTICLE METHOD

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of
Master of Science
in
Computer Science
by
Yitong Deng
Guarini School of Graduate and Advanced Studies
DARTMOUTH COLLEGE
Hanover, New Hampshire
January 2023

Examining Committee:

Prof. Bo Zhu, Chair

Prof. Wojciech Jarosz

Prof. Soroush Vosoughi

F. Jon Kull, Ph.D.
Dean of the Guarini School of Graduate and Advanced Studies
Abstract

In this thesis, we introduce a Moving Eulerian-Lagrangian Particle (MELP) method, a mesh-free method to simulate incompressible thin-film fluid systems: soap bubbles, bubble clusters, and foams. The realistic simulation of such systems depends upon the successful treatment of three aspects: (1) the soap film’s deformation due to the tendency to minimize the surface energy, giving rise to the bouncy characteristics of soap bubbles, (2) the tangential fluid flow on the thin film, causing the thickness to vary spatially, which in conjunction with thin-film interference creates evolving and highly sophisticated iridescent color patterns, (3) the topological changes due to collision, separation, and fragmentation, which may create partition surfaces and non-manifold junctions that spontaneously settle into honeycomb structures due to force balance. The interleaving complexities from all three fronts render the task of accurately and affordably simulating thin-film fluid an open problem for the Computational Physics and Computer Graphics community.

The proposed MELP method tackles these multifaceted challenges by employing a novel, bi-layer particle structure: a sparse set of Eulerian particles for dynamic interface tracking and PDE solving, and a fine set of Lagrangian particles for material and momentum transport. Such a design provides crucially advantageous numerical traits compared to existing frameworks. Compared to mesh-based methods, MELP’s particle-based nature makes it topologically agnostic, which allows it to conveniently simulate topological changes such as bubble-cluster formation and thin-film rupture.
Furthermore, these Lagrangian structures carry out fluid advection naturally, conserve mass by design, and track “sub-grid” flow details. Compared to existing particle methods, our bi-layer design improves drastically on the computational performance in terms of both stability and efficiency.

The advantage of this design will manifest in a wide range of experiments, including dynamic foam formation, Rayleigh-Taylor instability, Newton Black Films, and bubble bursting, showing an increased level of flow detail, increased number of regions in bubble clusters, and increased flexibility to recreate multi-junction formation on-the-fly. Furthermore, we validate its physical correctness against a variety of analytical baselines, by successfully recovering the equilibrium dihedral and tetrahedral angles, the exponential thickness profile of drainage under gravity, the curvature of partition surfaces, and the minimum surface area of double-bubbles.
Acknowledgement

I would like to thank my parents: Guogang Deng and Ruofang Du, for always lending me the freedom and support to explore what I am truly interested in as a student. Despite being thousands of miles away from Dartmouth, they’ve always made sure to convey to me their belief and appreciation, which anchors me emotionally during the interims of frustration and confusion. This thesis, along with all the joy I’ve had in its creation, would not have been possible without them.

I would like to thank my advisor: Professor Bo Zhu for his ongoing mentorship. When I started working with him as an undergraduate novice, he saw in me the research potential that I didn’t even think I had, and offered me a level of trust and opportunity way beyond what I could ever hope for. It has always been a pleasure working with him. His insight, enthusiasm, and patience inspired and nurtured my motivation to pursue further studies in Computer Graphics and physical simulation.

I want to thank Mengdi Wang for his crucial work in designing the code structure for this project. Collaborating with him has been a great learning experience thanks to his erudition and willingness to share. I would like to thank Dr. Shiying Xiong for teaching me much about differential geometry and fluid mechanics. I would like to thank Xiangxin Kong for laying down the foundation of the rendering pipeline. I would like to thank Zangyueyang Xian for his help in debugging the Multi-MELP implementation. I would also like to thank Annie Tang and Wanxin Hu for helping to render the results.
# Contents

Abstract .......................................................... ii
Acknowledgement ...................................................... iv

1 Introduction ....................................................... 1
   1.1 Fluid Thin Films .............................................. 1
   1.2 Previous Works ............................................... 2
   1.3 Motivation ................................................... 3
   1.4 Thesis Structure ............................................. 4

2 Thin-Film Continuous Model .................................. 5
   2.1 Geometry .................................................... 5
   2.2 Dynamics .................................................... 7

3 The Moving Eulerian-Lagrangian Particle (MELP) Method .... 10
   3.1 Overview .................................................... 10
   3.2 MELP ........................................................ 11
      3.2.1 Algorithm ............................................... 13
      3.2.2 $\mathcal{L}_2\mathcal{E}$ Transfer ............................... 15
      3.2.3 Geometry Computation .................................. 16
      3.2.4 Dynamics Computation .................................. 17
      3.2.5 $\mathcal{E}_2\mathcal{L}$ Transfer ................................ 21
3.2.6 $\mathcal{E}$ Advance ........................................... 21
3.2.7 $\mathcal{L}$ Advance ........................................... 22
3.2.8 Implementation Details ................................... 23

4 Multi-MELP 26
4.1 Overview ......................................................... 26
4.2 Motivating Example ............................................ 28
4.3 Multi-Region Tracking .......................................... 29
4.4 Collision Handling .............................................. 30
4.5 Surface Tension Sharing ....................................... 30
4.6 Material Transfer ............................................... 32

5 Experiments and Results 33
5.1 Numerical Validation ............................................ 33
5.2 Comparison with Single-Layer Particle Method ............. 36
5.3 Examples ......................................................... 38

6 Conclusion 50

Bibliography 52
List of Figures

2.1 An illustration of our one-sided geometric model. Left: a thin film lamella with thickness and local frames. Right: a triple-junction represented by three lamellae without directly modeling the singularity at $E$. ................................................................. 6

3.1 Left: the local density of $L$ determines the thickness. Right: $L$ advect with the full velocity, $E$ advect with the normal velocity. ............ 11

3.2 The computation workflow of a single simulation step in our proposed MELP framework. ................................................................. 13

4.1 The schematics of a multi-MELP simulation step; $L_1$ through $L_5$ are 5 MELP objects corresponding to the 5 regions. ......................... 27

4.2 The formation of a double-bubble. The top row shows the particle perspective; the bottom row depicts the control volumes with force analysis. ................................................................. 28

4.3 Left: illustration of the force $f_{st, atm}$ that must be accounted for. Right: material transport is done by transporting $L$ particles directly. . . . . 31
5.1 Left: in a double and a triple-bubble, three pieces of lamellae meet at $\approx 120^\circ$ angles along the border. Right: in a quadruple bubble, 6 partition surfaces (highlighted) form 4 borders (red arrows) meeting at $\approx 109^\circ$ angles.

5.2 Double-bubbles with different size ratios. The smaller the red bubble is, the more it protrudes into the larger (blue) one.

5.3 Top-left: curvatures of the partition surface for double-bubbles of different size ratios compared to analytical values. Top-right: thickness profile under gravity compared to analytical values. Bottom: the evolving surface area of two bubbles as they merge into a double-bubble.


5.5 Comparison with Wang et al. [2021]. Left: test of convergence to equilibrium thickness. Right: comparison of computational cost.

5.6 Left: Newton’s interference fringes under gravity. Right: comparison with Wang et al. [2021]: top-left: initial set-up; top-right: converged result of MELP; bottom-left: diverged result of Wang et al. [2021], bottom-right: converged result of Wang et al. [2021] with reduced parameters.

5.7 Comparison of the simulated flow quality of our proposed MELP method (top) and Wang et al. [2021] (bottom).

5.8 The flow, deformation, and bursting of a giant bubble, similar to the experiment done in Wang et al. [2021] Figure 5.

5.9 Different frames of a deforming rectangular film with black spots.

5.10 Different frames of a deforming half bubble with black spots.
5.11 Interaction among bubbles of different sizes, showcasing our system’s ability to restore the equilibrium states.

47

5.12 Four bubbles merge, reorganize from an unstable equilibrium to a stable one, and eventually disintegrate.

47

5.13 The dynamic formation of a double-bubble with intricate flow patterns, simulated by our proposed method. With the appropriate treatment of surface tension near the junction, two bubbles spontaneously settle into meeting angles of $\approx 120^\circ$, recovering what is known as the Plateau border.

48

5.14 300 bubbles falling into a container, forming a foam mountain.

48

5.15 13 bubbles merging together, with a heat source at the bottom creating “cyclones” on the surfaces of the bubbles.

49
Chapter 1

Introduction

1.1 Fluid Thin Films

Soap films, vapor bubbles, bubbles clusters, and liquid foams are seemingly mundane objects that we encounter every day while we are washing hands, rinsing dishes, boiling water, cooking food, and pouring out coffee or beer. Ubiquitous as they are, these fluid thin-film systems generate tremendous intrigue to scientists as they exhibit remarkably complex geometry and dynamics due to the coupled interaction between the interfacial flow, surface deformation, and topological evolution. Even the simplest cases of such systems, such as single bubbles or disks with manifold topology, can carry great complexity and value, as the strong mismatch between the length scale and thickness scale leads to high Reynolds numbers, making them the ideal test bed for observing and understanding eddies and turbulence. Such complex flow in conjunction with thin-film interference [Smits and Meyer, 1992, Belcour and Barla, 2017, Glassner, 2000, Iwasaki et al., 2004, Jaszkowski and Rzeszut, 2003], creates the fascinating, rapidly-developing, iridescent color patterns, that imbue the aesthetic and artistic significance. To recreate these phenomena, we would need to solve the Euler equations for inviscid fluid on deforming fluid domains, which is a challenging
1.2 Previous Works

problem on its own due to the necessity of efficient, dynamic data structures and robust PDE-solving techniques.

The situation would get even more challenging and interesting when we consider thin films with topological changes and non-manifold geometries, which occur abundantly for such systems. An example of such topological changes would be separating soap-film catenoids, where a single piece of soap tube is stretched to collapse into two soap disks; and an example of such non-manifold configurations is a triple-bubble, where an edge is formed at the junction of three bubbles along which not a single normal direction can be defined. In order to simulate foam systems with such topological complications, we need to extend the PDE-solving machinery developed for manifold thin films to work with singular points and edges. Then, we must also model the specific dynamic evolution of these non-manifold structures due to surface tension. Finally, we must also design the algorithm and discretization scheme to be ready for topological changes at all times. The interleaving complexities among all these levels render the full-scale simulation of incompressible fluid on thin films and foams a very challenging problem.

1.2 Previous Works

As discussed above, the successful simulation of thin films and foams amounts to the handling of three aspects — the 1) turbulent flow, 2) deforming geometry, and 3) evolving, non-manifold topology. To date, various works in computer graphics and computational physics have tackled the challenge of devising effective geometric data structures and PDE solvers to capture the vivid flow details on dynamic membranes. For instance, researchers have derived reduced governing equations [Chomaz, 2001, Couder et al., 1989] for thin film flows, constructed numerical algorithms to generate highly detailed surface flow on fixed spherical domains [Hill and Henderson, 2016,
1.3 Motivation

Yang et al., 2019, Huang et al., 2020], utilized dynamic meshes to enable thin-film deformation [Saye and Sethian, 2016, Da et al., 2015, Ishida et al., 2020, 2017], utilized level-sets to compute surface tension effects [Zheng et al., 2009], simulated bubble deformation and bursting purely with particles [Wang et al., 2020, 2021], and used a hybridization of meshes and points to compute interfacial phenomena [Chen et al., 2021, Hyde et al., 2020].

In the simulation of foams — multiple bubbles connected via non-manifold junctions, the main challenge transitions to modeling the dynamics of the junctions. Extensive research efforts have been devoted to the theoretical understanding [Cohen-Addad et al., 2013] and numerical validation [Saye and Sethian, 2013] of the dynamics and equilibrium states of these junctions. In geometric processing, researchers explore non-manifold differential operators that can accommodate PDE solving on foam structures [Sharp and Crane, 2020]. Saye and Sethian [2013] construct a comprehensive framework that takes into account the thickness evolution on a microscopic scale. In computer graphics, researchers have also developed continuum-based approaches to model the macroscopic behavior of foam materials [Ram et al., 2015, Yue et al., 2015].

1.3 Motivation

Despite the inspiring progress, developing an integrated algorithm that can jointly 1) capture the surface flow details at a high (pixel-level) resolution and 2) accommodate the complex geometric and topological evolution, remains a recalcitrant technical gap that hinders thin-film/foam simulation from advancing to the next level of visual authenticity. In particular, mesh-based methods are dynamic and efficient, but are inconvenient for handling topological changes. Particle-based methods are topologically flexible, but not as well-suited for accurate, large-scale PDE solving. Volumetric, grid-based methods can resolve fluid equations fully, stably, and accurately, but are
1.4 Thesis Structure

highly costly and lack the ability to capture the intricate, sub-grid color patterns that are essential to the visual appeal. Hence, our motivation is to unify the topological flexibility of particles, the space efficiency of surface meshes, and the stability of grids in a comprehensive numerical simulation system.

1.4 Thesis Structure

The rest of the thesis will be structured as follows: in Chapter 2, we will present and discuss our continuous model of thin fluid films. In Chapter 3, we will present the discretized MELP method, addressing its high-level intention, low-level implementation, and the different ways that it uniquely caters to the continuous physics. The MELP method on its own is capable of resolving single-lamellae scenarios but cannot accommodate multi-bubble settings. In Chapter 4, we will introduce Multi-MELP, which is a natural extension of MELP to enable non-manifold simulation. In Chapter 5, we will showcase a variety of simulated results and validation experiments which will be analyzed qualitatively and quantitatively. In particular, in-depth comparison tests will be performed against previous benchmarks. In Chapter 6, we will conclude by addressing its current limitations and future research directions.
Chapter 2

Thin-Film Continuous Model

2.1 Geometry

Lamellae  As depicted on the left of Figure 2.1, a thin film lamella is a layer of fluid trapped between two air-liquid interfaces. We refer to one of the interfaces as the base surface $S_B$, which is assumed to be a connected, orientable Riemannian 2-manifold in $\mathbb{R}^3$. A base surface may be open with boundary (e.g. a disk) or closed (e.g. a bubble). The orientability allows a continuous field $\textbf{n} : S_B \rightarrow S^2$ of outward pointing, unit normal vectors to be defined. For a disk, the outward direction is defined arbitrarily, while for a bubble, the outward direction points away from the enclosed volume.

At each point $p \in S_B$, given the normal vector, a tangent plane is uniquely determined, for which we construct an orthonormal basis with $\{\textbf{e}_1(p), \textbf{e}_2(p)\}$. We then define a field of local frames $\mathbf{R} : S_B \rightarrow SO3$ as $\mathbf{R}(p) = \{\textbf{e}_1(p), \textbf{e}_2(p), \textbf{n}(p)\}$ with coordinates $(u, v, z)$. Additionally, we define a field of mean curvatures $H : S_B \rightarrow \mathbb{R}$, a field of metric tensors $g : S_B \rightarrow \mathbb{R}^{2 \times 2}$, and a field of thickness $\eta : S_B \rightarrow \mathbb{R}$.

Our definition of the geometry of a lamella $L$ is then the 5-tuple: $(S_B, \mathbf{R}, H, g, \eta)$. The other interface $S_T$ can be defined as the image $f(S_B)$ of the function $f : S_B \rightarrow \mathbb{R}^3$.
with $f(p) = p + \eta(p)n(p)$. The geometric quantities, such as the mean curvature, is specified on $S_B$, but not on $S_T$. In these cases, we assume the quantity at $f(p) \in S_T$ equals that at $p \in S_B$. This is reasonable since the film’s thickness scale ($10^{-7}$ m) is minuscule compared to its length scale ($10^{-2}$ m).

**Junctions** As depicted on the right of Figure 2.1, a junction is formed at $E$ where multiple pieces of lamellae come into contact. These junctions are typically considered *non-manifold* when thin films are viewed as infinitesimally-thin mathematical surfaces. But from a volumetric standpoint, the junction $E$ is indeed a bulk of liquid confined by its three-manifold interfaces. In this light, we model this triple-bubble with three lamellae: $L_A$, $L_B$ and $L_C$, with $(S_B)_A$, $(S_B)_B$ and $(S_B)_C$ the three-manifold interfaces that together delineate $E$. The entire liquid volume is the union of the volume represented by $L_A$, $L_B$, and $L_C$.

It should be noted that near the contact areas, $(S_T)_A$, $(S_T)_B$, and $(S_T)_C$ are no longer air-liquid interfaces, but rather pseudo-interfaces between different regions.
We do not enforce that these pseudo-interfaces coincide exactly, assuming that imperfections at this level are negligible to the overall dynamics.

### 2.2 Dynamics

**Euler equations** We derive our thin film dynamic model based on the Euler equations for inviscid, incompressible flow with surface tension:

\[
\begin{align*}
\frac{\rho}{D} \frac{Du}{Dt} &= -\nabla p + f_\sigma + f_{ext}, \\
\nabla \cdot u &= 0,
\end{align*}
\] (2.1)

where \( \rho \) denotes the density, \( p \) the pressure, \( f_\sigma \) the surface tension force per unit volume, and \( f_{ext} \) the external forces, *e.g.* gravity.

**Surface tension** The surface tension force \( f_\sigma \) in Equation 2.1 is computed as

\[
f_\sigma = (\sigma H n + \nabla_s \sigma) \cdot \delta_I,
\] (2.2)

with \( \sigma \) denoting the surface tension coefficient, \( H \) and \( n \) the mean curvature and normal vector on the interfaces, \( \nabla_s \) the surface gradient operator, and \( \delta_I \) the Dirac delta function that is non-zero only on the interfaces. The first term: \( \sigma H n \) reflects the normal stress prescribed by the Young-Laplace Law [Finn, 1999], and the second term: \( \nabla_s \sigma \) reflects the tangential stress corresponding to the Marangoni effect. The surface tension \( \sigma \) relates to the surfactant concentration \( \Gamma \) as \( \sigma = \sigma_0 - \bar{R}T \Gamma \), where \( \sigma_0 \) is the surface tension for pure water, \( \bar{R} \) the ideal gas constant, and \( T \) the temperature [Xu et al., 2006].
2.2 Dynamics

Lamellae Following Ishida et al. [2020], we separate Equation 2.1 into its normal and tangential components as:

\[
\begin{align*}
\rho \frac{Du^\perp}{Dt} &= -\frac{\partial p}{\partial z} n + \delta_I \sigma H n + f^\perp_{\text{ext}}, \\
\rho \frac{Du^\top}{Dt} &= \delta_I \nabla_s \sigma + f^\top_{\text{ext}},
\end{align*}
\] (2.3)

Here we use the superscript \( \perp \) for normal components and \( \top \) for tangential components. The normal equation is obtained via projection, and the tangential equation is obtained via asymptotic simplification under the lubrication assumption [Chomaz, 2001, Huang et al., 2020]. We further assume that the fluid pressure gradient along \( z \) is negligible, as is done in Chomaz [2001], Ishida et al. [2020], so that \( \frac{\partial p}{\partial z} \) only reflects the air-liquid pressure jumps. Hence we have

\[
\frac{\partial p}{\partial z} = \delta_B \cdot (\tilde{p} - p_{\text{in}}) + \delta_T \cdot (p_{\text{out}} - \tilde{p}),
\] (2.4)

with \( \delta_B \) and \( \delta_T \) being Dirac delta functions that represent \( S_B \) and \( S_T \) respectively, and satisfying \( \delta_B + \delta_T = \delta_I \); \( p_{\text{in}} \) and \( p_{\text{out}} \) being inside and outside air pressures with the orientation decided by the normal vector; and \( \tilde{p} \) the characteristic fluid pressure, which is assumed constant here since the air-liquid pressure difference is much greater. Adding in the conservation equations of the surfactant concentration \( \Gamma \) and membrane thickness \( \eta \), we rewrite Equation 2.3 to obtain the full dynamic model:

\[
\begin{align*}
\rho \frac{Du^\perp}{Dt} &= -\frac{1}{\rho} \frac{\partial p}{\partial z} n + \delta_I (\sigma_0 - \bar{R} \Gamma) H n + \frac{1}{\rho} f^\perp_{\text{ext}}, \\
\rho \frac{Du^\top}{Dt} &= -\frac{2\bar{R} T}{\rho \eta} \nabla_s \Gamma + \frac{1}{\rho} f^\top_{\text{ext}}, \\
\frac{D\Gamma}{Dt} &= -\Gamma \nabla_s \cdot u, \\
\frac{D\eta}{Dt} &= -\eta \nabla_s \cdot u.
\end{align*}
\] (2.5)
2.2 Dynamics

Junctions As described in Section 2.1 above, a junction is partitioned by a set of lamellae into several regions, akin to the “symmetry units” in Koehler et al. [2004]. We assume the pseudo-interfaces among these regions impose slip boundary conditions, where the tangential flows are unconstrained while the normal velocities are matched. Material can transport between regions. In particular, near a junction a $\nabla_s$ operator shall be replaced by $\nabla$ in Equation 2.5. Due to our method’s codimensional nature, we opt not to evaluate volumetric derivatives explicitly, but rather simulate its behaviors using particles.
Chapter 3

The Moving Eulerian-Lagrangian Particle (MELP) Method

3.1 Overview

To simulate our continuous thin film model presented above, we design the Moving Eulerian-Lagrangian Particle (MELP) method: a novel, mesh-free method that can stably and efficiently simulate 1) complex, turbulent flow at a high level of detail, 2) aggressive shape deformation under surface tension, and 3) accurate evolution of non-manifold topologies according to Plateau’s laws. As our core contribution, we discretize fluid thin films using two collaborating particle sets: a sparse set of Eulerian particles for dynamic interface tracking and PDE solving, and a fine set of Lagrangian particles for material and momentum transport. This separation of tasks between deformation tracking and flow tracking enables enhanced performance on both fronts. The Eulerian particles can maintain a stable, uniform discretization despite the turbulent surfacial flow, as they can advect only with normal velocities, and freely redistribute in the tangent plane; the Lagrangian particles become more computationally affordable as they are responsible for advection only, and can thus
3.2 MELP

As shown in Figure 3.1, the MELP framework consists of a set of sparse Eulerian particles $\mathcal{E}$ and a set of fine Lagrangian particles $\mathcal{L}$. The Lagrangian particles carry physical quantities such as mass and volume, and perform material and momentum transport by shifting their positions. The Eulerian particles track the deformed thin film while maintaining uniform discretization, thereby offering a stable computational stencil on the moving surface. The two particle sets will collaborate both in repre-

Figure 3.1: Left: the local density of $\mathcal{L}$ determines the thickness. Right: $\mathcal{L}$ advect with the full velocity, $\mathcal{E}$ advect with the normal velocity.

be deployed at larger amounts to track out more sophisticated and accurate flow patterns. Our MELP method can also be seen as an extension of the Particle-in-Cell (PIC) method: first, we extend its simulation domain from a 2D Euclidian plane to deforming 2-manifolds, and we formulate a set of surface differential operators to enable this extension; secondly, we replace the grid with our meshless Eulerian particles, so that the entire system is highly flexible with topological changes. To do so, we devise a set of particle-to-particle gathering and interpolation schemes to mimic the particle-grid interactions in traditional PIC methods.
senting the thin film geometry and in solving the dynamic equations.

**Geometry** We have defined a lamella as a 5-tuple \((S_B, R, H, g, \eta)\). As shown on the left of Figure 3.1, \(S_B\) is uniformly discretized by the \(E\) particles, each of which controls some area \(a\). Given the point set, we approximate the local frame \(R\), mean curvature \(H\), and metric tensor \(g\) following Wang et al. [2020]. The remaining variable \(\eta\) will be determined by the distribution of \(L\) particles. As displayed on the left of Figure 3.1, due to the incompressibility constraint, the denser the \(L\) particles are, the larger \(\eta\) becomes — an idea leveraged by previous works [Wang et al., 2021, Solenthaler, 2011]. Each \(E\) particle then controls a fluid column with volume \(V = a \cdot \eta\).

**Dynamics** In solving the dynamic equations, the \(E\) and \(L\) particles collaborate in a similar fashion as the grids and particles in hybrid Eulerian-Lagrangian methods. The advection term is handled in the Lagrangian manner by shifting the positions of \(L\) particles. The projection term is solved on the sparser, uniformly-distributed \(E\) particles using Implicit Incompressible SPH (IISPH) [Ihmsen et al., 2013]. The material and momentum transfer between \(E\) and \(L\) is achieved using Affine Particle-In-Cell (APIC) [Jiang et al., 2015].

**Interface Tracking** It is shown that the advection of an interface is unaffected by the tangential velocity [Gibou et al., 2018]. As seen on the right of Figure 3.1, since the \(E\) particles are purely geometric, we can let them advect with the normal component of the material velocity without affecting the dynamics. We further incorporate the arbitrary Lagrangian-Eulerian idea, in which an artificial tangential velocity is granted on \(E\) to avoid deformation-induced clustering [Sahu et al., 2020]. We compute this artificial velocity following the particle shifting approach in the SPH literature [Lind et al., 2012].
3.2 MELP

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W$</td>
<td>3D kernel function</td>
</tr>
<tr>
<td>$\mathcal{W}$</td>
<td>2D kernel function</td>
</tr>
<tr>
<td>$x_A$</td>
<td>position of particle $A$</td>
</tr>
<tr>
<td>$W(A, B)$</td>
<td>$W(x_A - x_B)$</td>
</tr>
<tr>
<td>$\perp_A(w)$</td>
<td>project $w$ to the normal direction of $A$</td>
</tr>
<tr>
<td>$\top_A(w)$</td>
<td>project $w$ to the tangent plane of $A$</td>
</tr>
<tr>
<td>$r$</td>
<td>kernel support radius</td>
</tr>
<tr>
<td>$\mathcal{N}_E(A)$</td>
<td>all $E$ particles within radius $r$ from $x_A$</td>
</tr>
<tr>
<td>$\mathcal{N}_L(A)$</td>
<td>all $L$ particles within radius $r$ from $x_A$</td>
</tr>
<tr>
<td>Proj($A$)</td>
<td>project $x_A$ onto $S_B$</td>
</tr>
</tbody>
</table>

Table 3.1: A list of symbols and expressions for the MELP algorithm.

Figure 3.2: The computation workflow of a single simulation step in our proposed MELP framework.

### 3.2.1 Algorithm

The basic MELP procedure is illustrated in Figure 3.2 with labels corresponding to the following stages:

(1) **$L2E$ Transfer**: Transfer mass $m$, surfactant $c$, volume $V$, and momentum $p$ from $L$ to $E$ (Algorithm 1).
3.2 MELP

**Algorithm 1** \(\mathcal{L}2\mathcal{E}\) transfer

1: for each particle \(L \in \mathcal{L}\) do
2: \(\text{Compute } \alpha_L = \sum_{E \in N^E(L)} W(E, L)\).
3: for each particle \(E \in \mathcal{E}\) do
4: \(\text{for } q \in \{m, c, V, p\} \text{ do}\)
5: \(\text{Compute } q_E \text{ according to Equation 3.1}\)
6: \(\text{Compute } \hat{p}_E \text{ according to Equation 3.2}\)
7: \(u_E \leftarrow p_E + \hat{p}_E, u_E^⊥ \leftarrow T_E(u_E), u_E^\top \leftarrow \nabla_E(u_E)\)

**Algorithm 2** Geometry computation

1: for each particle \(L \in \mathcal{L}\) do
2: Evolve \(\eta_L\) according to Equation 3.3
3: for each particle \(E \in \mathcal{E}\) do
4: Compute \(a_E\) according to Equation 3.4
5: Compute \(\eta_E = \frac{V_E}{a_E}\)
6: Compute \(H_E\) and \(g_E\) according to Equation 3.5

(2) **Geometry Computation:** Compute thickness \(\eta\) for \(\mathcal{E}\) and \(\mathcal{L}\) particles; update control area \(a\), mean curvature \(H\) and metric tensor \(g\) for \(\mathcal{E}\) particles (Algorithm 2).

(3) **Dynamics Computation:** Solve Equation 2.5 on \(\mathcal{E}\) in the normal and tangential directions; update velocity \(u\) (Algorithm 3).

(4) **\(\mathcal{E}2\mathcal{L}\) Transfer:** Each \(\mathcal{L}\) particle interpolates \(u\) from nearby \(\mathcal{E}\) particles (Algorithm 4).

(5) **\(\mathcal{E}\) Advance:** Each \(\mathcal{E}\) particle advects with the normal velocity, deforming the surface \(S_B\). On the updated \(S_B\), shift \(\mathcal{E}\) particles tangentially to maintain uniform distribution (Algorithm 6).

(6) **\(\mathcal{L}\) Advance:** Each \(\mathcal{L}\) particle advects with the full velocity. Afterwards, project their positions onto \(S_B\) (Algorithm 7).

The following subsections will go through each of the six stages in detail. We
3.2 MELP

Algorithm 3 Dynamics computation with $\mathcal{E}$

1: Compute enclosed volume $\hat{V}$ with Equation 3.6
2: Compute enclosed pressure $p_{in}$ with the ideal gas law
3: Solve $\Gamma$ implicitly with Equation 3.9
4: for each particle $E \in \mathcal{E}$ do
5: \hspace{1em} Compute $\frac{Du^E_E}{Dt}$ with Equation 3.7
6: \hspace{1em} Compute $\frac{Du^E_\perp}{Dt}$ with Equation 3.25
7: \hspace{1em} Update $u^E_\perp$ and $u^E_\perp$ using symplectic Euler with $\Delta t$

Algorithm 4 $\mathcal{E}$2$L$ transfer

1: for each particle $L \in \mathcal{L}$ do
2: \hspace{1em} Compute $\mathbf{u}_L$ according to Equation 3.26
3: \hspace{1em} Compute $\mathbf{B}_L$ according to Equation 3.27
4: \hspace{1em} Compute $\mathbf{D}_L$ according to Equation 3.28

define relevant symbols and expressions in Table 3.1.

3.2.2 $\mathcal{L}$2$\mathcal{E}$ Transfer

For a generic quantity $q$, we conduct conservative transfer from $\mathcal{L}$ to $\mathcal{E}$ as:

$$ q_E = \sum_{L \in N^E(E)} \hat{W}(E,L) \cdot q_L, \quad (3.1) $$

where $\hat{W}$ has the partition of unity quality $\sum_{E \in N^E(L)} \hat{W}(E,L) = 1$. Given an SPH kernel $W$, we define $\hat{W}(E,L) = W(E,L)/\alpha_L$, where $\alpha_L = \sum_{E \in N^E(L)} W(E,L)$. Then, we transfer mass $m$, surfactant $c$, volume $V$, and momentum $p$ according to Equation 3.1. Furthermore, we construct an affine momentum $\hat{p}$ for APIC:

$$ \hat{p}_E = \sum_{L \in N^E(E)} \hat{W}(E,L) \cdot [\mathbf{B}_L(D_L)^{-1}_L (x_E - x_L)], \quad (3.2) $$

where $\mathbf{B}$ and $\mathbf{D}$ are the affine state and inertia-like tensor carried by the $\mathcal{L}$ particles. We reconstruct velocity $\mathbf{u}_E$ with $\mathbf{u}_E = \frac{p_E + \hat{p}_E}{m_E}$, and split it into its normal and tangen-
3.2 MELP

Algorithm 5 \( \mathcal{E} \) redistribution

1: Initialize \( \vec{u}^E \) to 0
2: while \( |\delta_{\text{max}} - \delta_{\text{min}}| \geq \varphi \) do
3: \hspace{1em} Solve \( C \) implicitly using Equation 3.30
4: \hspace{1em} for each particle \( E \in \mathcal{E} \) do
5: \hspace{2em} \( \frac{D\vec{u}^E}{Dt} \leftarrow -\beta \nabla_s C \)
6: \hspace{2em} \( \vec{u}^E \leftarrow \vec{u}^E + \Delta t \frac{D\vec{u}^E}{Dt} \)
7: \hspace{2em} \( \vec{x}^E \leftarrow \vec{x}^E + \Delta t \vec{u}^E \)
8: \hspace{1em} Update local frames and metric tensors
9: \hspace{1em} Update \( \delta \) according to Equation 3.4
10: \hspace{1em} for each particle \( E \in \mathcal{E} \) do
11: \hspace{2em} \( \vec{u}^E \leftarrow \vec{u}^E + \vec{u}^E \)

Algorithm 6 \( \mathcal{E} \) Advance

1: for each particle \( E \in \mathcal{E} \) do
2: \hspace{1em} Update \( \vec{u}^E \) with Equation 3.29
3: \hspace{1em} Update \( \vec{x}^E \) using symplectic Euler with \( \Delta t \)
4: \hspace{1em} Update local frames and metric tensors
5: \hspace{1em} Redistribute \( \mathcal{E} \) with Algorithm 5

Tial components as \( u^E_\perp = \perp_E (u_E), u^E_\top = \top_E (u_E) \), which are to be evolved separately in Section 3.2.4.

3.2.3 Geometry Computation

For each \( L \in \mathcal{L} \), we evolve thickness \( \eta_L \) according to Equation 2.5 after temporal discretization:

\[
\eta_{L(t)} = \eta_{L(t-1)} - \Delta t \eta_{L(t-1)} \nabla_s \cdot u. \tag{3.3}
\]

For each \( E \in \mathcal{E} \), we compute number density \( \delta_E \) and area \( a_E \) as:

\[
\delta_E = \sum_{E' \in \mathcal{N}_E(E)} \overline{W}(E, E'), \quad a_E = \frac{1}{\delta_E}. \tag{3.4}
\]
3.2 MELP

Algorithm 7 \( \mathcal{L} \) Advance

1: for each particle \( \mathcal{L} \in \mathcal{L} \) do
2: \quad Update \( \mathbf{x}_L \) with \( \mathbf{u}_L \) using symplectic Euler with \( \Delta t \)
3: \quad \( \mathbf{x}_L \leftarrow \text{Proj}(\mathcal{L}) \)

Then, since the particle volume \( V_E \) is already transferred to \( \mathcal{E} \) during the \( \mathcal{L}2\mathcal{E} \) step, the thickness \( \eta_E \) can be computed by \( \eta_E = V_E/a_E \).

For each \( E \in \mathcal{E} \), a neighboring particle \( E' \in N^\mathcal{E}(E) \) has coordinates in \( E \)'s local frame: \((u, v, z) = ((\mathbf{x}_{E'} - \mathbf{x}_E) \cdot \mathbf{e}_1, (\mathbf{x}_{E'} - \mathbf{x}_E) \cdot \mathbf{e}_2, (\mathbf{x}_{E'} - \mathbf{x}_E) \cdot \mathbf{n}) \). With \( \nabla_E \) being the 2D differential operator on the tangent plane of \( E \), we compute the mean curvature \( H_E \) and metric tensor \( g_E \) using neighboring \( \mathcal{E} \) particles as:

\[
\begin{align*}
H_E & = -\frac{1}{2} \nabla_E \cdot \left( \frac{\nabla z}{\sqrt{1 + \nabla z^2}} \right) \approx -\frac{1}{2} \nabla_E^2 z, \\
g_E & = \begin{bmatrix}
1 + (\frac{\partial z}{\partial u})^2, & \frac{\partial z}{\partial u} \frac{\partial z}{\partial v} \\
\frac{\partial z}{\partial u} \frac{\partial z}{\partial v}, & 1 + (\frac{\partial z}{\partial v})^2
\end{bmatrix}.
\end{align*}
\tag{3.5}
\]

3.2.4 Dynamics Computation

Normal Dynamics

Similar to Ishida et al. [2020], we assume that the normal velocity \( \mathbf{u}_E^\perp \) is constant in an \( \mathcal{E} \) particle \( E \)'s control column, i.e. \( \mathbf{u}_E^\perp = \frac{1}{V_E} \int_E \mathbf{u}_E^\perp dV \). Therefore \( \frac{\mathbf{D}u_E^\perp}{\mathbf{D}t} \approx \frac{1}{V_E} \int_E \mathbf{D}u_E^\perp dV \). We then need to integrate the right-hand side of the normal component in Equation 2.5. For the term \( \frac{\partial p}{\partial z} \), whose expression is given by Equation 2.4, integrating over the control column of \( E \) yields \( a_E \cdot (p_{\text{out}} - p_{\text{in}}) \). We assign \( p_{\text{out}} \) to be the atmospheric pressure \( p_{\text{atm}} \). If the lamella is open (disk), then we assign \( p_{\text{in}} = p_{\text{atm}} \). If the lamella is closed (bubble), we compute the enclosed pressure using the ideal gas law as: \( p_{\text{in}} = n_0 \tilde{R} T/\tilde{V} \) with \( n_0 \) being the enclosed molar mass and \( \tilde{V} \) the enclosed
volume, which we compute as:

\[ \hat{V} = \sum_{E \in \mathcal{E}} \zeta \cdot \frac{1}{3} a_E (O - E), \quad \zeta = \begin{cases} 
  1, & (O - E) \cdot n_E \geq 0, \\
  -1, & \text{otherwise}, 
\end{cases} \quad (3.6) \]

where \( O \) is an arbitrarily selected point in \( \mathbb{R}^3 \) [Zhang et al., 2001]. Similarly, we integrate \( \delta_t (\sigma_0 - \bar{RT}) \Gamma H \) and \( f_{\text{ext}}^\perp \) over the control column as \( 2a_E (\sigma_0 - \bar{RT}) \Gamma H E \) and \( V_E f_{\text{ext}}^\perp \) respectively. Hence we obtain the expression for \( \frac{D u_E^\perp}{Dt} \) as:

\[ \frac{D u_E^\perp}{Dt} = \frac{p_{\text{in}} - p_{\text{out}}}{\rho \eta_E} n_E + \frac{2(\sigma_0 - \bar{RT}) \Gamma H E}{\rho \eta_E} n_E + \frac{f_{\text{ext}}^\perp}{\rho}. \quad (3.7) \]

**Tangential Dynamics**

Following the temporal discretization scheme proposed by Huang et al. [2020], the thin film evolution along the tangential directions can be approximated as

\[
\begin{align*}
\frac{u^\top - u^{\top*}}{\Delta t} & = -\frac{2 \bar{RT}}{\rho \eta^*} \nabla_s \Gamma + \frac{1}{\rho} f_{\text{ext}}^{\top*}, \\
\frac{\Gamma - \Gamma^*}{\Delta t} & = -\Gamma^* \nabla \cdot u^\top, \\
\frac{\eta - \eta^*}{\Delta t} & = -\eta^* \nabla \cdot u^\top,
\end{align*}
\]

where \( u^{\top*} \), \( \Gamma^* \) and \( \eta^* \) are the respective quantities after advection, which we collect in the \( L^2 \mathcal{E} \) step.

Reorganizing Equation 3.8 yields an implicit equation of \( \Gamma \):

\[
\begin{align*}
( - \frac{1}{\Delta t \eta^*} ) \Gamma + ( \Delta t \frac{\bar{RT}}{\rho} \frac{1}{\eta^*} \nabla_s \Gamma + \Delta t \frac{\bar{RT}}{\rho} \frac{1}{\eta^*} \nabla^2_s \Gamma \\
= \nabla \cdot u^{\top*} - \frac{1}{\Delta t} + \Delta t \left( \frac{1}{\rho} \cdot f_{\text{ext}}^{\top} + \frac{1}{\rho} \nabla \cdot f_{\text{ext}}^{\top} \right). \quad (3.9)
\end{align*}
\]

We solve this equation using the Implicit Incompressible SPH method with a
relaxed Jacobi scheme with relaxation parameter $\omega = 0.2$. We will present the detailed derivation below.

**IISPH with Jacobi Iterations** To solve Equation 3.9 using Jacobi iterations, we need to compute its right-hand side (RHS) and the diagonal terms of the left-hand side (LHS), which express how the $i^{th}$ term of the LHS is related to $\Gamma_i$. We consider each of the three terms on the LHS independently and sum up the diagonal terms for each. For the first term, the diagonal terms are simply:

$$ (a_{ii})_1 = -\frac{1}{\Delta t \Gamma_i^*}. \quad (3.10) $$

For the second term on the LHS, we write out its SPH formulation:

$$ (\Delta t \frac{\bar{RT}}{\rho} (\nabla \frac{1}{\eta^*})_i) \cdot \nabla \Gamma_i \quad (3.11) $$

$$ = (\Delta t \frac{\bar{RT}}{\rho} (\nabla \frac{1}{\eta^*})_i) \cdot (\sum_{j \in N(i)} a_j (\Gamma_j - \Gamma_i) \nabla W_{ij}). \quad (3.12) $$

The diagonal coefficients would be:

$$ (a_{ii})_2 = \sum_{j \in N(i)} -a_j \nabla W_{ij} \cdot (\Delta t \frac{\bar{RT}}{\rho} (\nabla \frac{1}{\eta^*})_i). \quad (3.13) $$
3.2 MELP

For the third term on the LHS, which involves the Laplacian operator $\nabla^2$, we write out the SPH formulation of $\nabla^2 = \nabla \cdot \nabla$:

$$\nabla^2 \Gamma = \sum_{j \in N(i)} a_j (\nabla \Gamma_j - \nabla \Gamma_i) \cdot \nabla W_{ij}$$  \hfill (3.14)

$$= \sum_{j \in N(i)} a_j \left( \sum_{k \in N(j)} a_k (\Gamma_k - \Gamma_j) \nabla W_{jk} \right)$$  \hfill (3.15)

$$- \sum_{j \in N(i)} a_j (\Gamma_j - \Gamma_i) \nabla W_{ij} \cdot \nabla W_{ij}.$$  \hfill (3.16)

By the symmetry of neighbor searching (if $i$ is a neighbor of $j$, $j$ is a neighbor of $i$), one of the $k$ will be $i$, so setting $k \leftarrow i$ we express the diagonal coefficients of the third term as:

$$\left( (\nabla^2 \Gamma)_{ii} \right) = \sum_{j \in N(i)} a_j (a_i \nabla W_{ji})$$  \hfill (3.17)

$$- \sum_{j \in N(i)} -a_j \nabla W_{ij} \cdot \nabla W_{ij}$$  \hfill (3.18)

$$= \sum_{j \in N(i)} a_j (-a_i \nabla W_{ij})$$  \hfill (3.19)

$$- \sum_{j \in N(i)} -a_j \nabla W_{ij} \cdot \nabla W_{ij}$$  \hfill (3.20)

$$= - \sum_{j \in N(i)} a_j (-a_i \nabla W_{ij})$$  \hfill (3.21)

$$- \sum_{j \in N(i)} -a_j \nabla W_{ij} \cdot \nabla W_{ij}$$  \hfill (3.22)

$$\left( a_{ii} \right)_3 = \left( \Delta t \frac{\bar{RT}}{\rho \eta_t} \right) \cdot \left( \nabla^2 \Gamma \right)_{ii}.$$  \hfill (3.23)

Finally,

$$a_{ii} = \left( a_{ii} \right)_1 + \left( a_{ii} \right)_2 + \left( a_{ii} \right)_3.$$  \hfill (3.24)

Once the diagonal terms have been derived, the rest of the iterative process is analo-
gous to the original algorithm [Ihmsen et al., 2013]. In this derivation, we use \( i \) and \( j \) to represent the \( i \)th and \( j \)th \( \mathcal{E} \) particle in a MELP system.

Once \( \Gamma \) is solved, we evaluate for each \( E \in \mathcal{E} \) the tangential acceleration in Equation 2.5 as:

\[
\frac{D\mathbf{u}_E^\top}{Dt} = -\frac{2RT}{\rho \eta_E} \nabla_s \Gamma + \frac{1}{\rho} \mathbf{f}_{\text{ext}}^\top.
\] (3.25)

Then, \( \mathbf{u}_E^\perp \) and \( \mathbf{u}_E^\top \) are updated using a symplectic Euler step with \( \Delta t \).

### 3.2.5 \( \mathcal{E}/\mathcal{L} \) Transfer

For each \( L \in \mathcal{L} \), it collects three quantities from nearby \( \mathcal{E} \) particles: the velocity \( \mathbf{u}_L \), affine state \( \mathbf{B}_L \), and inertia-like tensor \( \mathbf{D}_L \) as:

\[
\mathbf{u}_L = \sum_{E \in \mathcal{N}^E(L)} \hat{W}(E, L) \cdot \mathbf{u}_E,
\] (3.26)

\[
\mathbf{B}_L = \sum_{E \in \mathcal{N}^E(L)} \hat{W}(E, L) \cdot \mathcal{T}_L(\mathbf{u}_E) \otimes \mathcal{T}_L(\mathbf{x}_E - \mathbf{x}_L),
\] (3.27)

\[
\mathbf{D}_L = \sum_{E \in \mathcal{N}^E(L)} \hat{W}(E, L) \cdot \mathcal{T}_L(\mathbf{x}_E - \mathbf{x}_L) \otimes \mathcal{T}_L(\mathbf{x}_E - \mathbf{x}_L).
\] (3.28)

### 3.2.6 \( \mathcal{E} \) Advance

Similar to the mesh velocity in Sahu et al. [2020], we define an \( \mathcal{E} \) velocity: \( \mathbf{u}_{\mathcal{E}} \), carried by individual \( \mathcal{E} \) particles, to govern their movements. In the normal direction, \( \mathbf{u}_{\mathcal{E}} \) needs to coincide with the material velocity \( \mathbf{u}_{E}^\perp \), while tangentially, \( \mathbf{u}_{\mathcal{E}} \) can use arbitrary velocities to maintain uniform distribution. We ensure this by setting:

\[
\mathbf{u}_{\mathcal{E}}(t) = \mathbf{u}_{E}^\perp(t) + \mathcal{T}_E(\mathbf{u}_{\mathcal{E}}^E(t - 1)),
\] (3.29)
3.2 MELP

which takes the tangential component of the previous $u_E^t$ and add to it the current normal velocity. Using $u_E^t$ we advance the positions of the $E$ particles using a symplectic Euler step with $\Delta t$, updating the tracked interface. We also update the local frames $R_E$ and metric tensors $g_E$. Then, we redistribute the $E$ particles to maintain uniform distribution. In particular, similar to particle shifting based on Fick’s law of diffusion, we compute a shifting velocity $\bar{u}_E^t$ to prompt particles to flow from high concentration regions to low concentration ones. Using number density $\delta$ to gauge the concentration, the problem translates to solving for a constant particle density on the surface with pseudo-pressure $C$:

$$\Delta \bar{t}^2 (-\delta^* \nabla^2_s C) = \bar{\delta} - (\delta^* + \Delta \bar{t} (-\delta^* \nabla \cdot u^E)),$$

where $\delta^*$ and $\bar{\delta}$ stand for the current and average number density of $E$ particles; and $\Delta \bar{t}$ stands for the temporal step size for redistribution. Equation 3.30 is solved using IISPH as with Equation 3.9. The full redistribution procedure is documented in Algorithm 5, where $\beta$ is the redistribution strength, and $\varphi$ the threshold deciding whether the distribution is satisfactory. We set $\beta$ to be the reciprocal of the largest value of $\nabla_s \delta$, and set $\varphi$ to be 3. The while loop in Algorithm 5 has a maximum number of iterations of 10. In practice, most advance steps require only one step of redistribution.

3.2.7 $L$ Advance

As described in Algorithm 7, $L$ particles advect with $u_L$ using a symplectic Euler step with $\Delta t$. They will be projected onto $S_B$ defined by $E$ with a Moving Least-Squares (MLS)-based approach.
3.2 MELP

3.2.8 Implementation Details

SPH Following Wang et al. [2021], we adopt SPH-based, surface differential operators as:

\[
\begin{align*}
(\nabla s q)_E &= \sum_{E' \in N^E(E)} a_{E'} (q_{E'} - q_E) \nabla_s W(E, E'), \\
(\nabla s \cdot w)_E &= \sum_{E' \in N^E(E)} a_{E'} \nabla_E (w_{E'} - w_E) \cdot \nabla_s W(E, E'), \\
(\nabla^2 s q)_E &= \sum_{E' \in N^E(E)} a_{E'} (q_{E'} - q_E)^2 \frac{2|\nabla_s W(E, E')|}{|x_E - x_{E'}|},
\end{align*}
\]

where \(\nabla_s W\) is the surface gradient of the 2D kernel function \(W\), which can be approximated as \(\nabla_s W = g \nabla W\) [Wang et al., 2020]. In practice, we approximate \(g\) with \(I_{2 \times 2}\) with no apparent degradation in performance. For both \(W\) and \(\bar{W}\), we use the Quintic spline kernel with radius \(r = 4 \cdot \Delta x\), where \(\Delta x\) reflects the \(\mathcal{E}\) particle separation. We handle particle insufficiency near solid boundaries with several layers of boundary particles with the same fineness as \(\mathcal{E}\). We also make use of the XSPH artificial viscosity [Schechter and Bridson, 2012] with viscosity parameter 0.99 to stabilize \(\mathbf{u}^E\).

Local Frame Computation At time \(t\), a particle \(E \in \mathcal{E}\) computes \(R_E\) as follows:

1. Perform PCA on \(N^E(E)\) and set the normalized eigenvector with the smallest eigenvalue as \(n'\).

2. Set \(n' = -n'\) if \(0 > n' \cdot n(t - 1)\).

3. Construct an intermediate frame \(R' = (e_1', e_2', n')\) where \(e_1'\) is an arbitrary vector perpendicular to \(n'\) and \(e_2' = n' \times e_1'\).

4. In the tangent plane, use 2D SPH to compute \(\nabla_{Ez} = (\frac{\partial z}{\partial u}, \frac{\partial z}{\partial v})^T\) and let \(e_1 = R'(1, 0, \frac{\partial z}{\partial u})^T, e_2 = R'(0, 1, \frac{\partial z}{\partial v})^T, n = e_1 \times e_2\). Finally, \(R_E = (e_1, e_2, n)\).
3.2 MELP

A particle \( L \in \mathcal{L} \) computes \( \mathbf{R}_L \) as follows:

1. Compute the average of \( \{ \mathbf{n}_E | E \in \mathcal{N}^E(L) \} \) weighted by \( W(L, E) \), set the normalized result to \( \mathbf{n} \).

2. Construct a frame \( \mathbf{R}_L = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{n}) \) where \( \mathbf{e}_1 \) is an arbitrary vector perpendicular to \( \mathbf{n} \) and \( \mathbf{e}_2 = \mathbf{n} \times \mathbf{e}_1 \).

For particle \( A \in \mathcal{L} \cup \mathcal{E} \), we compute \( \perp_A (\mathbf{w}) = (\mathbf{w} \cdot \mathbf{n}_A)\mathbf{n}_A \) and \( \top_A(\mathbf{w}) = \mathbf{w} - \perp_A(\mathbf{w}) \).

**Projection Computation** We compute \( \text{Proj}(A) \) as follows:

1. Given local frame \( \mathbf{R}_A \), on the tangent plane, run MLS with data samples \( \{ \mathbf{x}_B = (u_B, v_B, z_B) | B \in \mathcal{N}^E(A) \} \).

2. Fit \( z \) as a function of \((u, v)\).

3. Let \( \hat{z} \) denote the function evaluated at \((0, 0)^T\), set \( \mathbf{x}_A \leftarrow \mathbf{x}_A - \hat{z}\mathbf{n}_A \).

**Newton Black Films** This phenomenon enter the dynamic system through \( f_{\text{ext}} \).

The Newton Black Films (often referred to as black spots) are extremely thin regions on a soap film, where destructive light interference makes them appear black. We prefix a number of seeders in space that mark nearby \( \mathcal{L} \) particles as \( \mathcal{B} \) particles, whose color will be set to black. The \( \mathcal{B} \) particles receive an additional surface tension force from the \( \mathcal{B} \)-\( \mathcal{L} \) interface as if \( \mathcal{B} \) is a second fluid phase. This force is computed using a VOF approach, where \( \mathcal{E} \) particles estimate the fraction of nearby \( \mathcal{B} \) particles, and then compute surface tension following Akinci et al. [2013].
3.2 MELP

**Rim Surface Tension** The rim surface tension also enter the dynamic system through $f_{\text{ext}}$. The experienced force along the rim’s normal direction $n_{\text{rim}}$ is given by $f_{\sigma,\text{rim}} = 2\sigma + (2\sigma(\pi - 1)R_{\text{rim}})/r_c$ [Bush and Hasha, 2004] where $R_{\text{rim}}$ reflects the thickness of the rim and $r_c$ reflects the size of the thin film. We assume $r_c \gg R_{\text{rim}}$, hence $f_{\sigma}$ is dominated by the first term, so $f_{\sigma,\text{rim}} \approx 2\sigma$. We estimate the rim’s normal direction following Akinci et al. [2013] as $(n_{\text{rim}})_E = r \sum_{E' \in \mathcal{N}_E(E)} a_{E'} \nabla_s W(E, E')$. 
Chapter 4

Multi-MELP

4.1 Overview

The basic MELP method can handle the simulation of manifold thin films (e.g., single bubbles and disks), and in order to simulate complex foam dynamics with evolving topology, we further design *multi-MELP*, a meshless, multi-region tracking mechanism that enables the physically-based interaction among multiple MELP systems. The key innovation here is the soft-handling of the non-manifold junctions. For instance, a triple-junction is not modeled with a singular edge, but with three manifold interfaces tracked by three MELP systems. The coupled dynamics of the junction is computed by a *surface tension sharing* mechanism. Multi-MELP is conveniently extended from MELP, inherits MELP’s capacities in resolving high-quality interfacial flow, develops bubble clusters and foams entirely on-the-fly, and recovers Plateau’s laws accurately.

We define a foam $F = \{L_i\}_{i=0}^{n}$ as a set of $n$ lamella regions. As shown in Figure 4.1, multi-MELP simulates $F$ with $n$ MELP objects each corresponding to one region and running them as subroutines. A multi-MELP simulation step breaks down into the following stages:
4.1 Overview

Figure 4.1: The schematics of a multi-MELP simulation step; $L_1$ through $L_5$ are 5 MELP objects corresponding to the 5 regions.

(1) Multi-Region Tracking: In each region, each $E$ particle identifies other regions it may be coupled with via neighbor searching.

(2) Contact Handling: In each region, each $E$ particle computes non-penetration forces if it is inside another region; and damping forces if it is moving in opposite directions from another region.

(3) MELP Advance: Simulate each region independently using MELP, pause after the dynamics computation is complete.
4.2 Motivating Example

Figure 4.2: The formation of a double-bubble. The top row shows the particle perspective; the bottom row depicts the control volumes with force analysis.

(4) **Surface Tension Sharing:** In each region, each $E$ particle checks each region it couples with; modifies the velocity according to the other region. Resume the paused MELP simulations in (3).

(5) **Material Transfer:** In each region, each $L$ particle probabilistically decides another region to which it is transported, based on the surfactant concentration $\Gamma$ and velocity $u$.

4.2 Motivating Example

Figure 4.2 illustrates a motivating example, where two bubbles collide to form a double-bubble — the simplest form of foam. When two separate lamellae coalesce into a shared surface, topological adaptation occurs as singular points are formed at the top and bottom, trisecting the thin film into three manifold pieces. With our one-sided geometric representation, we turn the topological change into a dynamic one. Topologically, it remains unchanged that there are two manifold inner surfaces.
4.3 Multi-Region Tracking

Dynamically, the inner surfaces of the partition are now constrained by the matching-velocity boundary condition described in Section 2.2. In Figure 4.2, consider $b_A \in L_A$ (blue) and $r_B \in L_B$ (red) previously unattached. After coalescence, they become $g_A$ and $g_B$ (green) which are no longer allowed to move relative to each other. We model their dynamic equivalence via symmetrization. We depict the particles’ control volumes at the bottom of Figure 4.2. For both $g_A$ and $g_B$, we compute their net force as if they each represent the volume $g_A + g_B$. This guarantees that $g_A$ and $g_B$ will move in accordance if they have the same initial velocity. Note that the region $g_A + g_B$ does not need to be determined explicitly. As shown in the free-body diagram, computing the net force of $g_A + g_B$ boils down to computing $f_{st}$ and $f_{air}$ on both $L_A$ and $L_B$, where $f_{air}$ and $f_{st}$ are air pressure and surface tension induced forces corresponding to the first and the second terms of Equation 3.7. These forces can also be continuously evaluated using SPH interpolation, so no explicit particle pairing is needed. This procedure is what we refer to as surface tension sharing.

4.3 Multi-Region Tracking

For an $E$ particle $P \in L_K \in F$, we compile a list of regions with which it is coupled. For each $L_S \neq L_K$, we let $\mathcal{N} = \mathcal{N}^{L_S}(P)$, which is the set of $E$ particles in $L_S$ found within the neighborhood of $P$. If $|\mathcal{N}| = 0$ then $L_K$ is clearly not coupled with $L_S$. Otherwise, we compute the sum of area $\hat{a} = \sum_{E \in \mathcal{N}} a_E$. This expression gauges the amount of area of $L_S$ that the neighborhood of $P$ encircles. We then compute the same sum of area $\hat{\alpha} = \sum_{E \in \mathcal{N}^{L_K}(P)} a_E$ in $L_K$. If $\hat{a} \ll \hat{\alpha}$, then $P$ is relatively far away from $L_S$; if $\hat{a} \approx \hat{\alpha}$, then $a$ is in between $L_K$ and $L_S$. We then compute a coupling score $\gamma_{P,S} = \min(1, \frac{\hat{\alpha}}{\hat{a}})$, and store the tuple $(S, \gamma_{P,S})$ for $P$. 

29
4.4 Collision Handling

Handling the collision of multiple bubbles entails the treatment of 1) non-penetration, and 2) damping. Consider an $E$ particle $P \in L_K$. For each tuple $(S, \gamma_{P,S})$ it has stored, we penalize if $P$ is inside $L_S$, which can be detected if $\text{Proj}_{LS}(P) - P$ is outward-pointing to $L_S$. In that case, a non-penetration force is computed as $\theta_1 \cdot (\text{Proj}_{LS}(P) - P)$ where $\theta_1$ is the penalty strength. For damping, we first compute an average $E$ velocity of all nearby regions weighted by $\gamma$, as:

$$u^E_{\text{avg}} = \frac{u^E_P + \sum_{(S,\gamma_{P,S})} \gamma_{P,S} u^E_{\text{Proj}_{LS}(P)}}{1 + \sum_{(S,\gamma_{P,S})} \gamma_{P,S}}. \quad (4.1)$$

Here, $u^E_{\text{Proj}_{LS}(P)}$ is the SPH interpolation of the $E$ velocity on region $L_S$ at position $\text{Proj}_{LS}(P)$. Then we damp $u^E_P$ with $u^E_P = (1 - \theta_2)u^E_P + \theta_2 u^E_{\text{avg}}$ where $\theta_2 \in (0, 1)$ is the damping strength.

4.5 Surface Tension Sharing

Given an $E$ particle $P \in L_K$, if $P$ is not coupled with any other region, then $f_{\text{net}} = 2f_{\text{st},L_K} + f_{\text{air},L_K}$ as in the lamella setting. Otherwise, we consider each tuple $(S, \gamma_{P,S})$ of $P$ and compute the shared forces as:

$$f_{\text{st},LS} = \frac{\sigma_{\text{Proj}_{LS}(P)} \cdot H_{\text{Proj}_{LS}(P)}}{\rho(\eta_{\text{Proj}_{LS}(P)} + \eta_P)} \cdot n_{\text{Proj}_{LS}(P)}, \quad (4.2)$$

$$f_{\text{air},LS} = \frac{p_{\text{in},LS}}{\rho(\eta_{\text{Proj}_{LS}(P)} + \eta_P)} \cdot n_{\text{Proj}_{LS}(P)}. \quad (4.3)$$

We compute the projection $\text{Proj}_{LS}(P)$ and interpolate $\sigma, \eta, H, n$ on $L_S$ at $\text{Proj}_{LS}(P)$. The term $p_{\text{in},LS}$ is the enclosed air pressure for $L_S$ computed via the ideal gas law.

There is one caveat — in the lamella case, we account for the external interface (the
4.5 Surface Tension Sharing

Figure 4.3: Left: illustration of the force $f_{st, \text{atm}}$ that must be accounted for. Right: material transport is done by transporting $L$ particles directly.

one with the atmosphere) by doubling the surface tension force $f_{st}$. However, when a particle represents a multi-junction, we have only considered interfaces delineated by another lamella particle set. This is illustrated on the left of Figure 4.3. The control volume of $P$ is shadowed in pink. We can compute $f_{st,L_K}$, $f_{air,L_K}$, $f_{st,L_S}$ and $f_{air,L_S}$ as described. However, the surface tension force from the external interface, which is $f_{st, \text{atm}}$ in orange, is not computed. To compute $f_{st, \text{atm}}$, we first compute the pseudo-normal vector $n_{atm}$ with:

$$n_{atm} = \frac{n_P + \sum_{(S, \gamma_{P,S})} \gamma_{P,S} n_{Proj_{L_S}}(P)}{1 + \sum_{(S, \gamma_{P,S})} \gamma_{P,S}},$$

(4.4)

which is a weighted average of the normal vectors of nearby regions. If $|n_{atm}| \ll 1$, then the particle $P$ is deemed an internal point, and $f_{st, \text{atm}} = 0$. Otherwise, on the local frame with normal vector $\frac{n_{atm}}{|n_{atm}|}$, we compute $H_{atm}$ according to Equation 3.5. Then, we compute $f_{st, \text{atm}} = \frac{\sigma_P \cdot H_{atm}}{\rho (\eta_{Proj_{L_S}}(P) + \eta_P)} \cdot n_{atm}$. Finally, we have:

$$f_{net} = f_{st,L_K} + f_{air,L_K} + \sum_{(S, \gamma_{P,S})} (f_{st,L_S} + f_{air,L_S}) + f_{st, \text{atm}}.$$  

(4.5)
4.6 Material Transfer

As observed in Section 2.2, near a multi-junction, the surfacial $\nabla_s$ no longer applies and should be replaced by the volumetric $\nabla$ in Equation 2.5. To simulate its behavior we 1) allow material to be advected to and from regions, and 2) prompt material to flow from regions with high $\Gamma$ to the ones with low $\Gamma$. As depicted on the right of Figure 4.3, we devise a probabilistic scheme to directly migrate $L$ particles from one region to another region, conserving the transported quantities. For each $L$ particle $Q \in L_K$, let $P$ denote its nearest $E$ neighbor. For each tuple $(S, \gamma_{P,S})$ that $P$ stores, we compute two probability scores $C_{1,Ls}$ and $C_{2,Ls}$ for $Q$ as follows:

\[
C_{1,Ls} = \psi_1 \cdot \left(1 - \min(1, \frac{|x_Q + \Delta t u_Q - x_{\text{proj}_{Ls}(Q)}|}{|x_Q - x_{\text{proj}_{Ls}(Q)}|})\right), \tag{4.6}
\]

\[
C_{2,Ls} = \psi_2 \cdot \left(1 - \min(1, \frac{\Gamma_{\text{proj}_{Ls}(P)}}{\Gamma_P})\right), \tag{4.7}
\]

where $\psi_1$ and $\psi_2$ are the transport strength parameters. In computing $C_{1,Ls}$, $x_Q + \Delta t u_Q$ is the position of $Q$ at the next timestep, $x_{\text{proj}_{Ls}(Q)}$ is the nearest point to $Q$ on $L_s$. If $u_Q$ is driving $Q$ towards $L_s$, then this score would be high and vice versa. In computing $C_{2,Ls}$, we compute the ratio of $\Gamma$ between $L_s$ and $L_K$. If $L_s$ has a significantly lower surfactant concentration than $L_K$, a high probability score would ensue. We let $L_G$ denote the region with the largest sum of the two probabilities, let $C_G$ denote that sum, and move $L$ particle $Q$ from $L_K$ to $L_G$ at probability $\min(1, C_G)$. 

32
Chapter 5

Experiments and Results

5.1 Numerical Validation

Plateau Border  Plateau’s laws prescribe that soap films always meet in groups of threes, along edges that create three dihedral angles of $\arccos(-\frac{1}{2}) = 120^\circ$ each. These edges are commonly referred to as the Plateau borders. These Plateau borders then meet in groups of fours, creating angles of $\arccos(-\frac{1}{3}) \approx 109.47^\circ$ each. With the surface tension sharing mechanism, our method accurately recovers both rules. As shown in Figure 5.1, we verify our approach on a double-bubble, a triple-bubble, and a quadruple-bubble. In each setup, the bubbles are initially separated, and the borders are developed dynamically upon contact. As reported in Table 5.1, the measured dihedral angles deviate from the analytical value with $\leq 2\%$ error, while the edge angles deviate with $\leq 5\%$ error, which testifies to the efficacy of our framework.

Curvature of Partition Surface  When two bubbles with different radii — the larger being $R_1$ and the smaller being $R_2$ — form a double-bubble, the smaller bubble will protrude into the larger one, creating a spherical partition surface with radius $R_P = \frac{R_1R_2}{R_1-R_2}$ and curvature $\kappa_P = \frac{1}{R_P}$. This is due to the three-way balance of Young-
5.1 Numerical Validation

<table>
<thead>
<tr>
<th>Plateau Border Testing</th>
<th>Set-up</th>
<th>Double-Bubble</th>
<th>Triple-Bubble</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pairs</td>
<td>Angle</td>
<td>Angle</td>
</tr>
<tr>
<td></td>
<td>1—2</td>
<td>118.62</td>
<td>120.84</td>
</tr>
<tr>
<td></td>
<td>1—3</td>
<td>122.30</td>
<td>118.96</td>
</tr>
<tr>
<td></td>
<td>2—3</td>
<td>119.06</td>
<td>120.20</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>1.167%</td>
<td>0.7%</td>
</tr>
<tr>
<td></td>
<td>1—2</td>
<td>1—3</td>
<td>2—3</td>
</tr>
<tr>
<td></td>
<td>1.917%</td>
<td>0.833%</td>
<td>0.867%</td>
</tr>
<tr>
<td></td>
<td>0.867%</td>
<td>0.167%</td>
<td>0.167%</td>
</tr>
<tr>
<td></td>
<td>2—3</td>
<td>120.20</td>
<td>103.99</td>
</tr>
<tr>
<td></td>
<td>110.74</td>
<td>103.99</td>
<td>110.74</td>
</tr>
<tr>
<td></td>
<td>3.41%</td>
<td>2.64%</td>
<td>2.99%</td>
</tr>
<tr>
<td></td>
<td>1—4</td>
<td>114.74</td>
<td>113.52</td>
</tr>
<tr>
<td></td>
<td>103.99</td>
<td>103.99</td>
<td>110.74</td>
</tr>
<tr>
<td></td>
<td>4.81%</td>
<td>3.70%</td>
<td>1.16%</td>
</tr>
<tr>
<td></td>
<td>2—4</td>
<td>113.52</td>
<td>110.74</td>
</tr>
<tr>
<td></td>
<td>3.70%</td>
<td>1.16%</td>
<td>5.00%</td>
</tr>
<tr>
<td></td>
<td>3—4</td>
<td>110.74</td>
<td>103.99</td>
</tr>
<tr>
<td></td>
<td>5.00%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: Numerical results to validate multi-MELP’s adherence to Plateau’s laws. The pairs are labeled corresponding to Figure 5.1.

<table>
<thead>
<tr>
<th>Curvature $\kappa_P$ of the Partition Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$ (m)</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>0.05</td>
</tr>
<tr>
<td>0.05</td>
</tr>
<tr>
<td>0.05</td>
</tr>
<tr>
<td>0.05</td>
</tr>
<tr>
<td>0.05</td>
</tr>
<tr>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 5.2: The partition surface curvature: analytical values vs. our experimental values. The † symbol represents the ground truth.

Laplace pressures and air pressures, which is handled naturally by our algorithm. We validate with 6 testing setups, where one bubble has a fixed radius $R_1 = 0.05m$, and the other one has a varying radius $R_2$ among $\{0.4R_1, 0.5R_1, 0.6R_1, 0.7R_1, 0.8R_1, 0.9R_1\}$. As showcased in Figure 5.2, the smaller $R_2$ is, the more curved the partition surface becomes. The numerical results are documented in Table 5.2 and plotted on the top-left of Figure 5.3, as they conform well to the analytical values with $\leq 3.5\%$ error.

**Surface Area Minimization** The standard double-bubble, shown in Figure 5.4, is a minimal surface with the steady-state surface area $\hat{a}$ given by: $\hat{a} = 27\pi(\frac{V}{9\epsilon})^{\frac{2}{3}}$ with $\hat{V}$ being the enclosed volume of each region. We verify our method’s ability to recover this with two bubbles of radius 0.05m, initially separated, that are dynamically fused
5.1 Numerical Validation

Figure 5.1: Left: in a double and a triple-bubble, three pieces of lamellae meet at \( \approx 120^\circ \) angles along the border. Right: in a quadruple bubble, 6 partition surfaces (highlighted) form 4 borders (red arrows) meeting at \( \approx 109^\circ \) angles.

into a double-bubble via contact. The initial surface area would be \( \bar{a} = 0.0628 \text{m}^2 \) and the expected final surface area would be \( \hat{a} = 0.0594 \text{m}^2 \). As reflected in Figure 5.3, before the merge occurs at \( t \approx 3 \text{s} \), the total area oscillates around \( \bar{a} \), which then stabilizes to \( \hat{a} \) with periodic oscillation.

**Drainage under Gravity** When a piece of thin film is placed vertically, gravity creates a tendency for the fluid to flow downwards. Near the bottom where fluid amasses, more surfactant will occupy the fluid-air interface, creating a Marangoni acceleration to counteract the gravitational acceleration, eventually reaching an equilibrium. The steady-state thickness profile is derived by [Couder et al., 1989] as \( \eta(z) = \eta_0 e^{- \frac{\rho g \eta_0 z}{2(\sigma_0 - \sigma)}} \) where \( \eta_0 \) is the film thickness when laid flat. Setting \( \eta_0 = 400 \text{nm} \), we verify our method’s correspondence to the analytical solution on the top-right of Figure 5.3. Additionally, the exponential thickness variation creates Newton’s interference fringes with gradually thinning color stripes towards the bottom, which is
5.2 Comparison with Single-Layer Particle Method

As with the previously proposed single-layer particle method [Wang et al., 2021], MELP is also connectivity-free and hence shares the convenience in handling codimension transitions and simulating complex scenes like thin film bursting. However, the separation of tasks with our bi-layer design ensures that the simulation domain is uniformly discretized regardless of the flow dynamics, offering enhanced numerical stability which in turn allows for the adoption of real-world parameters infeasible for the single-layer model.

We demonstrate this with a simple set-up depicted on the right of Figure 5.6: a circular thin film is initialized with spatially-varying thickness $\eta$ (top-left circle), which tends to be evened out via the Marangoni effect. The simulation is carried out for 5 seconds, and the variance of $\eta$ is plotted on the left of Figure 5.5. Using real-
5.2 Comparison with Single-Layer Particle Method

world surface tension parameters, the MELP simulation quickly converges, with its variance approaching the anticipated value of 0. The end result is a spatially-uniform thickness field indicated by the uniform, green color (top-right circle). Using the method of Wang et al. [2021], the variance blows up even with CFL number = 0.033, as the dynamics is too numerically demanding for its explicit SPH solver. This is reflected on the bottom-left circle in which the color/thickness field is highly noisy. To obtain stability, we need to reduce the surface tension parameter to 0.1× the real-world value (bottom-right circle). However, this numerical compromise alters the dynamic characteristics, turning nimble and rapid flows into slowly oscillating compression waves, significantly degrading the visual realism. As depicted in Figure 5.7, both algorithms simulate the same configuration with the same external force. Using real-world parameters, the MELP method responds to the external force acutely, developing multiple vortices that together create an intricate, swirling color palette; the single-layer method, in comparison, offers motion that is visibly more damped, creates coarser flow details, and displays slow, sweeping longitudinal waves uncharacteristic of thin film fluids.

Another reason for MELP’s improved visual performance over the method of Wang et al. [2021] is the dramatically increased number of particles being simulated, at a comparable or lower computational cost. As elaborated in Table 5.3, for Figure 5.7, MELP advects ∼ 700000 $\mathcal{L}$ particles driven by ∼ 7000 $\mathcal{E}$ particles, providing a significant resolution boost over the ∼ 40000 particles in the single-layer model. This can be attributed to the decoupling between the advection resolution and the dynamics resolution. Indeed, a single MELP iteration is still almost 8 times as costly, but with the large step size that it supports, it eventually yields a speed-up of over 40%, as illustrated on the right of Figure 5.5. Consequently, using comparable computational resources, our proposed method outputs simulation sequences with over
5000000 particles against those with at most 170000 particles in Wang et al. [2021].

5.3 Examples

The detailed specifications of all the examples simulated by our proposed system, including the computational resources used, are provided in Table 5.4. Photorealistic rendering is carried out in Houdini with meshes reconstructed from the simulated particles. The color is computed from thin film interference using ColorPy [Kness, 2008] with CIE Standard Illuminant D65. For physical fidelity we limit the CFL number to be strictly less than 1, which does not reflect the numerical capacity of our model. For dynamic scenes involving multiple bubbles, we are limited to CFL number = 0.33 due to the explicit handling of the multi-region interaction.

**Giant Bubble**  As depicted in Figure 5.8, a deformed bubble is initialized by applying displacement mapping to a sphere of radius 0.1m, using 2-octave Perlin noise with frequency = 5 and scale = 0.06. The thickness field is also initialized with Perlin noise, manifesting in the initial, smooth color gradient. The flow is driven by a heat source below the bubble that creates an upward motion. Consequently, the bubble displays a golden tint at the bottom ($\eta \approx 350\text{nm}$) and a green tint at the top ($\eta \approx 500\text{nm}$). An external force later punctures the bubble from the right, causing the thin film to retract under the rim surface tension. The bursting, which takes place in a smaller timescale than the deformation or flow, is simulated at a $15\times$ slow motion, which is handled automatically by our program.

**Deforming Rectangle with Black Spots**  As depicted in Figure 5.9, we initialize a rectangular thin film with length = 0.16m and height = 0.09m. A constant thickness gradient is initially imposed, with thickness $\eta \approx 500\text{nm}$ at the top and $\eta \approx 250\text{nm}$ at
the bottom, which is the slightly perturbed using Perlin noise. Such a configuration creates the Rayleigh-Taylor instability that causes the turbulent flow. An out-of-plane sweeping force is applied to prompt the deformation. Black spots are seeded periodically at the bottom.

**Deforming Half Bubble**  As depicted in Figure 5.10, a half-bubble of radius 0.05m is initialized, with the initial thickness variation generated in the same way as the giant bubble. The flow is driven by a heat source located below the half bubble. The gentle deformation is propelled by a horizontal sweeping wind. Black spots are seeded at the bottom boundary periodically, similar to the rectangle example.

**Bubbles of Different Sizes**  As shown in Figure 5.11, a bubble of radius 0.025m, another one of radius 0.05m, and a half bubble of radius 0.1m are simulated, in order to verify our system’s ability to handle large size differences. The two bubbles are put into contact first, forming a double-bubble, with the smaller one protruding into the larger one. Afterwards, an external acceleration drives the double-bubble into the half bubble. The downwards momentum causes the double-bubble to slide down the half bubble. As it slides down, it also tilts counter-clockwise, which decreases the angle it forms with the half bubble. The sliding motion is counteracted by the surface tension’s tendency to restore 120° angles, and the system gradually settles into an equilibrium.

**Dynamic Reorganization of 4 Bubbles**  It is known that the three-way Plateau border is the only stable equilibrium for multiple thin films to convene. However, unstable equilibriums exist — for instance, when four bubbles meet at a cross shape to create an edge that joins four surfaces with dihedral angles of 90° each. Such an unstable equilibrium should morph into a stable Plateau border given a small
perturbation. With this experiment, we test our system’s ability to recreate this phenomenon. As depicted in Figure 5.12, we initialize four bubbles in a rectangular formation, with initial velocities driving them to the center. Upon contact, they naturally form 4 partition surfaces, meeting along the central edge at 90° angles. However, the surface tension is slightly varied among the four bubbles, causing a small asymmetry in the force balance. Under this perturbation, a new partition surface is gradually pulled out from the initial edge, developing into two Plateau borders with ≈ 120° angles. Once this new configuration stabilizes, we delete one of the partition surfaces to have the right two bubbles merge into a single one, which is later punctured from the top-right. The momentum caused by the thin film retraction is coupled to the dynamics computation of the remaining double-bubble.

**Rayleigh-Taylor Instability on a Double-Bubble** As depicted in Figure 5.13, two bubbles of radius 0.5m are initially separated and aligned vertically. The top one has thickness $\eta \approx 500$nm and the bottom one has $\eta \approx 250$nm, as they are tinted purple and blue reflecting their respective thickness values. With initial velocities towards the center, two bubbles collide and develop into a double-bubble with a shared surface in between. At the same time, material transfer between both bubbles begins. As fluid is transferred from top to bottom under gravity, Rayleigh-Taylor instability is created, and the thinner fluid in the lower region is propelled to the upper one in exchange. Eventually, the bottom region becomes thick and the top region becomes thin, causing the tints to reverse, where the lower region appears purple and the upper one appears blue.

**Foam Mountain** This example puts to test our system’s caliber in stably handling bubble clusters or foams at a much larger scale. As depicted in Figure 5.14, three hundred bubbles, whose radii are randomly selected between 0.008m to 0.012m, are
5.3 Examples

<table>
<thead>
<tr>
<th>Name</th>
<th>Real Param.</th>
<th>CFL #</th>
<th># Particles</th>
<th>t/itr.</th>
<th>itr./frame</th>
<th>t./frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wang et al. [2021] Equilibrium</td>
<td>✓</td>
<td>0.033</td>
<td>4200</td>
<td>0.07</td>
<td>55.9</td>
<td>3.91</td>
</tr>
<tr>
<td>Wang et al. [2021] Equilibrium</td>
<td>✗</td>
<td>0.1</td>
<td>4200</td>
<td>0.07</td>
<td>3.8</td>
<td>0.27</td>
</tr>
<tr>
<td>MELP Equilibrium</td>
<td>✓</td>
<td>0.99</td>
<td>16800 + 1050 E</td>
<td>0.33</td>
<td>1.06</td>
<td>0.35</td>
</tr>
<tr>
<td>Wang et al. [2021] Flow</td>
<td>✗</td>
<td>0.1</td>
<td>40000</td>
<td>0.47</td>
<td>24.91</td>
<td>11.71</td>
</tr>
<tr>
<td>MELP Flow</td>
<td>✓</td>
<td>0.99</td>
<td>693900 L + 6900 E</td>
<td>3.1</td>
<td>2.25</td>
<td>6.98</td>
</tr>
</tbody>
</table>

Table 5.3: Performance comparison between the single-layer method in Wang et al. [2021] and MELP.

poured down from five “faucets” of bubbles located above. Bubbles that land within the container gradually build up a honeycomb structure — a foam mountain. Bubbles that collide with the container are automatically deleted. Once the bubbles have stopped pouring, and the cluster stabilized, we sporadically delete bubbles at random. The remaining bubbles reorganize by contracting inwards to fill the gaps.

Cyclones on 13 Bubbles  As shown in Figure 5.15, 13 bubbles with radii from 0.0435m to 0.06m, and thickness from 400nm to 600nm are initialized. Their centroids are initialized by FCC packing with uniform random offsets. Under initial velocities towards the center, these bubbles come into contact and spontaneously settle into stable Plateau borders. A heat source is deployed at the bottom, causing fluid to flow from the bottom to the top, which manifests in the stratification of color, with the thinnest region at the bottom being dark gold (≈ 180nm) and the thickest region at the top being purple (≈ 550nm). This heat-driven convection gradually develops into “cyclones” on the bubble surfaces. The bottom row of Figure 5.15 documents the reorganization process — one partition surface between two bubbles is deleted, creating a bubble that is larger than all the others. The bubbles around it reorganize and merge to achieve a new equilibrium.
### 5.3 Examples

<table>
<thead>
<tr>
<th>Name</th>
<th>CFL number</th>
<th>Number of $E$</th>
<th>Number of $L$</th>
<th>Ratio</th>
<th>Time/Iter (s)</th>
<th>Using</th>
<th>Depicted In</th>
</tr>
</thead>
<tbody>
<tr>
<td>Giant Bubble</td>
<td>0.99</td>
<td>163842</td>
<td>2621442</td>
<td>1:16</td>
<td>30.2</td>
<td>A</td>
<td>Figure 5.8</td>
</tr>
<tr>
<td>Deforming Rectangle</td>
<td>0.99</td>
<td>159367</td>
<td>2557467</td>
<td>1:16</td>
<td>24.6</td>
<td>B</td>
<td>Figure 5.9</td>
</tr>
<tr>
<td>Half Bubble</td>
<td>0.99</td>
<td>81921</td>
<td>5242884</td>
<td>1:64</td>
<td>63.3</td>
<td>B</td>
<td>Figure 5.10</td>
</tr>
<tr>
<td>Two Bubbles</td>
<td>0.99</td>
<td>20480</td>
<td>0</td>
<td></td>
<td>0.72</td>
<td>A</td>
<td>Figure 5.4</td>
</tr>
<tr>
<td>Different Size</td>
<td>0.33</td>
<td>33280</td>
<td>0</td>
<td></td>
<td>1.32</td>
<td>A</td>
<td>Figure 5.11</td>
</tr>
<tr>
<td>Four Bubbles</td>
<td>0.99</td>
<td>40960</td>
<td>0</td>
<td></td>
<td>1.45</td>
<td>A</td>
<td>Figure 5.12</td>
</tr>
<tr>
<td>R-T Instability</td>
<td>0.99</td>
<td>81924</td>
<td>4772266</td>
<td>1:58</td>
<td>55.7</td>
<td>A</td>
<td>Figure 5.13</td>
</tr>
<tr>
<td>Foam Mountain</td>
<td>0.33</td>
<td>192311</td>
<td>0</td>
<td></td>
<td>9.1</td>
<td>A</td>
<td>Figure 5.14</td>
</tr>
<tr>
<td>13 Bubbles</td>
<td>0.99</td>
<td>133146</td>
<td>2129946</td>
<td>1:16</td>
<td>13.3</td>
<td>A</td>
<td>Figure 5.15</td>
</tr>
</tbody>
</table>

Table 5.4: The catalog of experiments with the MELP method. [A] represents a computer with AMD Ryzen(TM) ThreadRipper 3990X, and [B] represents a computer with Intel(R) Core(TM) i9-9980XE.
5.3 Examples

Figure 5.3: Top-left: curvatures of the partition surface for double-bubbles of different size ratios compared to analytical values. Top-right: thickness profile under gravity compared to analytical values. Bottom: the evolving surface area of two bubbles as they merge into a double-bubble.
5.3 Examples

Figure 5.4: Formation and evolution of a double-bubble. Top: photorealistic rendering, bottom: particle view.

Figure 5.5: Comparison with Wang et al. [2021]. Left: test of convergence to equilibrium thickness. Right: comparison of computational cost.
5.3 Examples

Figure 5.6: Left: Newton’s interference fringes under gravity. Right: comparison with Wang et al. [2021]: top-left: initial set-up; top-right: converged result of MELP; bottom-left: diverged result of Wang et al. [2021], bottom-right: converged result of Wang et al. [2021] with reduced parameters.

Figure 5.7: Comparison of the simulated flow quality of our proposed MELP method (top) and Wang et al. [2021] (bottom).
Figure 5.8: The flow, deformation, and bursting of a giant bubble, similar to the experiment done in Wang et al. [2021] Figure 5.

Figure 5.9: Different frames of a deforming rectangular film with black spots.
5.3 Examples

Figure 5.10: Different frames of a deforming half bubble with black spots.

Figure 5.11: Interaction among bubbles of different sizes, showcasing our system's ability to restore the equilibrium states.

Figure 5.12: Four bubbles merge, reorganize from an unstable equilibrium to a stable one, and eventually disintegrate.
5.3 Examples

Figure 5.13: The dynamic formation of a double-bubble with intricate flow patterns, simulated by our proposed method. With the appropriate treatment of surface tension near the junction, two bubbles spontaneously settle into meeting angles of $\approx 120^\circ$, recovering what is known as the Plateau border.

Figure 5.14: 300 bubbles falling into a container, forming a foam mountain.
Figure 5.15: 13 bubbles merging together, with a heat source at the bottom creating “cyclones” on the surfaces of the bubbles.
Chapter 6

Conclusion

In this thesis, we introduce a novel, mesh-free framework to tackle the multifaceted computational challenges in simulating incompressible flow on dynamically deforming thin films and topologically adapting foams. Using two collaborative particle sets, we devise a concise and coherent numerical framework to expressively discretize thin film volumes, robustly track moving interfaces, efficiently solve the dynamic PDEs and conveniently perform topological evolutions. Our method marries traditional particle simulation techniques like SPH with ideas from the level-set theory, arbitrary Lagrangian-Eulerian simulation, and Particle-In-Cell methods, to create a stable, efficient, and versatile simulation system yielding state-of-the-art realism. Furthermore, we propose an innovative perspective for modeling non-manifold junctions, featuring one-sided representations of the inner surfaces, which is simple to implement and predictable to run, creating results both visually plausible and numerically accurate.

The main limitations of our approach are as follows: 1) the coupling of multiple regions near the junctions is carried out via explicit force computations, which limits the step size and can cause instability in aggressive scenarios, 2) the modeling of the flow dynamics on the partition surfaces and multi-junctions can be improved with more accurate physics e.g. incorporating the influence of the junction curvature,
Conclusion

3) the coupling between the ideal gas equation and the thin film fluid equations is not momentum-conserving, which can cause drifting artifacts, and 4) the current framework does not support the physical interaction between thin films and solids.

Our proposed method opens up new possibilities in tackling dynamic problems on topologically evolving, non-manifold geometries. Example applications include spider webs, cosmic webs, porous materials, metamaterial structures, etc. One immediate future challenge is to apply the MELP framework to simulate codimension-two-dominant physical systems featuring filament structures and their junctions. The coupling between codimension-one and codimension-two structures (e.g. the interaction between rims and thin films), is another important problem that can be addressed in our future work. We also plan to incorporate implicit representations such as level-sets into MELP, so as to create flexible moving-surface solvers for handling large-scale, topologically-complicated phenomena.
Bibliography


F. Da, C. Batty, C. Wojtan, and E. Grinspun. Double bubbles sans toil and trou-
ble: Discrete circulation-preserving vortex sheets for soap films and foams. ACM


F. Gibou, R. Fedkiw, and S. Osher. A review of level-set methods and some recent


W. Huang, J. Iseringhausen, T. Kneiphof, Z. Qu, C. Jiang, and M. B. Hullin.
Chemomechanical simulation of soap film flow on spherical bubbles. ACM
http://dx.doi.org/10.1145/3386569.3392094.

D. A. B. Hyde, S. W. Gagniere, A. Marquez-Razon, and J. Teran. An implicit updated
Lagrangian formulation for liquids with large surface energy. ACM TOG, 39(4),

M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner. Implicit in-
compressible sph. IEEE transactions on visualization and computer graphics, 20

S. Ishida, M. Yamamoto, R. Ando, and T. Hachisuka. A hyperbolic geometric flow
for evolving films and foams. ACM Transactions on Graphics (TOG), 36(6):1–11,
2017.


