A moving Eulerian-Lagrangian particle method for thin film and foam simulation

Article *in* ACM Transactions on Graphics · July 2022

DOI: 10.1145/3528223.3530174

6 authors, including:

Mengdi Wang
Peking University
6 PUBLICATIONS **37** CITATIONS
SEE PROFILE

Shiying Xiong
Dartmouth College
30 PUBLICATIONS **194** CITATIONS
SEE PROFILE

Bo Zhu
Dartmouth College
40 PUBLICATIONS **495** CITATIONS
SEE PROFILE

A Moving Eulerian-Lagrangian Particle Method for Thin Film and Foam Simulation

YITONG DENG, MENGDI WANG, XIANGXIN KONG, SHIYING XIONG, ZANGYUEYANG XIAN, and BO ZHU, Dartmouth College, USA

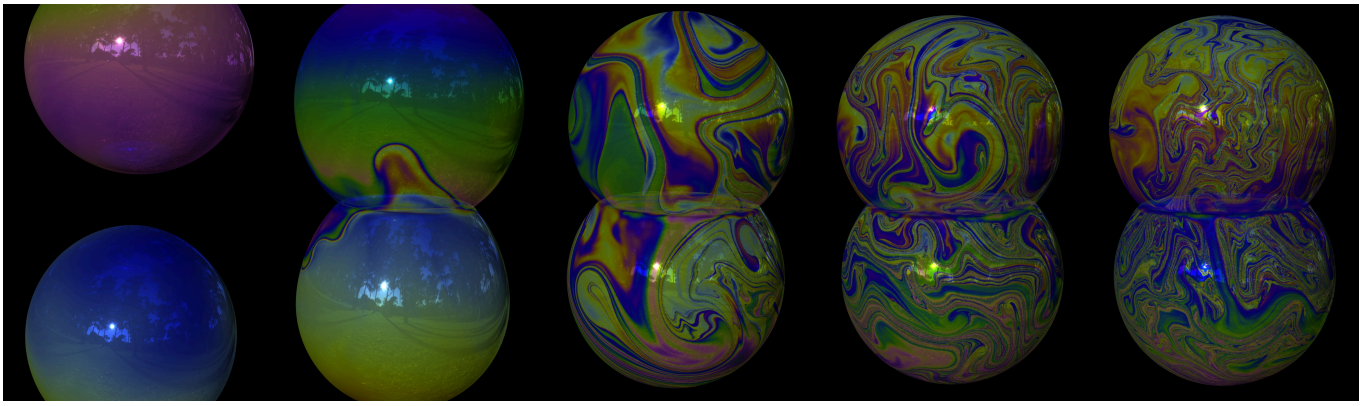

Fig. 1. The dynamic formation of a double-bubble with intricate flow patterns, simulated by our proposed method. With the appropriate treatment of surface tension near the junction, two bubbles spontaneously settle into meeting angles of $\approx 120^\circ$, recovering what is known as the Plateau border.

We present the Moving Eulerian-Lagrangian Particles (MELP), a novel mesh-free method for simulating incompressible fluid on thin films and foams. Employing a bi-layer particle structure, MELP jointly simulates detailed, vigorous flow and large surface deformation at high stability and efficiency. In addition, we design multi-MELP: a mechanism that facilitates the physically-based interaction between multiple MELP systems, to simulate bubble clusters and foams with non-manifold topological evolution. We showcase the efficacy of our method with a broad range of challenging thin film phenomena, including the Rayleigh-Taylor instability across double-bubbles, foam fragmentation with rim surface tension, recovery of the Plateau borders, Newton black films, as well as cyclones on bubble clusters.

CCS Concepts: • **Computing methodologies** → **Modeling and simulation**.

Additional Key Words and Phrases: thin film dynamics, surface tension flow, Eulerian-Lagrangian methods, moving surface

ACM Reference Format:
Yitong Deng, Mengdi Wang, Xiangxin Kong, Shiying Xiong, Zangyueyang Xian, and Bo Zhu. 2022. A Moving Eulerian-Lagrangian Particle Method for Thin Film and Foam Simulation. *ACM Trans. Graph.* 41, 4, Article 154 (July 2022), 17 pages. https://doi.org/10.1145/3528223.3530174

Authors' address: Yitong Deng, yitong.deng.gr@dartmouth.edu; Mengdi Wang, mengdi.wang.gr@dartmouth.edu; Xiangxin Kong, xiangxin.kong.gr@dartmouth.edu; Shiying Xiong, shiying.xiong@dartmouth.edu; Zangyueyang Xian, f0040nb@dartmouth.edu; Bo Zhu, bo.zhu@dartmouth.edu, Dartmouth College, Hanover, NH, USA.

0730-0301/2022/7-ART154 $15.00
https://doi.org/10.1145/3528223.3530174

1 INTRODUCTION

Fluid thin films: soap lamellae, bubble clusters, and foam networks, exhibit remarkably complex geometry and dynamics due to the coupled interaction between the interfacial flow, surface deformation, and topological evolution. On the level of lamellae, these fluid systems are intriguing for their sophisticated, swirling color patterns jointly created by the turbulent flow dynamics and light interference [Belcour and Barla 2017; Glassner 2000; Iwasaki et al. 2004; Jaszkowski and Rzeszut 2003; Smits and Meyer 1992]. On the level of foams, surface tension evolves these liquid lamellae into topologically complicated networks with meticulously-connected, non-manifold borders according to Plateau's laws [Kraynik et al. 2004; Rosen and Kunjappu 2012; Saye and Sethian 2013; Thomas et al. 2015; Weaire and Phelan 1996]. The interleaving complexities between both levels render the full-scale simulation of incompressible fluid on thin films and foams particularly challenging.

The challenge of simulating thin films and foams can be divided into three aspects — the 1) *turbulent flow*, 2) *deforming geometry*, and 3) *evolving, non-manifold topology*. Tackling these demanding tasks, a vast literature in computer graphics and computational physics has been devoted to devising effective geometric data structures and PDE solvers to capture the vivid flow details on dynamic membranes. For instance, researchers have constructed numerical algorithms to generate highly detailed surface flow on fixed spherical domains [Hill and Henderson 2016; Huang et al. 2020; Yang et al. 2019], to model the topological evolution of foams using dynamic meshes [Ishida et al. 2020, 2017], and to simulate bubble deformation and bursting with particles [Wang et al. 2020, 2021]. 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.

Responding to these multifaceted challenges, 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. Our system is powered by two core innovations. First, 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 be deployed at larger amounts to track out more sophisticated and accurate flow patterns.

Secondly, we design multi-MELP, a meshless, multi-region tracking mechanism that enables the physically-based interaction among multiple MELP systems to simulate complex foam dynamics with evolving topology. The key innovation 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 summarize our main contributions as:

(1) A novel, bi-layer particle representation (MELP) for solving dynamic PDEs on moving thin films that achieves state-of-the-art fineness and visual realism.
(2) A novel, mesh-free approach to modeling foam junctions (multi-MELP) that enables dynamic, on-the-fly foam formation according to Plateau's laws.
(3) The versatile simulation of complex phenomena, *e.g.* large foam clusters with hundreds of regions, Rayleigh-Taylor flow across Plateau borders, as well as Newton black films.

2 RELATED WORKS

Film simulation. The simulation of thin film lamellae (*e.g.* soap bubbles or membranes with boundaries) has drawn great research interest across physics and graphics. On one hand, researchers have derived reduced governing equations [Chomaz 2001; Couder et al. 1989] and PDE solvers [Saye and Sethian 2013] to model the thin film evolution on spherical-coordinate grids [Hill and Henderson 2016; Huang et al. 2020], level-sets [Zheng et al. 2009], meshes [Da et al. 2015, 2016; Ishida et al. 2020, 2017; Saye and Sethian 2013, 2016; Wang and Chern 2021; Zhu et al. 2014], particles [Wang et al. 2020], or using a hybridization of meshes and points [Chen et al. 2021; Hyde et al. 2020]. On the other hand, the iridescent color patterns of thin films can be physically computed using thin film interference [Glassner 2000; Iwasaki et al. 2004; Jaszkowski and Rzeszut 2003] to create convincing visual effects. One central challenge for thin film (lamella) simulation is to jointly simulate the interfacial flow and the membrane deformation. In particular, our work is motivated by Ishida et al. [2020, 2017] and Wang et al. [2021], both of which unify thin film deformation with thickness evolution.

Foam simulation. 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]. In this paper, we focus on simulating wet foam structures to simultaneously resolve the flow details and topological evolution.

Eulerian-Lagrangian methods. The MELP method is built upon multiple lines of previous work in hybrid Eulerian-Lagrangian simulation. Our design is motivated by PIC/FLIP [Zhu and Bridson 2005], MPM [Stomakhin et al. 2013], and vortex methods [Koumoutsakos 2005] among others. Unlike traditional methods, whose Eulerian components remain static in the world space, both the Eulerian and Lagrangian components in MELP move based on physical velocities to track a dynamic, codimensional fluid domain. In this regard, MELP is also motivated by arbitrary Lagrangian-Eulerian (ALE) methods [Hirt et al. 1974; Sahu et al. 2020], along with other moving Eulerian methods such as translational grids [Cohen et al. 2010; English et al. 2013] and Eulerian solids [Levin et al. 2011].

Moving surface. We build the mathematical foundation of MELP upon the theory of moving surface calculus [Afas 2018; Grinfeld 2013; Grinfeld et al. 2012], which decouples a dynamic interface's normal and tangential motions with a secondary reference frame. This line of research encompasses a broad range of physical applications ranging from basic fluid systems [Grinfeld 2010a,b,c; Grinfeld et al. 2009; Morgenroth et al. 2020], fluid and electron bubbles [Grinfeld 2009; Svintradze 2019], burning [You and Yang 2020], to magnetostatic systems [Grinfeld and Grinfeld 2017]. The MELP method employs the normal coordinate system originated from the moving surface calculus to advance the Eulerian particles.

3 CONTINUUM MODEL

3.1 Geometry

Lamellae. As depicted on the left of Figure 2, 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 $\mathbf{n} : S_B \to S^2$ of *outward-pointing*, unit normal vectors to be defined. For a disk,

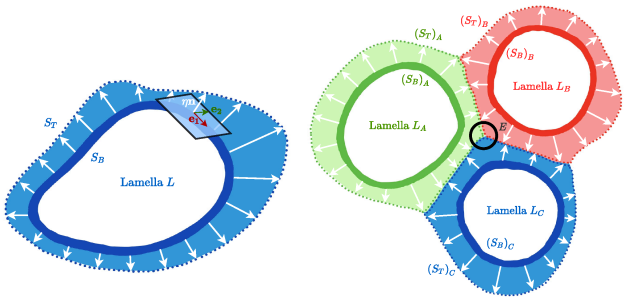

Fig. 2. An illustration of our *one-sided* geometric model. Left: a thin film lamella with thickness and local frames. Right: a triple-junction represented by three lamellae without directly modeling the singularity at E.

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 $\{\mathbf{e}_1(p), \mathbf{e}_2(p)\}$. We then define a field of local frames $\mathbf{R} : S_B \to SO3$, as $\mathbf{R}(p) = \{\mathbf{e}_1(p), \mathbf{e}_2(p), \mathbf{n}(p)\}$ with coordinates (u, v, z). Additionally, we define a field of mean curvatures $H : S_B \to \mathbb{R}$, a field of metric tensors $g : S_B \to \mathbb{R}^{2\times2}$, and a field of thickness $\eta : S_B \to \mathbb{R}$.

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 $f : S_B \to \mathbb{R}^3$ with $f(p) = p + \eta(p)\mathbf{n}(p)$. The geometric quantities, *e.g.* the mean curvature, are 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 as 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, 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.

3.2 Dynamics

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

$$\begin{cases} \rho \dfrac{D\boldsymbol{u}}{Dt} = -\nabla p + \boldsymbol{f}_\sigma + \boldsymbol{f}_{\text{ext}}, \\ \nabla \cdot \boldsymbol{u} = 0, \end{cases} \tag{1}$$

where ρ denotes the density, p the pressure, $\boldsymbol{f}_\sigma$ the surface tension force per unit volume, and $\boldsymbol{f}_{\text{ext}}$ the external forces, *e.g.* gravity.

Surface tension. The surface tension force $\boldsymbol{f}_\sigma$ in Equation 1 is computed as

$$\boldsymbol{f}_\sigma = (\sigma H \mathbf{n} + \nabla_s \sigma) \cdot \delta_I, \tag{2}$$

with σ denoting the surface tension coefficient, H and $\mathbf{n}$ the mean curvature and normal vector on the interfaces, ∇_s the surface gradient operator, and δ_I the Dirac delta function that is non-zero only on the interfaces. The first term: $\sigma H\mathbf{n}$ 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 σ relates to the surfactant concentration Γ as $\sigma = \sigma_0 - \bar{R}T\Gamma$, where σ_0 is the surface tension for pure water, $\bar{R}$ the ideal gas constant, and T the temperature [Xu et al. 2006].

Lamellae. Following Ishida *et al.* [2020], we separate Equation 1 into its normal and tangential components as:

$$\begin{cases} \rho \dfrac{D\boldsymbol{u}^\perp}{Dt} = -\dfrac{\partial p}{\partial z}\mathbf{n} + \delta_I \sigma H \mathbf{n} + \boldsymbol{f}_{\text{ext}}^\perp, \\ \rho \dfrac{D\boldsymbol{u}^\top}{Dt} = \delta_I \nabla_s \sigma + \boldsymbol{f}_{\text{ext}}^\top. \end{cases} \tag{3}$$

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

$$\frac{\partial p}{\partial z} = \delta_B \cdot (\tilde{p} - p_{\text{in}}) + \delta_T \cdot (p_{\text{out}} - \tilde{p}), \tag{4}$$

with δ_B and δ_T being Dirac delta functions that represent S_B and S_T respectively, and satisfying $\delta_B + \delta_T = \delta_I$; p_{in} and p_{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 Γ and membrane thickness η, we rewrite Equation 3 to obtain the full dynamic model:

$$\begin{cases} \dfrac{D\boldsymbol{u}^\perp}{Dt} = -\dfrac{1}{\rho}\dfrac{\partial p}{\partial z}\mathbf{n} + \dfrac{\delta_I(\sigma_0 - \bar{R}T\Gamma)H}{\rho}\mathbf{n} + \dfrac{1}{\rho}\boldsymbol{f}_{\text{ext}}^\perp, \\ \dfrac{D\boldsymbol{u}^\top}{Dt} = -\dfrac{2\bar{R}T}{\rho\eta}\nabla_s\Gamma + \dfrac{1}{\rho}\boldsymbol{f}_{\text{ext}}^\top, \\ \dfrac{D\Gamma}{Dt} = -\Gamma\nabla_s \cdot \boldsymbol{u}, \\ \dfrac{D\eta}{Dt} = -\eta\nabla_s \cdot \boldsymbol{u}. \end{cases} \tag{5}$$

Junctions. As described in Subsection 3.1, 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 ∇_s operator shall be replaced by ∇ in Equation 5. Due to our method's codimensional nature, we opt not to evaluate volumetric derivatives explicitly, but rather simulate its behaviors using particles.

4 MELP

We propose the Moving Eulerian-Lagrangian Particle (MELP) method for dynamic interface tracking and surface flow simulation. As shown in Figure 3, 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 representing the thin film geometry and in solving the dynamic equations.

Geometry. We have defined a lamella as a 5-tuple $(S_B, \mathbf{R}, H, g, \eta)$. As shown on the left of Figure 3, S_B is uniformly discretized by the $\mathcal{E}$ particles, each of which controls some area a. Given the point set, we approximate the local frame $\mathbf{R}$, mean curvature H, and metric tensor g following [Wang et al. 2020]. The remaining variable η will be determined by the distribution of $\mathcal{L}$ particles. As displayed on the left of Figure 3, due to the incompressibility constraint, the denser the $\mathcal{L}$ particles are, the larger η becomes — an idea leveraged by previous works [Solenthaler 2011; Wang et al. 2021]. Each $\mathcal{E}$ particle then controls a fluid column with volume $V = a \cdot \eta$.

Dynamics. In solving the dynamic equations, the $\mathcal{E}$ and $\mathcal{L}$ 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 $\mathcal{L}$ particles. The projection term is solved on the sparser, uniformly-distributed $\mathcal{E}$ particles using Implicit Incompressible SPH (IISPH) [Ihmsen et al. 2013]. The material and momentum transfer between $\mathcal{E}$ and $\mathcal{L}$ 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, since the $\mathcal{E}$ particles are purely geometric, we can let them advect with the normal component of the material velocity without affecting the dynamics. Figure 4 depicts the temporal evolution of $\mathcal{E}$ advecting with the normal velocity (red dots) and $\mathcal{L}$ advecting with the full velocity (blue circles) in a steady spiral flow with velocity $\boldsymbol{u} = (-xy, x^2)$. It can be observed that both $\mathcal{E}$ and $\mathcal{L}$ remain on the same interface, with $\mathcal{E}$ enjoying a considerably more uniform distribution. We further incorporate the arbitrary Lagrangian-Eulerian idea, in which an artificial tangential velocity is granted on $\mathcal{E}$ 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].

4.1 Algorithm

The basic MELP procedure is illustrated in Figure 5 with labels corresponding to the following stages:

(1) ***$\mathcal{L}$2$\mathcal{E}$ Transfer:*** Transfer mass m, surfactant c, volume V, and momentum $\boldsymbol{p}$ from $\mathcal{L}$ to $\mathcal{E}$ (Algorithm 1).

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

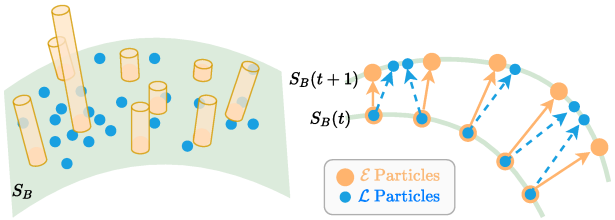

Fig. 3. Left: the local density of $\mathcal{L}$ determines the thickness. Right: $\mathcal{L}$ advect with the full velocity, $\mathcal{E}$ advect with the normal velocity.

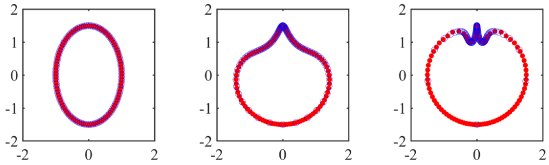

Fig. 4. The temporal evolution of the $\mathcal{E}$ particles (red dots) and $\mathcal{L}$ particles (blue circles) in a steady spiral flow.

Table 1. A list of symbols and expressions for the MELP algorithm.

Symbol	Meaning
W	3D kernel function
$\overline{W}$	2D kernel function
$\boldsymbol{x}_A$	position of particle A
$W(A,B)$	$W(\boldsymbol{x}_A - \boldsymbol{x}_B)$
$\perp_A(\boldsymbol{w})$	project $\boldsymbol{w}$ to the normal direction of A
$\top_A(\boldsymbol{w})$	project $\boldsymbol{w}$ to the tangent plane of A
$\overline{W}(A,B)$	$\overline{W}(\top_A(\boldsymbol{x}_A - \boldsymbol{x}_B))$
r	kernel support radius
$\mathcal{N}^{\mathcal{E}}(A)$	all $\mathcal{E}$ particles within radius r from $\boldsymbol{x}_A$
$\mathcal{N}^{\mathcal{L}}(A)$	all $\mathcal{L}$ particles within radius r from $\boldsymbol{x}_A$
Proj(A)	project $\boldsymbol{x}_A$ onto S_B

(3) ***Dynamics Computation:*** Solve Equation 5 on $\mathcal{E}$ in the normal and tangential directions; update velocity $\boldsymbol{u}$ (Algorithm 3).

(4) ***$\mathcal{E}$2$\mathcal{L}$ Transfer:*** Each $\mathcal{L}$ particle interpolates $\boldsymbol{u}$ 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 define relevant symbols and expressions in Table 1.

4.2 $\mathcal{L}$2$\mathcal{E}$ Transfer

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

$$q_E = \sum_{L \in \mathcal{N}^{\mathcal{L}}(E)} \hat{W}(E,L) \cdot q_L, \tag{6}$$

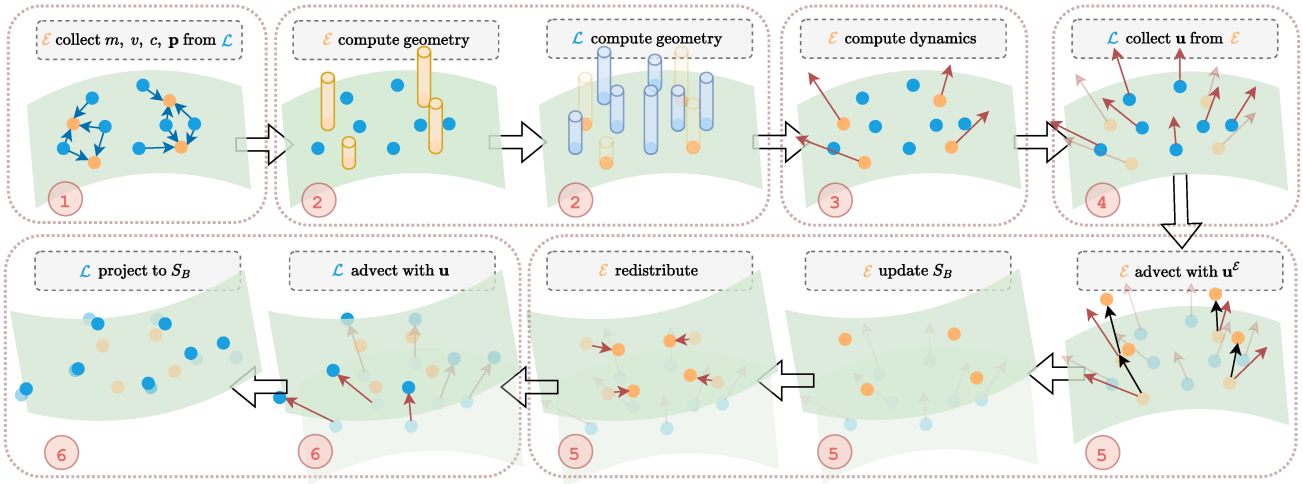

Fig. 5. The computation workflow of a single simulation step in our proposed MELP framework.

where $\hat{W}$ has the *partition of unity* quality $\sum_{E\in\mathcal{N}^{\mathcal{E}}(L)}\hat{W}(E,L)=1$. Given an SPH kernel W, we define $\hat{W}(E,L)=W(E,L)/\alpha_L$, where $\alpha_L=\sum_{E\in\mathcal{N}^{\mathcal{E}}(L)}W(E,L)$. Then, we transfer mass m, surfactant c, volume V, and momentum $\boldsymbol{p}$ according to Equation 6. Furthermore, we construct an *affine momentum* $\hat{\boldsymbol{p}}$ for APIC:

$$\hat{\boldsymbol{p}}_E=\sum_{L\in\mathcal{N}^{\mathcal{L}}(E)}\hat{W}(E,L)\cdot[\mathbf{B}_L(\mathbf{D}_L)^{-1}\top_L(\boldsymbol{x}_E-\boldsymbol{x}_L)], \tag{7}$$

where $\mathbf{B}$ and $\mathbf{D}$ are the affine state and inertia-like tensor carried by the $\mathcal{L}$ particles. We reconstruct velocity $\boldsymbol{u}_E$ with $\boldsymbol{u}_E=\frac{\boldsymbol{p}_E+\hat{\boldsymbol{p}}_E}{m_E}$, and split it into its normal and tangential components as $\boldsymbol{u}_E^{\perp}=\perp_E(\boldsymbol{u}_E)$, $\boldsymbol{u}_E^{\top}=\top_E(\boldsymbol{u}_E)$, which are to be evolved separately in Section 4.4.

4.3 Geometry Computation

For each $L\in\mathcal{L}$, we evolve thickness η_L according to Equation 5 after temporal discretization:

$$\eta_{L(t)}=\eta_{L(t-1)}-\Delta t\eta_{L(t-1)}\nabla_s\cdot\boldsymbol{u}. \tag{8}$$

For each $E\in\mathcal{E}$, we compute number density δ_E and area a_E as:

$$\delta_E=\sum_{E'\in\mathcal{N}^{\mathcal{E}}(E)}\overline{W}(E,E'),\ a_E=\frac{1}{\delta_E}. \tag{9}$$

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

For each $E\in\mathcal{E}$, a neighboring particle $E'\in\mathcal{N}^{\mathcal{E}}(E)$ has coordinates in E's local frame: $(u,v,z)=((\boldsymbol{x}_{E'}-\boldsymbol{x}_E)\cdot\mathbf{e}_1,(\boldsymbol{x}_{E'}-\boldsymbol{x}_E)\cdot\mathbf{e}_2,(\boldsymbol{x}_{E'}-\boldsymbol{x}_E)\cdot\mathbf{n})$. With ∇_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:

$$\begin{cases}H_E &= -\frac{1}{2}\nabla_E\cdot\left(\frac{\nabla z}{\sqrt{1+\nabla z^2}}\right)\approx-\frac{1}{2}{\nabla_E}^2 z,\\ g_E &= \begin{bmatrix}1+\left(\frac{\partial z}{\partial u}\right)^2, \frac{\partial z}{\partial u}\frac{\partial z}{\partial v}\\ \frac{\partial z}{\partial u}\frac{\partial z}{\partial v}, 1+\left(\frac{\partial z}{\partial v}\right)^2\end{bmatrix}.\end{cases} \tag{10}$$

Algorithm 1 $\mathcal{L}2\mathcal{E}$ transfer

1: **for** each particle $L\in\mathcal{L}$ **do**
2: Compute $\alpha_L=\sum_{E\in\mathcal{N}^{\mathcal{E}}(L)}W(E,L)$.
3: **end for**
4: **for** each particle $E\in\mathcal{E}$ **do**
5: **for** $q\in\{m,c,V,\boldsymbol{p}\}$ **do**
6: Compute q_E according to Equation 6
7: **end for**
8: Compute $\hat{\boldsymbol{p}}_E$ according to Equation 7
9: $\boldsymbol{u}_E\leftarrow\frac{\boldsymbol{p}_E+\hat{\boldsymbol{p}}_E}{m_E}$, $\boldsymbol{u}_E^{\perp}\leftarrow\perp_E(\boldsymbol{u}_E)$, $\boldsymbol{u}_E^{\top}\leftarrow\top_E(\boldsymbol{u}_E)$
10: **end for**

Algorithm 2 Geometry computation

1: **for** each particle $L\in\mathcal{L}$ **do**
2: Evolve η_L according to Equation 8
3: **end for**
4: **for** each particle $E\in\mathcal{E}$ **do**
5: Compute a_E according to Equation 9
6: Compute $\eta_E=\frac{V_E}{a_E}$
7: Compute H_E and g_E according to Equation 10
8: **end for**

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

1: Compute enclosed volume $\hat{V}$ with Equation 11
2: Compute enclosed pressure p_{in} with the ideal gas law
3: Solve Γ implicitly with Equation 14
4: **for** each particle $E\in\mathcal{E}$ **do**
5: Compute $\frac{D\boldsymbol{u}_E^{\perp}}{Dt}$ with Equation 12
6: Compute $\frac{D\boldsymbol{u}_E^{\top}}{Dt}$ with Equation 15
7: Update $\boldsymbol{u}^{\perp}$ and $\boldsymbol{u}^{\top}$ using symplectic Euler with Δt
8: **end for**

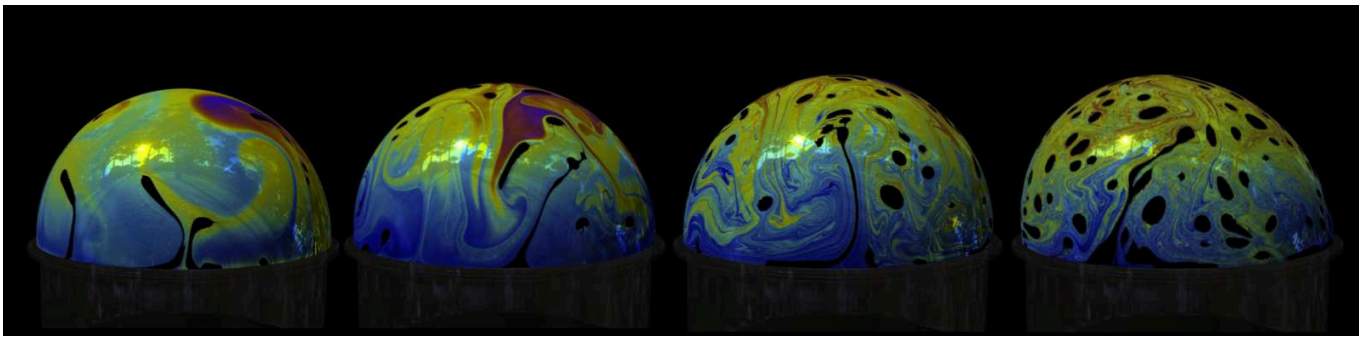

Fig. 6. Different frames of a deforming half bubble with black spots.

4.4 Dynamics Computation

4.4.1 Normal Dynamics. Similar to Ishida et al. [2020], we assume that the normal velocity $\boldsymbol{u}^\perp$ is constant in an $\mathcal{E}$ particle E's control column, *i.e.* $\boldsymbol{u}_E^\perp = \frac{1}{V_E}\int_E \boldsymbol{u}^\perp dV$. Therefore $\frac{D\boldsymbol{u}_E^\perp}{Dt} \approx \frac{1}{V_E}\int_E \frac{D\boldsymbol{u}^\perp}{Dt} dV$. We then need to integrate the right-hand side of the normal component in Equation 5. For the term $\frac{\partial p}{\partial z}$, whose expression is given by Equation 4, integrating over the control column of E yields $a_E \cdot (p_{\text{out}} - p_{\text{in}})$. We assign p_{out} to be the atmospheric pressure p_{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: $p_{\text{in}} = n_0 \bar{R}T/\hat{V}$ with n_0 being the enclosed molar mass and $\hat{V}$ the enclosed volume, which we compute as:

$$\hat{V} = \sum_{E\in\mathcal{E}} \zeta \cdot \frac{1}{3} a_E (O - E), \quad \zeta = \begin{cases} 1, & (O-E)\cdot \mathbf{n}_E \geq 0, \\ -1, & \text{otherwise}, \end{cases} \tag{11}$$

where O is an arbitrarily selected point in $\mathbb{R}^3$ [Zhang et al. 2001]. Similarly, we integrate $\delta_I(\sigma_0 - \bar{R}T\Gamma)H$ and $\boldsymbol{f}_{\text{ext}}^\perp$ over the control column as $2a_E(\sigma_0 - \bar{R}T\Gamma)H_E$ and $V_E \boldsymbol{f}_{\text{ext}}^\perp$ respectively. Hence we obtain the expression for $\frac{D\boldsymbol{u}_E^\perp}{Dt}$ as:

$$\frac{D\boldsymbol{u}_E^\perp}{Dt} = \frac{p_{\text{in}} - p_{\text{out}}}{\rho\eta_E}\mathbf{n}_E + \frac{2(\sigma_0 - \bar{R}T\Gamma)H_E}{\rho\eta_E}\mathbf{n}_E + \frac{\boldsymbol{f}_{\text{ext}}^\perp}{\rho}. \tag{12}$$

4.4.2 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

$$\begin{cases} \dfrac{\boldsymbol{u}^\top - \boldsymbol{u}^{\top *}}{\Delta t} = -\dfrac{2\bar{R}T}{\rho\eta^*}\nabla_s\Gamma + \dfrac{1}{\rho}\boldsymbol{f}_{\text{ext}}^\top, \\ \dfrac{\Gamma - \Gamma^*}{\Delta t} = -\Gamma^*\nabla\cdot\boldsymbol{u}^\top, \\ \dfrac{\eta - \eta^*}{\Delta t} = -\eta^*\nabla\cdot\boldsymbol{u}^\top, \end{cases} \tag{13}$$

where $\boldsymbol{u}^{\top *}$, Γ^* and η^* are the respective quantities after advection, which we collect in the $\mathcal{L}2\mathcal{E}$ step.

Reorganizing Equation 13 yields an implicit equation of Γ:

$$\begin{aligned} &(-\frac{1}{\Delta t\Gamma^*})\Gamma + (\Delta t\frac{\bar{R}T}{\rho}\nabla\frac{1}{\eta^*})\cdot\nabla_s\Gamma + (\Delta t\frac{\bar{R}T}{\rho}\frac{1}{\eta^*})\nabla_s^2\Gamma \\ &= \nabla\cdot\boldsymbol{u}^{\top *} - \frac{1}{\Delta t} + \Delta t(\nabla\frac{1}{\rho}\cdot\boldsymbol{f}_{\text{ext}}^\top + \frac{1}{\rho}\nabla\cdot\boldsymbol{f}_{\text{ext}}^\top). \end{aligned} \tag{14}$$

We solve this equation using the Implicit Incompressible SPH method with a relaxed Jacobi scheme with relaxation parameter $\omega = 0.2$. Once Γ is solved, we evaluate for each $E \in \mathcal{E}$ the tangential acceleration in Equation 5 as:

$$\frac{D\boldsymbol{u}_E^\top}{Dt} = -\frac{2\bar{R}T}{\rho\eta_E}\nabla_s\Gamma + \frac{1}{\rho}\boldsymbol{f}_{\text{ext}}^\top. \tag{15}$$

Then, $\boldsymbol{u}_E^\perp$ and $\boldsymbol{u}_E^\top$ are updated using a symplectic Euler step with Δt.

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

1: **for** each particle $L \in \mathcal{L}$ **do**
2: Compute $\boldsymbol{u}_L$ according to Equation 16
3: Compute $\mathbf{B}_L$ according to Equation 17
4: Compute $\mathbf{D}_L$ according to Equation 18
5: **end for**

4.5 $\mathcal{E}2\mathcal{L}$ Transfer

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

$$\boldsymbol{u}_L = \sum_{E\in\mathcal{N}^{\mathcal{E}}(L)} \hat{W}(E,L)\cdot\boldsymbol{u}_E, \tag{16}$$

$$\mathbf{B}_L = \sum_{E\in\mathcal{N}^{\mathcal{E}}(L)} \hat{W}(E,L)\cdot\top_L(\boldsymbol{u}_E)\otimes\top_L(\boldsymbol{x}_E - \boldsymbol{x}_L), \tag{17}$$

$$\mathbf{D}_L = \sum_{E\in\mathcal{N}^{\mathcal{E}}(L)} \hat{W}(E,L)\cdot\top_L(\boldsymbol{x}_E - \boldsymbol{x}_L)\otimes\top_L(\boldsymbol{x}_E - \boldsymbol{x}_L). \tag{18}$$

4.6 $\mathcal{E}$ Advance

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

$$\boldsymbol{u}_E^{\mathcal{E}}(t) = \boldsymbol{u}_E^\perp(t) + \top_E(\boldsymbol{u}_E^{\mathcal{E}}(t-1)), \tag{19}$$

which takes the tangential component of the previous $\boldsymbol{u}_E^{\mathcal{E}}$ and add to it the current normal velocity. Using $\boldsymbol{u}_E^{\mathcal{E}}$ we advance the positions of the $\mathcal{E}$ particles using a symplectic Euler step with Δt, updating the tracked interface. We also update the local frames $\mathbf{R}_E$ and metric tensors g_E. Then, we *redistribute* the $\mathcal{E}$ particles to maintain uniform distribution. In particular, similar to particle shifting based on

Fick's law of diffusion, we compute a shifting velocity $\overline{\boldsymbol{u}}_E^{\mathcal{E}}$ to prompt particles to flow from high concentration regions to low concentration ones. Using number density δ to gauge the concentration, the problem translates to solving for a constant particle density on the surface with pseudo-pressure C:

$$-\overline{\Delta t^2}(-\delta^*\nabla_s^2 C) = \bar{\delta} - (\delta^* + \overline{\Delta t}(-\delta^*\nabla\cdot\boldsymbol{u}^{\mathcal{E}})), \tag{20}$$

where δ^* and $\bar{\delta}$ stand for the current and average number density of $\mathcal{E}$ particles; and $\overline{\Delta t}$ stands for the temporal step size for redistribution. Equation 20 is solved using IISPH as with Equation 14. The full redistribution procedure is documented in Algorithm 5, where β is the redistribution strength, and φ the threshold deciding whether the distribution is satisfactory. We set β to be the reciprocal of the largest value of $\nabla_s\delta$, and set φ 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.

4.7 $\mathcal{L}$ Advance

As described in Algorithm 7, $\mathcal{L}$ particles advect with $\boldsymbol{u}_L$ using a symplectic Euler step with Δt. They will be projected onto S_B defined by $\mathcal{E}$ with a Moving Least-Squares (MLS)-based approach.

Algorithm 5 $\mathcal{E}$ redistribution

Initialize $\overline{\boldsymbol{u}}^{\mathcal{E}}$ to $\mathbf{0}$
while $\frac{|\delta_{\max}-\delta_{\min}|}{|\mathcal{E}|} \geq \varphi$ **do**
  Solve C implicitly using Equation 20
  for each particle $E \in \mathcal{E}$ **do**
    $\frac{D\overline{\boldsymbol{u}}_E^{\mathcal{E}}}{Dt} \leftarrow -\beta\nabla_s C$
    $\overline{\boldsymbol{u}}_E^{\mathcal{E}} \leftarrow \overline{\boldsymbol{u}}_E^{\mathcal{E}} + \overline{\Delta t}\frac{D\overline{\boldsymbol{u}}_E^{\mathcal{E}}}{Dt}$
    $\boldsymbol{x}_E \leftarrow \boldsymbol{x}_E + \overline{\Delta t}\overline{\boldsymbol{u}}_E^{\mathcal{E}}$
  end for
  Update local frames and metric tensors
  Update δ according to Equation 9
end while
for each particle $E \in \mathcal{E}$ **do**
  $\boldsymbol{u}_E^{\mathcal{E}} \leftarrow \boldsymbol{u}_E^{\mathcal{E}} + \overline{\boldsymbol{u}}_E^{\mathcal{E}}$
end for

Algorithm 6 $\mathcal{E}$ Advance

for each particle $E \in \mathcal{E}$ **do**
  Update $\boldsymbol{u}_E^{\mathcal{E}}$ with Equation 19
  Update $\boldsymbol{x}_E$ using symplectic Euler with Δt
end for
Update local frames and metric tensors
Redistribute $\mathcal{E}$ with Algorithm 5

Algorithm 7 $\mathcal{L}$ Advance

for each particle $L \in \mathcal{L}$ **do**
  Update $\boldsymbol{x}_L$ with $\boldsymbol{u}_L$ using symplectic Euler with Δt
  $\boldsymbol{x}_L \leftarrow \mathrm{Proj}(L)$
end for

4.8 MELP Implementation Details

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

$$\begin{cases} (\nabla_s q)_E = \sum\limits_{E'\in\mathcal{N}^{\mathcal{E}}(E)} a_{E'}\,(q_{E'} - q_E)\,\nabla_s\overline{W}(E,E'), \\ (\nabla_s\cdot\boldsymbol{w})_E = \sum\limits_{E'\in\mathcal{N}^{\mathcal{E}}(E)} a_{E'}\top_E(\boldsymbol{w}_{E'} - \boldsymbol{w}_E)\cdot\nabla_s\overline{W}(E,E'), \\ (\nabla_s^2 q)_E = \sum\limits_{E'\in\mathcal{N}^{\mathcal{E}}(E)} a_{E'}\,(q_{E'} - q_E)\,\dfrac{2|\nabla_s\overline{W}(E,E')|}{|\boldsymbol{x}_E - \boldsymbol{x}_{E'}|}. \end{cases} \tag{21}$$

where $\nabla_s\overline{W}$ is the surface gradient of the 2D kernel function $\overline{W}$, which can be approximated as $\nabla_s\overline{W} = g\nabla\overline{W}$ [Wang et al. 2020]. In practice, we approximate g with $\mathbf{I}_{2\times2}$ with no apparent degradation in performance. For both W and $\overline{W}$, we use the Quintic spline kernel with radius $r = 4\cdot\Delta x$, where Δx 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 $\boldsymbol{u}^{\mathcal{E}}$.

Local Frame and Projection. At time t, a particle $E \in \mathcal{E}$ computes $\mathbf{R}_E$ as follows: 1) perform PCA on $\mathcal{N}^{\mathcal{E}}(E)$ and set the normalized eigenvector with the smallest eigenvalue as $\mathbf{n}'$, 2) set $\mathbf{n}' = -\mathbf{n}'$ if $0 > \mathbf{n}'\cdot\mathbf{n}(t-1)$, 3) construct an intermediate frame $\mathbf{R}' = (\mathbf{e_1}', \mathbf{e_2}', \mathbf{n}')$ where $\mathbf{e_1}'$ is an arbitrary vector perpendicular to $\mathbf{n}'$ and $\mathbf{e_2}' = \mathbf{n}'\times\mathbf{e_1}'$, 4) in the tangent plane, use 2D SPH to compute $\nabla_E z = (\frac{\partial z}{\partial u}, \frac{\partial z}{\partial v})^T$ and let $\mathbf{e_1} = \mathbf{R}'(1,0,\frac{\partial z}{\partial u})^T$, $\mathbf{e_2} = \mathbf{R}'(0,1,\frac{\partial z}{\partial v})^T$, $\mathbf{n} = \mathbf{e_1}\times\mathbf{e_2}$. Finally, $\mathbf{R}_E = (\mathbf{e_1}, \mathbf{e_2}, \mathbf{n})$. A particle $L \in \mathcal{L}$ computes $\mathbf{R}_L$ as follows: 1) compute the average of $\{\mathbf{n}_E | E \in \mathcal{N}^{\mathcal{E}}(L)\}$ 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 $\perp_A(\boldsymbol{w}) = (\boldsymbol{w}\cdot\mathbf{n}_A)\mathbf{n}_A$ and $\top_A(\boldsymbol{w}) = \boldsymbol{w} - \perp_A(\boldsymbol{w})$. We compute $\mathrm{Proj}(A)$ as follows: given local frame $\mathbf{R}_A$, on the tangent plane run MLS with data samples $\{\boldsymbol{x}_B = (u_B, v_B, z_B) | B \in \mathcal{N}^{\mathcal{E}}(A)\}$, fitting z as a function of (u,v). Then let $\hat{z}$ denote the function evaluated at $(0,0)^T$, and set $\boldsymbol{x}_A \leftarrow \boldsymbol{x}_A - \hat{z}\mathbf{n}_A$.

Newton Black Films and Rim Surface Tension. Both effects enter the dynamic system through $\boldsymbol{f}_{\mathrm{ext}}$. The Newton Black Films (*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]. The rim surface tension along the rim's normal direction $\mathbf{n}_{\mathrm{rim}}$ is given by $f_{\sigma,\mathrm{rim}} = 2\sigma + (2\sigma(\pi-1)R_{\mathrm{rim}})/r_c$ [Bush and Hasha 2004] where R_{rim} reflects the thickness of the rim and r_c reflects the size of the thin film. We assume $r_c \gg R_{\mathrm{rim}}$, hence f_σ is dominated by the first term, so $f_{\sigma,\mathrm{rim}} \approx 2\sigma$. We estimate the rim's normal direction following Akinci et al. [2013] as $(\mathbf{n}_{\mathrm{rim}})_E = r\sum_{E'\in\mathcal{N}^{\mathcal{E}}(E)} a_{E'}\nabla_s\overline{W}(E,E')$.

5 MULTI-MELP

We define a foam $F = \{L_i\}_{i=0}^{n}$ as a set of n lamella regions. As shown in Figure 7, 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:

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

(2) ***Contact Handling:*** In each region, each $\mathcal{E}$ 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) ***Surface Tension Sharing:*** In each region, each $\mathcal{E}$ 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 $\mathcal{L}$ particle probabilistically decides another region to which it is transported, based on the surfactant concentration Γ and velocity $\boldsymbol{u}$.

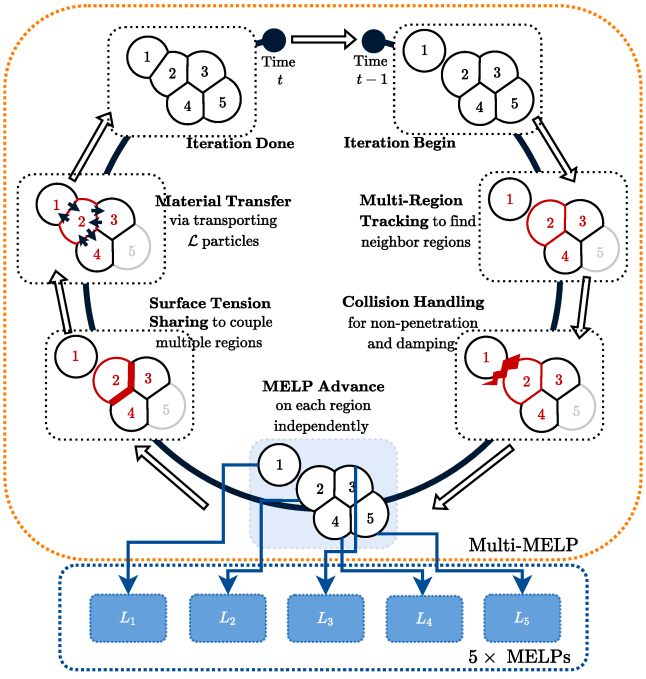

Fig. 7. The schematics of a multi-MELP simulation step; L_1 through L_5 are 5 MELP objects corresponding to the 5 regions.

5.1 Motivating Example

Figure 8 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

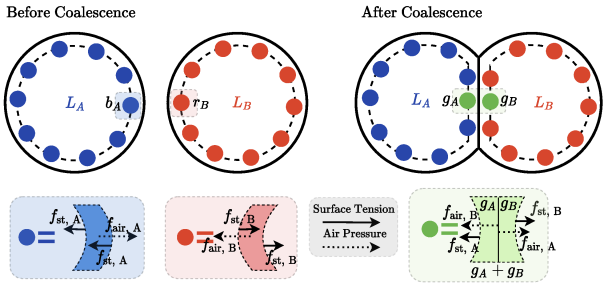

Fig. 8. The formation of a double-bubble. The top row shows the particle perspective; the bottom row depicts the control volumes with force analysis.

two manifold inner surfaces. Dynamically, the inner surfaces of the partition are now constrained by the matching-velocity boundary condition described in Section 3.2. In Figure 8, 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 8. 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 12. 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.*

5.2 Multi-Region Tracking

For an $\mathcal{E}$ 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 $\mathcal{E}$ 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 $\tilde{a} = \sum_{E\in\mathcal{N}} a_E$. This expression gauges the amount of area of L_S that the neighborhood of P encircles. we then compute the same sum of area $\hat{a} = \sum_{E\in\mathcal{N}^{L_K}(P)} a_E$ in L_K. If $\tilde{a} \ll \hat{a}$, then P is relatively far away from L_S; if $\tilde{a} \approx \hat{a}$, then a is in between L_K and L_S. We then compute a *coupling score* $\gamma_{P,S} = \min(1, \frac{\tilde{a}}{\hat{a}})$, and store the tuple $(S, \gamma_{P,S})$ for P.

5.3 Collision Handling

Handling the collision of multiple bubbles entails the treatment of 1) non-penetration, and 2) damping. Consider an $\mathcal{E}$ 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 $\text{Proj}_{L_S}(P) - P$ is outward-pointing to L_S. In that case, a non-penetration force is computed as $\theta_1 \cdot (\text{Proj}_{L_S}(P) - P)$ where θ_1 is the penalty strength. For damping, we first compute an average $\mathcal{E}$ velocity of all nearby regions weighted by γ, as

$$\boldsymbol{u}^{\mathcal{E}}_{\text{avg}} = \frac{\boldsymbol{u}^{\mathcal{E}}_{P} + \sum_{(S,\gamma_{P,S})} \gamma_{P,S} \boldsymbol{u}^{\mathcal{E}}_{\text{Proj}_{L_S}(P)}}{1 + \sum_{(S,\gamma_{P,S})} \gamma_{P,S}}. \tag{22}$$

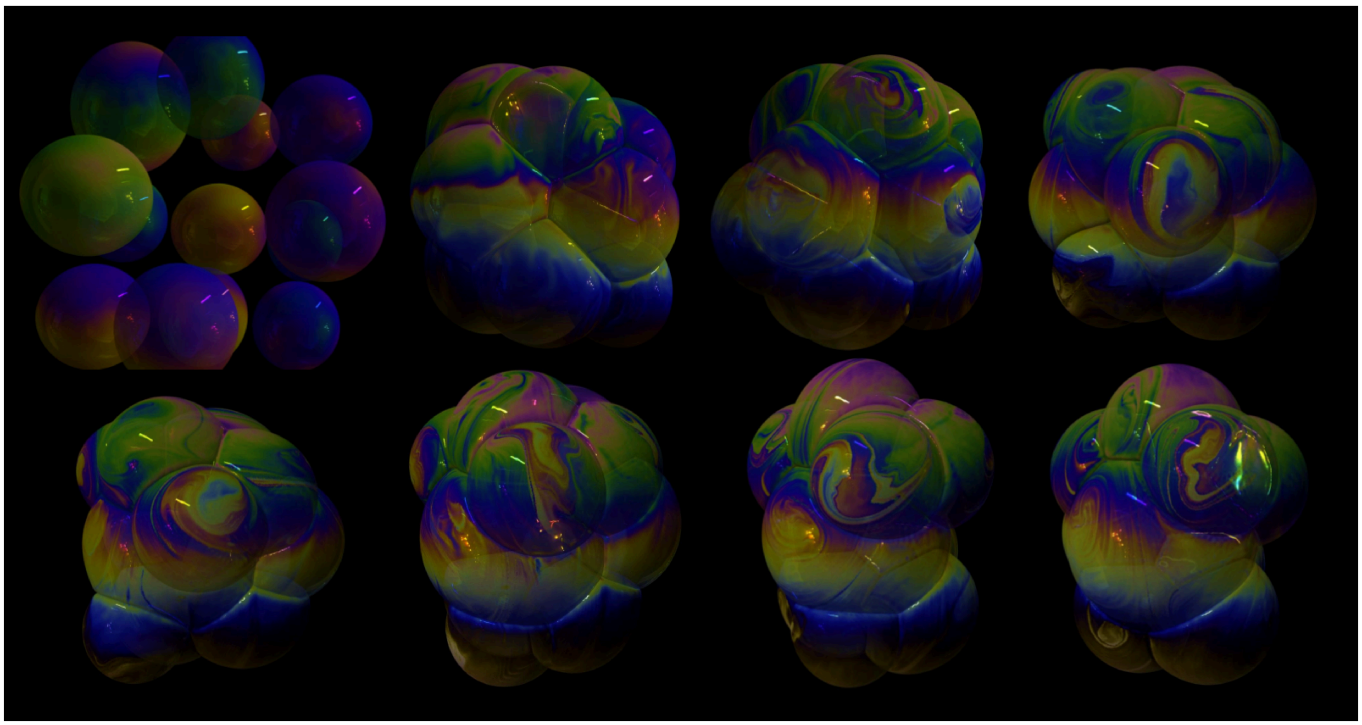

Fig. 9. 13 bubbles merging together, with a heat source at the bottom creating "cyclones" on the surfaces of the bubbles.

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

5.4 Surface Tension Sharing

Given an $\mathcal{E}$ 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:

$$f_{\text{st},L_S} = \frac{\sigma_{\text{Proj}_{L_S}(P)} \cdot H_{\text{Proj}_{L_S}(P)}}{\rho(\eta_{\text{Proj}_{L_S}(P)} + \eta_P)} \cdot \mathbf{n}_{\text{Proj}_{L_S}(P)}, \tag{23}$$

$$f_{\text{air},L_S} = \frac{p_{\text{in},L_S}}{\rho(\eta_{\text{Proj}_{L_S}(P)} + \eta_P)} \cdot \mathbf{n}_{\text{Proj}_{L_S}(P)}. \tag{24}$$

We compute the projection $\text{Proj}_{L_S}(P)$ and interpolate σ, η, H, $\mathbf{n}$ on L_S at $\text{Proj}_{L_S}(P)$. The term p_{in,L_S} 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 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 10. 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 $f_{\text{st, atm}}$ in orange, is not computed. To compute $f_{\text{st, atm}}$, we first compute the pseudo-normal

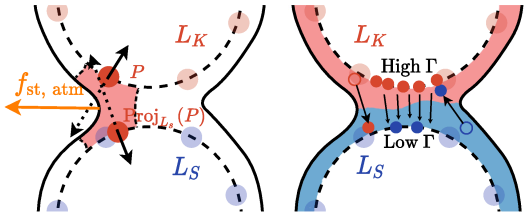

Fig. 10. Left: illustration of the force $f_{\text{st, atm}}$ that must be accounted for. Right: material transport is done by transporting $\mathcal{L}$ particles directly.

vector $\mathbf{n}_{\text{atm}}$ with

$$\mathbf{n}_{\text{atm}} = \frac{\mathbf{n}_P + \sum_{(S,\gamma_{P,S})} \gamma_{P,S} \mathbf{n}_{\text{Proj}_{L_S}(P)}}{1 + \sum_{(S,\gamma_{P,S})} \gamma_{P,S}}, \tag{25}$$

which is a weighted average of the normal vectors of nearby regions. If $|\mathbf{n}_{\text{atm}}| \ll 1$, then the particle P is deemed an internal point, and $f_{\text{st, atm}} = 0$. Otherwise, on the local frame with normal vector $\frac{\mathbf{n}_{\text{atm}}}{|\mathbf{n}_{\text{atm}}|}$, we compute H_{atm} according to Equation 10. Then, we compute $f_{\text{st, atm}} = \frac{\sigma_P \cdot H_{\text{atm}}}{\rho(\eta_{\text{Proj}_{L_S}(P)} + \eta_P)} \cdot \mathbf{n}_{\text{atm}}$. Finally, we have:

$$f_{\text{net}} = f_{\text{st},L_K} + f_{\text{air},L_K} + \sum_{(S,\gamma_{P,S})} (f_{\text{st},L_S} + f_{\text{air},L_S}) + f_{\text{st, atm}}. \tag{26}$$

5.5 Material Transfer

As observed in Section 3.2, near a multi-junction, the surfacial ∇_s no longer applies and should be replaced by the volumetric ∇ in Equation 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 Γ to the ones with low Γ. As depicted on the

right of Figure 10, we devise a probabilistic scheme to directly migrate $\mathcal{L}$ particles from one region to another region, conserving the transported quantities. For each $\mathcal{L}$ particle $Q \in L_K$, let P denote its nearest $\mathcal{E}$ neighbor. For each tuple $(S, \gamma_{P,S})$ that P stores, we compute two probability scores C_{1,L_S} and C_{2,L_S} for Q as follows:

$$C_{1,L_S} = \psi_1 \cdot (1 - \min(1, \frac{|(\boldsymbol{x}_Q + \Delta t \boldsymbol{u}_Q) - \boldsymbol{x}_{\text{Proj}_{L_S}(Q)}|}{|\boldsymbol{x}_Q - \boldsymbol{x}_{\text{Proj}_{L_S}(Q)}|})), \tag{27}$$

$$C_{2,L_S} = \psi_2 \cdot (1 - \min(1, \frac{\Gamma_{\text{Proj}_{L_S}(P)}}{\Gamma_P})), \tag{28}$$

where ψ_1 and ψ_2 are the transport strength parameters. In computing C_{1,L_S}, $\boldsymbol{x}_Q + \Delta t \boldsymbol{u}_Q$ is the position of Q at the next timestep, $\boldsymbol{x}_{\text{Proj}_{L_S}(Q)}$ is the nearest point to Q on L_S. If $\boldsymbol{u}_Q$ is driving Q towards L_S, then this score would be high and vice versa. In computing C_{2,L_S}, we compute the ratio of Γ 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 $\mathcal{L}$ particle Q from L_K to L_G at probability $\min(1, C_G)$.

6 EXPERIMENTS AND RESULTS

6.1 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 $\arccos(-\frac{1}{2}) = 120°$ each. These edges are commonly referred to as the Plateau borders. These Plateau borders then meet in groups of fours, creating angles of $\arccos(-\frac{1}{3}) \approx 109.47°$ each. With the surface tension sharing mechanism, our method accurately recovers both rules. As shown in Figure 11, 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 2, 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 $R_P = \frac{R_1 R_2}{R_1 - R_2}$ and curvature $\kappa_P = \frac{1}{R_P}$.This is due to the three-way balance of Young-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 $R_1 = 0.05$m, 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 12, the smaller R_2 is, the more curved the partition surface becomes. The numerical results are documented in Table 3 and plotted on the top-left of Figure 13, as they conform well to the analytical values with $\leq 3.5\%$ error.

Surface Area Minimization. The standard double-bubble, shown in Figure 22, is a minimal surface with the steady-state surface area $\hat{a}$ given by: $\hat{a} = 27\pi(\frac{\hat{V}}{9\pi})^{\frac{2}{3}}$ 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 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 13, before the merge occurs at $t \approx 3$s, the total area oscillates around $\bar{a}$, which then stabilizes to $\hat{a}$ with periodic oscillation.

Table 2. Numerical results to validate multi-MELP's adherence to Plateau's laws. The pairs are labeled corresponding to Figure 11.

Plateau Border Testing						
Set-up	Double-Bubble			Triple-Bubble		
Pairs	1—2	1—3	2—3	1—2	1—3	2—3
Angle	118.62	122.30	119.06	120.84	118.96	120.20
Error	1.167%	1.917%	0.833%	0.7%	0.867%	0.167%
Set-up	Quadruple-Bubble					
Pairs	1—2	1—3	1—4	2—3	2—4	3—4
Angle	106.26	107.82	114.74	113.52	110.74	103.99
Error	2.93%	1.51%	4.81%	3.70%	1.16%	5.00%

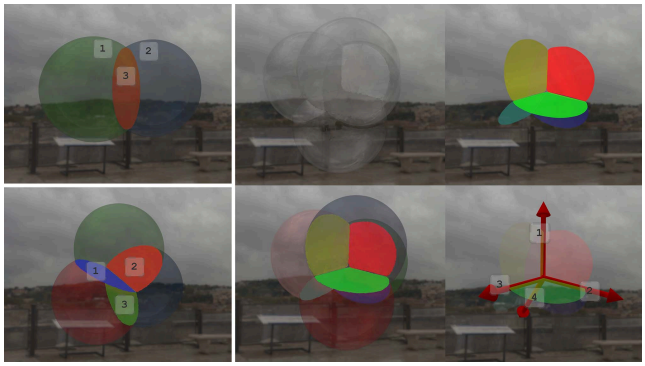

Fig. 11. Left: in a double and a triple-bubble, three pieces of lamellae meet at $\approx 120°$ angles along the border. Right: in a quadruple bubble, 6 partition surfaces (highlighted) form 4 borders (red arrows) meeting at $\approx 109°$ angles.

Drainage under Gravity. When a piece of thin film is placed vertically, gravitation 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 η_0 is the film thickness when laid flat. Setting $\eta_0 = 400nm$, we verify our method's correspondence to the analytical solution on the top-right of Figure 13. Additionally, the exponential thickness variation creates Newton's interference fringes with gradually thinning color stripes towards the bottom, which is depicted on the left of Figure 16.

6.2 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

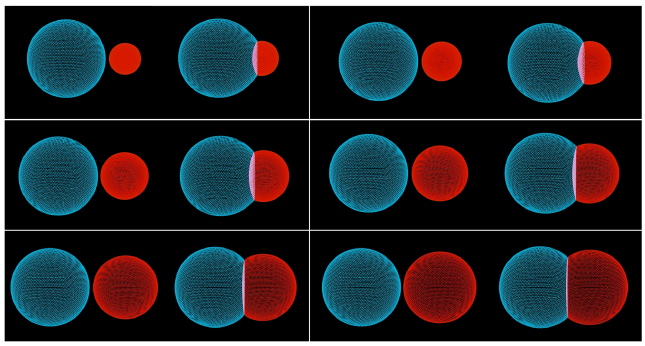

Fig. 12. Double-bubbles with different size ratios. The smaller the red bubble is, the more it protrudes into the larger, blue one.

Table 3. The partition surface curvature: analytical values vs. our experimental values. The † symbol represents the ground truth.

Curvature κ_P of the Partition Surface					
R_1 (m)	R_2 (m)	R_P † (m)	κ_P † ($\frac{1}{\text{m}}$)	κ_P ($\frac{1}{\text{m}}$)	Error
0.05	0.02	0.033	30	28.98	3.41%
0.05	0.025	0.05	20	20.17	0.84%
0.05	0.03	0.075	13.33	13.12	1.57%
0.05	0.035	0.117	8.57	8.34	2.64%
0.05	0.04	0.2	5	4.85	2.99%
0.05	0.045	0.45	2.22	2.28	2.64%

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 16: a circular thin film is initialized with spatially-varying thickness η (top-left circle), which tends to be evened out via the Marangoni effect. The simulation is carried out for 5s, and the variance of η is plotted on the left of Figure 14. Using real-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 15, 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.

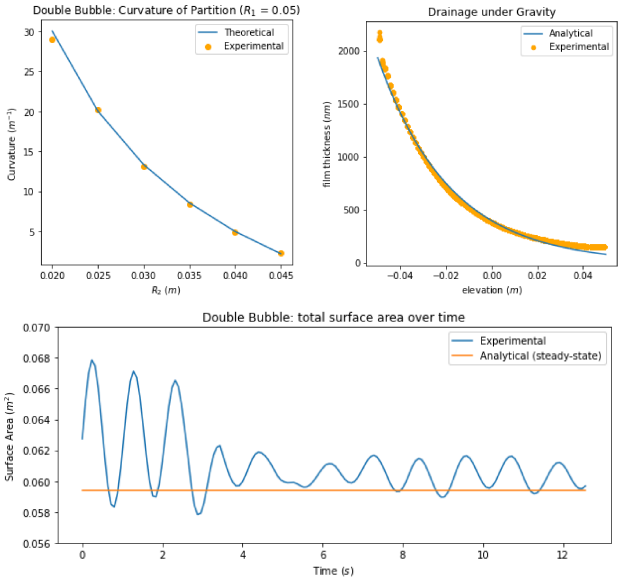

Fig. 13. Top-left: curvatures of the partition surface for double-bubbles of different size ratios compared to analytical values. Top-right: thickness profile under gravity compared to analytical values. Bottom: the evolving surface area of two bubbles as they merge into a double-bubble.

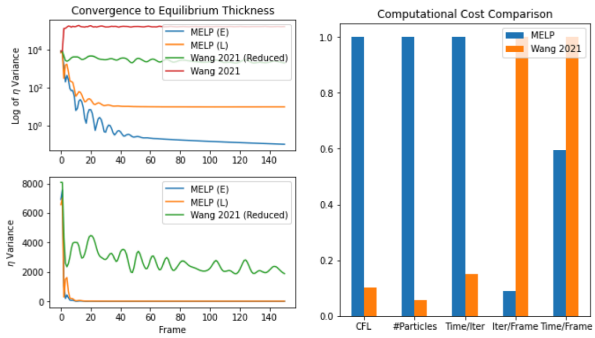

Fig. 14. Comparison with Wang et al. [2021]. Left: test of convergence to equilibrium thickness. Right: comparison of computational cost.

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 4, for Figure 15, MELP advects $\sim$ 700000 $\mathcal{L}$ particles driven by $\sim$ 7000 $\mathcal{E}$ particles, providing a significant resolution boost over the $\sim$ 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 14. 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].

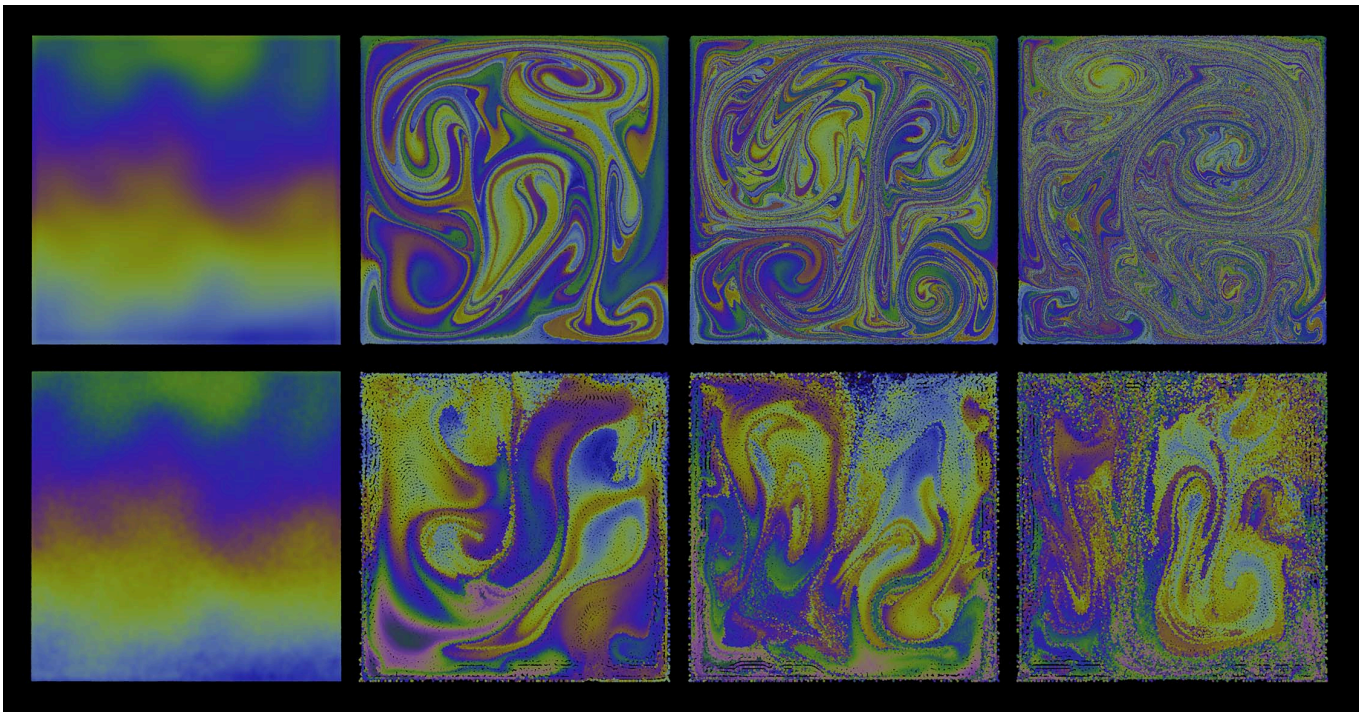

Fig. 15. Comparison of the simulated flow quality of our proposed MELP method (top) and Wang et al. [2021] (bottom).

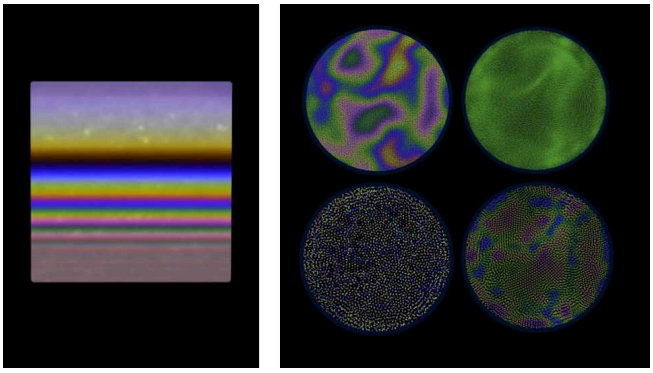

Fig. 16. Left: Newton's interference fringes under gravity. Right: comparison with Wang et al. [2021]: top-left: initial set-up; top-right: converged result of MELP; bottom-left: diverged result of Wang et al. [2021], bottom-right: converged result of Wang et al. [2021] with reduced parameters.

6.3 Examples

The detailed specifications of all the examples simulated by our proposed system, including the computational resources used, are provided in Table 5. 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 19, 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 ($\eta \approx 350$nm) and a green tint at the top ($\eta \approx 500$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× slow motion, which is handled automatically by our program.

Deforming Rectangle with Black Spots. As depicted in Figure 17, we initialize a rectangular thin film with length = 0.16m and height = 0.09m. A constant thickness gradient is initially imposed, with thickness $\eta \approx 500$nm at the top and $\eta \approx 250$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 6, 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 21, 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 20, 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 $\approx 120°$ 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 1, 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 18, three hundred bubbles, whose radii are randomly selected between 0.008m to 0.012m, are 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.

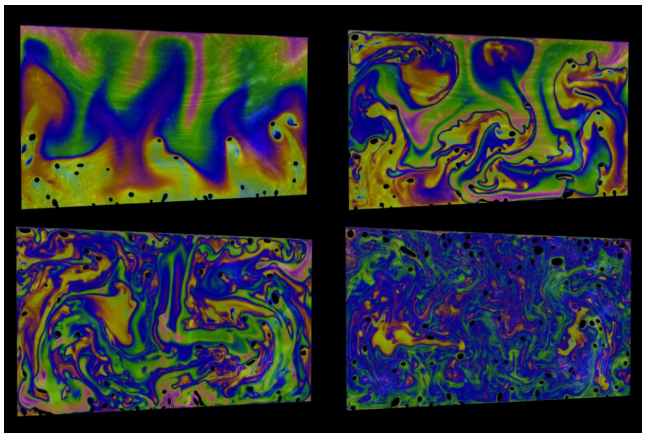

Fig. 17. Different frames of a deforming rectangular film with black spots.

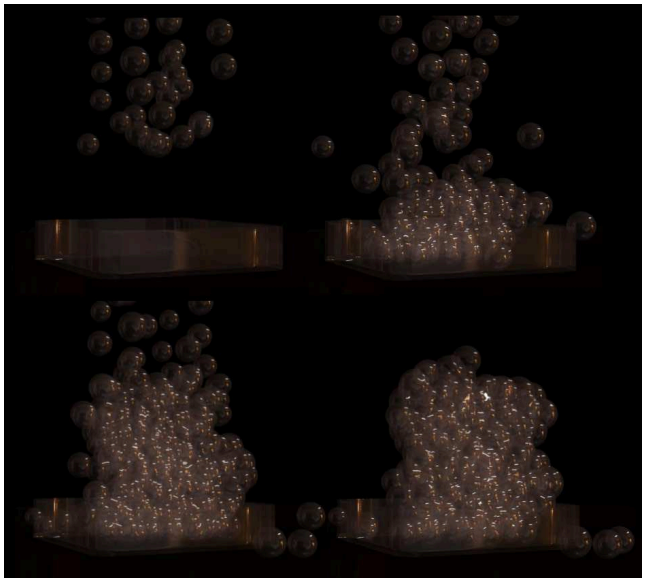

Fig. 18. 300 bubbles falling into a container, forming a foam mountain.

Cyclones on 13 Bubbles. As shown in Figure 9, 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 ($\approx$ 180 nm) and the thickest region at the top being purple ($\approx$ 550nm). This heat-driven convection gradually develops into "cyclones" on the bubble surfaces. The bottom row of Figure 9 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.

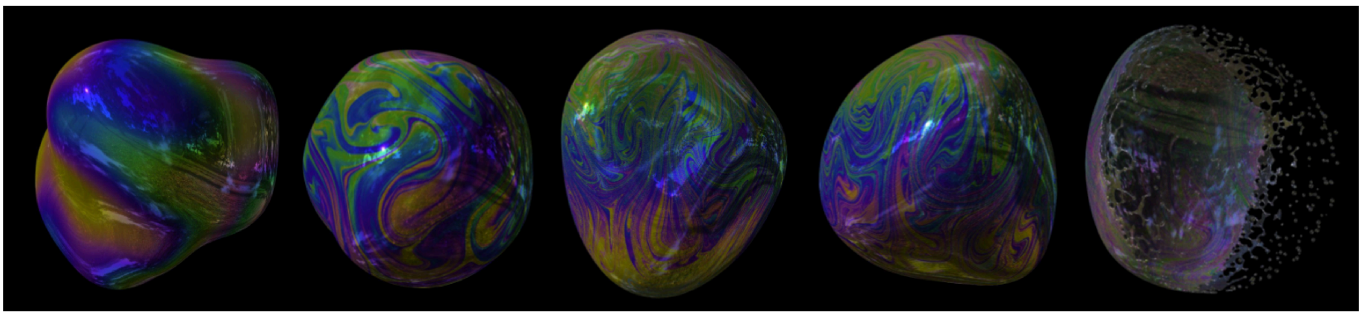

Fig. 19. The flow, deformation, and bursting of a giant bubble, similar to the experiment done in Wang et al. [2021] Figure 5.

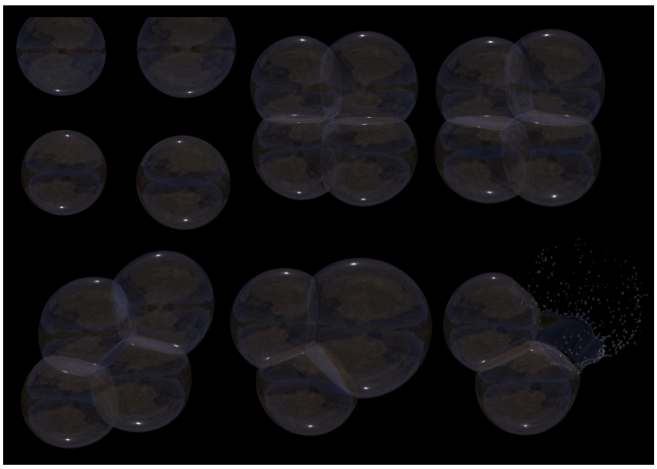

Fig. 20. Four bubbles merge, reorganize from an unstable equilibrium to a stable one, and eventually disintegrate.

7 DISCUSSION AND LIMITATIONS

We present 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 settings, 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, 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.

ACKNOWLEDGMENTS

We would like to thank all the anonymous reviewers for their constructive feedback. We thank Wanxin Hu and Annie Tang for their help with the rendering. We acknowledge NSF-1919647, 2106733, and 2144806 for the funding support. We credit the Houdini Education license for producing the rendered images and animations.

REFERENCES

Keith C Afas. 2018. Extending the Calculus of Moving Surfaces to Higher Orders. *arXiv preprint arXiv:1806.02335* (2018).

Nadir Akinci, Gizem Akinci, and Matthias Teschner. 2013. Versatile surface tension and adhesion for SPH fluids. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–8.

Laurent Belcour and Pascal Barla. 2017. A practical extension to microfacet theory for the modeling of varying iridescence. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14.

John WM Bush and Alexander E Hasha. 2004. On the collision of laminar jets: fluid chains and fishbones. *Journal of fluid mechanics* 511 (2004), 285–310.

Jingyu Chen, Victoria Kala, Alan Marquez-Razon, Elias Gueidon, David A. B. Hyde, and Joseph Teran. 2021. A Momentum-Conserving Implicit Material Point Method for Surface Tension with Contact Angles and Spatial Gradients. *ACM TOG* 40, 4 (2021), 1–16. https://doi.org/10.1145/3450626.3459874

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

Jonathan M Cohen, Sarah Tariq, and Simon Green. 2010. Interactive fluid-particle simulation using translating Eulerian grids. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*. 15–22.

Sylvie Cohen-Addad, Reinhard Höhler, and Olivier Pitois. 2013. Flow in foams and flowing foams. *Annual Review of Fluid Mechanics* 45 (2013), 241–267.

Y Couder, JM Chomaz, and M Rabaud. 1989. On the hydrodynamics of soap films. *Physica D: Nonlinear Phenomena* 37, 1-3 (1989), 384–405.

Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double bubbles sans toil and trouble: Discrete circulation-preserving vortex sheets for soap films and foams. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9.

Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-only liquids. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–12.

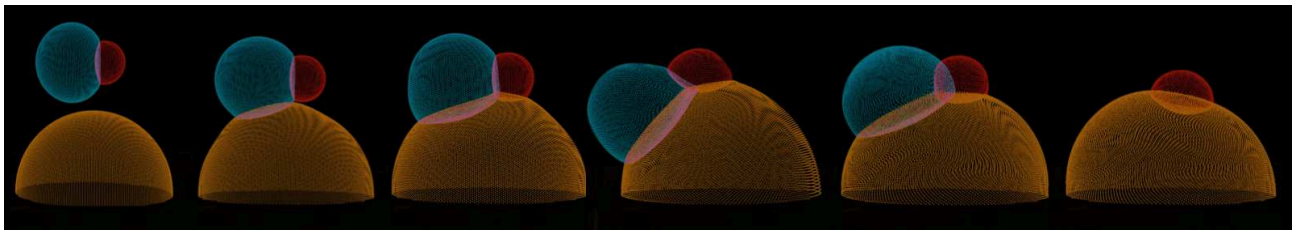

Fig. 21. Interaction among bubbles of different sizes, showcasing our system's ability to restore the equilibrium states.

Table 4. Performance comparison between the single-layer method in Wang et al. [2021] and MELP.

MELP vs. [Wang et al. 2021]: Computational Cost						
Name	Real Parameter	CFL Number	Number of Particles	Time/Iter (s)	Iter/Frame	Time/Frame (s)
[Wang et al. 2021] Equilibrium	✓	0.033	4200	0.07	55.9	3.91
[Wang et al. 2021] Equilibrium	✗	0.1	4200	0.07	3.8	0.27
MELP Equilibrium	✓	0.99	16800 $\mathcal{L}$ + 1050 $\mathcal{E}$	0.33	1.06	0.35
[Wang et al. 2021] Flow	✗	0.1	40000	0.47	24.91	11.71
MELP Flow	✓	0.99	693900 $\mathcal{L}$ + 6900 $\mathcal{E}$	3.1	2.25	6.98

Table 5. 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.

MELP: Catalog of Examples							
Name	CFL number	Number of $\mathcal{E}$	Number of $\mathcal{L}$	Ratio	Time/Iter (s)	Using	Depicted In
Giant Bubble	0.99	163842	2621442	1:16	30.2	A	Figure 19
Deforming Rectangle with Black Spots	0.99	159367	2557467	1:16	24.6	B	Figure 17
Deforming Half Bubble	0.99	81921	5242884	1:64	63.3	B	Figure 6
Two Bubbles Contact	0.99	20480	0	-	0.72	A	Figure 22
Bubbles of Different Sizes	0.33	33280	0	-	1.32	A	Figure 21
Dynamic Reorganization of Four Bubbles	0.99	40960	0	-	1.45	A	Figure 20
R-T Instability on a Double-Bubble	0.99	81924	4772266	1:58	55.7	A	Figure 1
Foam Mountain	0.33	192311	0	-	9.1	A	Figure 18
Cyclones on 13 Bubbles	0.99	133146	2129946	1:16	13.3	A	Figure 9

R Elliot English, Linhai Qiu, Yue Yu, and Ronald Fedkiw. 2013. An adaptive discretization of incompressible flow using a multitude of moving Cartesian grids. *J. Comput. Phys.* 254 (2013), 107–154.

Robert Finn. 1999. Capillary surface interfaces. *Notices of the AMS* 46, 7 (1999), 770–781.

Frederic Gibou, Ronald Fedkiw, and Stanley Osher. 2018. A review of level-set methods and some recent applications. *J. Comput. Phys.* 353 (2018), 82–109.

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

Michael Grinfeld and Pavel Grinfeld. 2017. *Modeling of Stability of Electrostatic and Magnetostatic Systems.* Technical Report. US Army Research Laboratory Aberdeen Proving Ground United States.

Pavel Grinfeld. 2009. Shape optimization and electron bubbles. *Numerical Functional Analysis and Optimization* 30, 7-8 (2009), 689–710.

P Grinfeld. 2010a. Hamiltonian dynamic equations for fluid films. *Studies in Applied Mathematics* 125, 3 (2010), 223–264.

Pavel Grinfeld. 2010b. Variable thickness model for fluid films under large displacement. *Physical review letters* 105, 13 (2010), 137802.

Pavel Grinfeld. 2010c. Viscous equations of fluid film dynamics. *Computers Materials and Continua* 19, 3 (2010), 239.

Pavel Grinfeld. 2013. *Introduction to tensor analysis and the calculus of moving surfaces.* Springer.

Pavel Grinfeld et al. 2009. Exact nonlinear equations for fluid films and proper adaptations of conservation theorems from classical hydrodynamics. *Journal of Geometry and Symmetry in Physics* 16 (2009), 1–21.

Pavel Grinfeld et al. 2012. A better calculus of moving surfaces. *Journal of Geometry and Symmetry in Physics* 26 (2012), 61–69.

David J Hill and Ronald D Henderson. 2016. Efficient fluid simulation on the surface of a sphere. *ACM Transactions on Graphics (TOG)* 35, 2 (2016), 1–9.

Cyrill W Hirt, Anthony A Amsden, and JL Cook. 1974. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of computational physics* 14, 3 (1974), 227–253.

Weizhen Huang, Julian Iseringhausen, Tom Kneiphof, Ziyin Qu, Chenfanfu Jiang, and Matthias B. Hullin. 2020. Chemomechanical Simulation of Soap Film Flow on Spherical Bubbles. *ACM Transactions on Graphics* 39, 4 (2020). https://doi.org/{10.1145/3386569.3392094}

David A. B. Hyde, Steven W. Gagniere, Alan Marquez-Razon, and Joseph Teran. 2020. An Implicit Updated Lagrangian Formulation for Liquids with Large Surface Energy. *ACM TOG* 39, 4 (2020). https://doi.org/10.1145/3414685.3417845

Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. 2013. Implicit incompressible SPH. *IEEE transactions on visualization and computer graphics* 20, 3 (2013), 426–435.

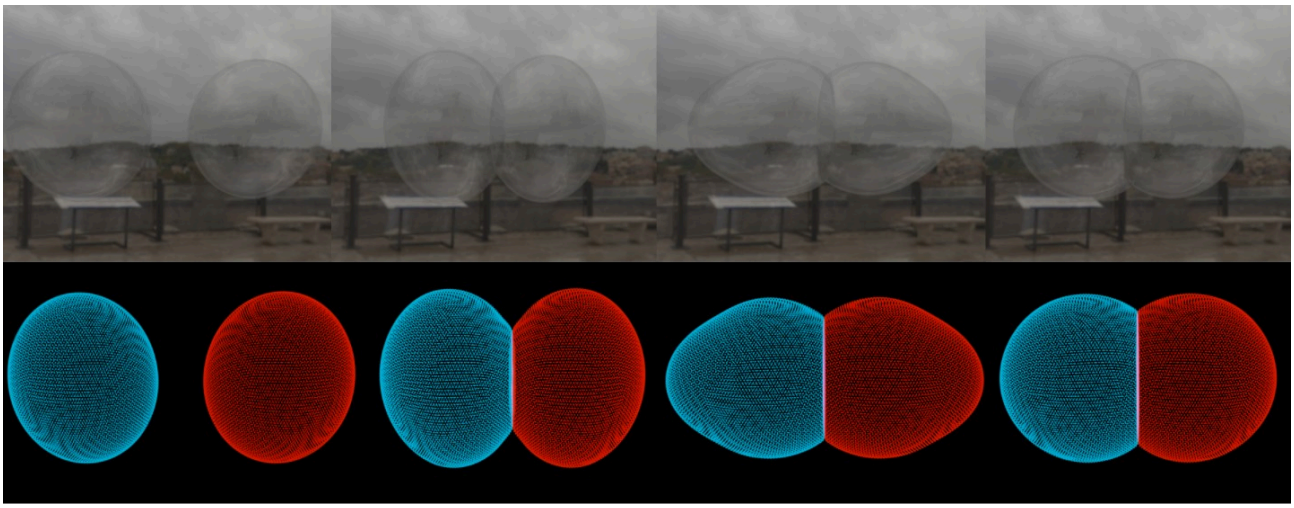

Fig. 22. Formation and evolution of a double-bubble. Top: photorealistic rendering, bottom: particle view.

Sadashige Ishida, Peter Synak, Fumiya Narita, Toshiya Hachisuka, and Chris Wojtan. 2020. A Model for Soap Film Dynamics with Evolving Thickness. *ACM Transactions on Graphics* 39, 4, Article 31 (2020), 31:1–31:11 pages. https://doi.org/{10.1145/3386569.3392405}

Sadashige Ishida, Masafumi Yamamoto, Ryoichi Ando, and Toshiya Hachisuka. 2017. A hyperbolic geometric flow for evolving films and foams. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–11.

Kei Iwasaki, Keichi Matsuzawa, and Tomoyuki Nishita. 2004. Real-time rendering of soap bubbles taking into account light interference. In *Proceedings Computer Graphics International, 2004.* IEEE, 344–348.

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

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.

M. Kness. 2008. ColorPy-A Python package for handling physical descriptions of color and light spectra. (2008).

Stephan A Koehler, Sascha Hilgenfeldt, and HA Stone. 2004. Foam drainage on the microscale: I. Modeling flow through single Plateau borders. *Journal of colloid and interface science* 276, 2 (2004), 420–438.

Petros Koumoutsakos. 2005. Multiscale flow simulations using particles. *Annu. Rev. Fluid Mech.* 37 (2005), 457–487.

Andrew M Kraynik, Douglas A Reinelt, and Frank van Swol. 2004. Structure of random foam. *Physical Review Letters* 93, 20 (2004), 208301.

David IW Levin, Joshua Litven, Garrett L Jones, Shinjiro Sueda, and Dinesh K Pai. 2011. Eulerian solid simulation with contact. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 1–10.

S.J. Lind, R. Xu, P.K. Stansby, and B.D. Rogers. 2012. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *J. Comput. Phys.* 231, 4 (2012), 1499 – 1523. https://doi.org/10.1016/j.jcp.2011.10.027

Dieter Morgenroth, Stefan Reinhardt, Daniel Weiskopf, and Bernhard Eberhardt. 2020. Efficient 2D simulation on moving 3D surfaces. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 27–38.

Daniel Ram, Theodore Gast, Chenfanfu Jiang, Craig Schroeder, Alexey Stomakhin, Joseph Teran, and Pirouz Kavehpour. 2015. A material point method for viscoelastic fluids, foams and sponges. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* 157–163.

Milton J Rosen and Joy T Kunjappu. 2012. *Surfactants and interfacial phenomena.* John Wiley & Sons.

Amaresh Sahu, Yannick AD Omar, Roger A Sauer, and Kranthi K Mandadapu. 2020. Arbitrary Lagrangian–Eulerian finite element method for curved and deforming surfaces: I. General theory and application to fluid interfaces. *J. Comput. Phys.* 407 (2020), 109253.

Robert I Saye and James A Sethian. 2013. Multiscale modeling of membrane rearrangement, drainage, and rupture in evolving foams. *Science* 340, 6133 (2013), 720–724.

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

Hagit Schechter and Robert Bridson. 2012. Ghost SPH for animating water. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–8.

Nicholas Sharp and Keenan Crane. 2020. A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum (SGP)* 39, 5 (2020).

Brian E Smits and Gary W Meyer. 1992. Newton's colors: simulating interference phenomena in realistic image synthesis. In *Photorealism in Computer Graphics.* Springer, 185–194.

Barbara Solenthaler. 2011. *SPH Based Shallow Water Simulation.* The Eurographics Association.

Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.

David V Svintradze. 2019. Shape dynamics of bouncing droplets. *Scientific reports* 9, 1 (2019), 1–10.

Gilberto L Thomas, Julio M Belmonte, François Graner, James A Glazier, and Rita MC de Almeida. 2015. 3D simulations of wet foam coarsening evidence a self similar growth regime. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* 473 (2015), 109–114.

Hui Wang, Yongxu Jin, Anqi Luo, Xubo Yang, and Bo Zhu. 2020. Codimensional surface tension flow using moving-least-squares particles. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 42–1.

Mengdi Wang, Yitong Deng, Xiangxin Kong, Aditya H. Prasad, Shiying Xiong, and Bo Zhu. 2021. Thin-Film Smoothed Particle Hydrodynamics Fluid. *ACM Trans. Graph.* 40, 4, Article 110 (jul 2021), 16 pages. https://doi.org/10.1145/3450626.3459864

Stephanie Wang and Albert Chern. 2021. Computing minimal surfaces with differential forms. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.

D Weaire and R Phelan. 1996. The physics of foam. *Journal of Physics: Condensed Matter* 8, 47 (1996), 9519.

J. Z. Wu, H. Y. Ma, and M. D. Zhou. 2006. *Vorticity and Vortex Dynamics.* Springer.

Jian Jun Xu, Zhilin Li, John Lowengrub, and Hongkai Zhao. 2006. A level-set method for interfacial flows with surfactant. *J. Comput. Phys.* 212, 2 (2006), 590–616.

Bowen Yang, William Corse, Jiecong Lu, Joshuah Wolper, and Chen-Fanfu Jiang. 2019. Real-Time Fluid Simulation on the Surface of a Sphere. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 1 (2019), 1–17.

Jiaping You and Yue Yang. 2020. Modelling of the turbulent burning velocity based on Lagrangian statistics of propagating surfaces. *Journal of Fluid Mechanics* 887 (2020).

Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2015. Continuum foam: A material point method for shear-dependent flows. *ACM Transactions on Graphics (TOG)* 34, 5 (2015), 1–20.

Y. L. Zhang, K. S. Yeo, B. C. Khoo, and C. Wang. 2001. 3D Jet Impact of Toroidal Bubbles. *J. Comput. Phys.* 166 (2001), 336–360.

Wen Zheng, Jun-Hai Yong, and Jean-Claude Paul. 2009. Simulation of bubbles. *Graphical Models* 71, 6 (2009), 229–239.

Bo Zhu, Ed Quigley, Matthew Cong, Justin Solomon, and Ronald Fedkiw. 2014. Codimensional surface tension flow on simplicial complexes. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.

Y. Zhu and R. Bridson. 2005. Animating sand as a fluid. *ACM Trans. Graph. (SIGGRAPH Proc.)* 24, 3 (2005), 965–972.

A DERIVATION OF THE MOMENTUM EQUATION

Since the transportation of physical quantities on a moving surface S_B is independent of the velocity field outside S_B, we may as well assume that the surface S_B is immersed in a continuous 3D velocity field. Assuming S_δ is an arbitrary Lagrangian material surface element on S_B, the physical quantity defined on S_δ satisfies the following transportation equation [Wu et al. 2006]

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{S_\delta} F\mathrm{d}S = \int_{S_\delta}\left(\frac{\mathrm{D}F}{\mathrm{D}t} + \mathbf{n}\cdot\mathbf{B}\cdot\mathbf{n}F\right)\mathrm{d}S, \tag{29}$$

where F is either a scalar or a vector, and $\mathbf{B} = \mathbf{I}(\nabla\cdot\boldsymbol{u}) - (\nabla\boldsymbol{u})^T$ is the divergence-free surface-deformation tensor with $\mathbf{I}$ being the identity matrix.

Setting $F = \rho\eta$ in Equation 29 and using the mass conservation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{S_\delta}\rho\eta\mathrm{d}S = 0, \tag{30}$$

we obtain

$$\int_{S_\delta}\left(\frac{\mathrm{D}\rho\eta}{\mathrm{D}t} + \mathbf{n}\cdot\mathbf{B}\cdot\mathbf{n}\rho\eta\right)\mathrm{d}S = 0. \tag{31}$$

From the classical Newton's second law, the rate of change of fluid momentum of a material surface element S_δ must be balanced by the total body force $\boldsymbol{f}_L$ exerted over S_δ and tangential stress Π exerted on its boundary ∂S_δ. Assigning $F = \rho\eta\boldsymbol{u}$ in Equation 29, the integral momentum balance reads

$$\int_{S_\delta}\left(\frac{\mathrm{D}\rho\eta\boldsymbol{u}}{\mathrm{D}t} + \mathbf{n}\cdot\mathbf{B}\cdot\mathbf{n}\rho\eta\boldsymbol{u}\right)\mathrm{d}S = \int_{S_\delta}\rho\eta\boldsymbol{f}_L\mathrm{d}S + \int_{\partial S_\delta}\rho\eta\Pi\mathbf{n}_\partial\mathrm{d}l, \tag{32}$$

where $\mathbf{n}_\partial$ is the unit normal of ∂S_δ.

Subtracting Equation 31 multiplied by $\boldsymbol{u}$ from Equation 32 and using Stokes' theorem, we obtain

$$\int_{S_\delta}\left(\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} - \boldsymbol{f}_L + \boldsymbol{\nabla}_s\Pi\right)\mathrm{d}S = 0. \tag{33}$$

Since S_δ is arbitrarily selected, Equation 33 can be converted into a differential form Equation 3.

B IISPH WITH JACOBI ITERATIONS

To solve Equation 14 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 Γ_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:

$$(a_{ii})_1 = -\frac{1}{\Delta t\Gamma_i^*}. \tag{34}$$

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

$$(\Delta t\frac{\bar{R}T}{\rho}(\nabla\frac{1}{\eta^*})_i)\cdot\nabla\Gamma_i \tag{35}$$

$$= (\Delta t\frac{\bar{R}T}{\rho}(\nabla\frac{1}{\eta^*})_i)\cdot(\sum_{j\in\mathcal{N}(i)} a_j(\Gamma_j-\Gamma_i)\nabla W_{ij}). \tag{36}$$

The diagonal coefficients would be:

$$(a_{ii})_2 = \sum_{j\in\mathcal{N}(i)} -a_j\nabla W_{ij}\cdot(\Delta t\frac{\bar{R}T}{\rho}(\nabla\frac{1}{\eta^*})_i). \tag{37}$$

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

$$\nabla^2\Gamma = \sum_{j\in\mathcal{N}(i)} a_j(\nabla\Gamma_j - \nabla\Gamma_i)\cdot\nabla W_{ij} \tag{38}$$

$$= \sum_{j\in\mathcal{N}(i)} a_j(\sum_{k\in\mathcal{N}(j)} a_k(\Gamma_k-\Gamma_j)\nabla W_{jk} \tag{39}$$

$$- \sum_{j\in\mathcal{N}(i)} a_j(\Gamma_j-\Gamma_i)\nabla W_{ij})\cdot\nabla W_{ij}. \tag{40}$$

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:

$$(\nabla^2\Gamma)_{ii} = \sum_{j\in\mathcal{N}(i)} a_j(a_i\nabla W_{ji} \tag{41}$$

$$- \sum_{j\in\mathcal{N}(i)} -a_j\nabla W_{ij})\cdot\nabla W_{ij} \tag{42}$$

$$= \sum_{j\in\mathcal{N}(i)} a_j(-a_i\nabla W_{ij} \tag{43}$$

$$- \sum_{j\in\mathcal{N}(i)} -a_j\nabla W_{ij})\cdot\nabla W_{ij} \tag{44}$$

$$= -\sum_{j\in\mathcal{N}(i)} a_j(-a_i\nabla W_{ij} \tag{45}$$

$$- \sum_{j\in\mathcal{N}(i)} -a_j\nabla W_{ij})\cdot\nabla W_{ij} \tag{46}$$

$$(a_{ii})_3 = (\Delta t\frac{\bar{R}T}{\rho}\frac{1}{\eta^*})\cdot(\nabla^2\Gamma)_{ii}. \tag{47}$$

Finally,

$$a_{ii} = (a_{ii})_1 + (a_{ii})_2 + (a_{ii})_3. \tag{48}$$

Once the diagonal terms have been derived, the rest of the iterative process is analogous 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.