A Moving Least-Squares/Level-Set Particle Method for Bubble and Foam Simulation

Hui Wang, Zhi Wang, Shulin Hong, Xubo Yang, Bo Zhu†

Abstract—We present a novel particle-grid scheme for simulating bubble and foam flow. At the core of our approach lies a particle representation that combines the computational nature of moving least-squares particles and particle level-set methods. Specifically, we assign a dedicated particle system to each individual bubble, enabling accurate tracking of its interface evolution and topological changes in a foaming fluid system. The particles within each bubble's particle system serve dual purposes. Firstly, they function as a surface discretization, allowing for the solution of surfactant flow physics on the bubble's membrane. Additionally, these particles act as interface trackers, facilitating the evolution of the bubble's shape and topology within the multiphase fluid domain. The combination of particle systems from all bubbles contributes to the generation of an unsigned level-set field, further enhancing the simulation of coupled multiphase flow dynamics. By seamlessly integrating our particle representation into a multiphase, volumetric flow solver, our method enables the simulation of a broad range of intricate bubble and foam phenomena. These phenomena exhibit highly dynamic and complex structural evolution, as well as interfacial flow details.

Index Terms—level set, interface tracking, particle methods, moving least-squares, multiphase fluid, surface tension.

1 Introduction

Bubbles and foams are ubiquitous in nature, popping with enchanting colors from soapy water containing surfactants, forming complex patterns during hygiene practices with hand sanitizer, and arising through captivating interfacial wave dynamics. The formation, movement, and eventual bursting of an individual bubble serve as the foundation for understanding essential mechanics in numerous fluid processes, showcasing captivating and intricate color patterns, surface deformations, and topological transformations. As multiple bubbles adhere and cluster together, fascinating foam structures spontaneously emerge along the fluid interface. These complex structures flow within the fluid body, influenced by the interplay of surface tension, surfactant dynamics, and fluid flow.

Despite the inherent connections between bubbles and foams in terms of their fluid dynamics, they are commonly treated differently in the world of numerical simulations. Soap bubble simulations, historically, have been considered as modeling thin films, and most simulation frameworks were developed using explicit, Lagrangian meshes [1], [2] or surface particles [3], [4]. In contrast, foam simulations are treated as volumetric phenomena, where the Eulerian grid plays a crucial role in discretizing the fluid domain and tracking multiple fluid volume fractions [5] or interface representations [6], [7], typically in an implicit manner. When grids and particles are combined in simulating bubble/foam flow (e.g., as seen in [8]–[10]), the Lagrangian components commonly serve a complementary role, specifically utilized to track numerous small drops or bubbles that cannot be adequately characterized by the grid cells. Such inhomogeneous numerical treatments of bubbles and foams, however, stemming from the homogeneous physical principles of surface tension and surfactant dynamics, result in apparent limitations on both fronts. On one hand, bubble simulations primarily aim to capture the vibrant intricacies of interfacial flow, often disregarding the surrounding fluid environment (such as water or airflow). On the other hand, foam simulations typically focus on addressing fluid volume transport and topological evolution, paying little attention to the finer details of interfacial or membrane flow.

In this paper, we have developed a novel particle-grid scheme for simulating bubble and foam flow. Our approach differs from previous literature, where particles were primarily used for mass transport or subcellular bubble volume capturing. Instead, we utilize multiple particle systems to represent the nonmanifold membranes of foam and track their topological changes. Our key concept involves assigning a dedicated particle system to each individual bubble, allowing us to track its interface evolution. The particles within a bubble's particle system play dual roles in the simulation. First, they act as *interface trackers*, facilitating the evolution of the bubble's shape and topology within the multiphase fluid domain. The combination of particle systems belonging to all bubbles facilitates the generation of an unsigned level-set field based on local least-squares fitting, which further aids in solving the coupled multiphase flow dynamics between surface tension and fluid incompressibility discretized on a background grid. Second, these particles serve as a *surface discretization*, enabling the solution of flow physics on the bubble's membrane. Specifically, these discrete particles are employed to fit a moving least-squares surface, discretizing the film surface and facilitating the analysis of surfactant flow details. Due to the combined mechanisms of moving least-squares fitting and level-set approximation associated with each particle, we refer to our

• *Hui Wang, Zhi Wang, Shulin Hong and Xubo Yang are with School of Software, Shanghai Jiao Tong University. E-mail: wanghehv@sjtu.edu.cn, wangzatu@gmail.com, {zqpshigui, yangxubo}@sjtu.edu.cn*
• *Bo Zhu is with School of Interactive Computing, Georgia Institute of Technology. E-mail: bo.zhu@gatech.edu*
† *Corresponding author*

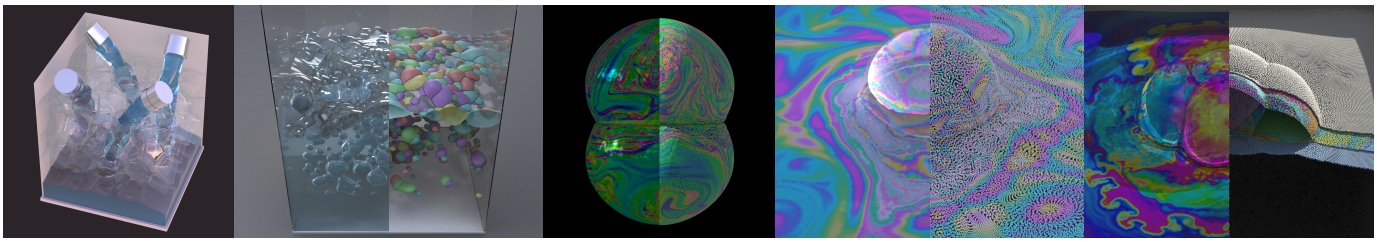


Fig. 1. Simulation of various foam and bubble phenomena using our approach. *Far left:* Four jets emitting above foam clusters, flowing through the gaps between them; *Middle left:* Numerous submerged bubbles rising in an aquarium. We plot the colored meshes of the bubbles on the right. *Middle:* Two colliding bubbles undergo oscillations, fostering intricate interfacial flow. We render the Lagrangian particles colored according to their thickness on the right. *Middle right:* A submerged bubble ascending and resting on the water surface, coated with a surfactant layer that exhibits visually captivating patterns. We render the colored Lagrangian particles on the right. *Far right:* Multiple bubbles interact within a liquid tank, their interfaces interconnecting to create a complex system featuring multiregional volumetric and interfacial flow. We visualize both the MLSLS particles and Lagrangian particles on the right.

particles as *Moving Least-Squares/Level-Set (MLSLS) Particles*.

Our approach distinguishes itself from both the traditional Eulerian and Lagrangian approaches for bubble and foam simulation. On one hand, when compared with the traditional volumetric methods such as multiple level-set [7], [11] and VOF [5] schemes, our method possesses strong Lagrangian properties in both geometric and dynamic aspects due to the utilization of interface particles. These particles not only provide discretization stencils for solving tangential flow physics but also enable robust and accurate multiple interface tracking capabilities, which proves particularly advantageous when dealing with scenes involving numerous non-coalescing bubbles of varying sizes. In this regard, our approach can be understood as a novel particle level-set method designed to address multiphase interface tracking problems, incorporating inherent region identification and feature-preserving capabilities.

On the other hand, our approach enhances the expressiveness of traditional Lagrangian approaches (e.g., [1]) in modeling complex and turbulent volumetric flow phenomena. Specifically, our approach seamlessly integrates into a multiphase, volumetric flow solver (e.g., [7]), allowing for the simulation of complex bubble and foam behaviors with intricate topological features while preserving essential interfacial flow details. Therefore, our approach can also be understood as a volumetric extension of the moving particle methods (e.g., [3], [4]) that emerged recently, broadening their applicability from film-only fluid domains to encompass multiphase volumes. Due to the combination of merits on both ends, our approach facilitates the simulation of a wide range of bubble and foam phenomena that were previously challenging for pure Eulerian or Lagrangian methods. These phenomena include colliding bubbles, bubble clusters, jets on bubbles, foaming flow, and the full life cycle of a rising bubble (Figure 1).

We summarize the main contributions as follows:

- A new Lagrangian representation combining computational merits from both moving least squares and particle level sets for multiphase interface tracking
- A coupled system to track the topological evolution and solve multiphase foam physics
- A unified foam simulation framework to simulate bubble and foam phenomena manifesting complex tangential flow details and topological film evolution

2 Related Work

Bubble simulation: The simulation of films and bubbles, which are regarded as the codimensional thin surface mainly driven by the surface tension, has been an active area of research in computer graphics. The Lagrangian structures, including both meshes [1], [2], [12]–[15] and particles [3], [4], [16], [17] are widely employed in bubble simulation, due to their inherent capability of representing and tracking surface geometries. However, with only the surface geometry, the Lagrangian methods often require specially designed techniques [18], [19] to handle collision, penetration, and topological changes, especially when dealing with a large number of interfaces. The dynamics of bubbles consists of the interfacial flow and the membrane deformation, which are usually resolved in a decoupled manner [2], [4]. To address the interfacial flow, researchers usually reduce the fluid equations onto the surface and introduce the varying thickness and surfactant concentration on the surface, which are then resolved based on the surface parametrization [2], [3], [20], [21]. The membrane deformation is commonly resolved in a reduced manner by dropping the velocity inside the bubble and enforcing a global incompressibility constraint. The surface-only technique [22], [23] further extends the surface structures to solve the volumetric fluid equations. In this paper, we enhance the surface meshless particles by incorporating a background grid, enabling us to efficiently handle the topological changes and solve the volumetric flow.

Foam simulation: Foams are typically treated as the interfaces between different subdomains in the multiphase Eulerian simulation framework [7], [24], forming the non-coalescing complex topological structures as they advect with the volumetric velocity. The dynamics of the foam surfaces is often neglected, due to their significantly smaller mass compared with the liquid phase. From a dynamics perspective, the presence of the foam interfaces introduces jump conditions of pressure and velocity gradient [25], [26], which can be easily solved on the background grid in the multiphase Eulerian framework. From the geometric perspective, the interactions between foams and the volumetric liquid form intricate geometries, especially in turbulent foaming liquid flow. To track multiple non-manifold interfaces, researchers proposed various methods to define the implicit surface representation on the grid,

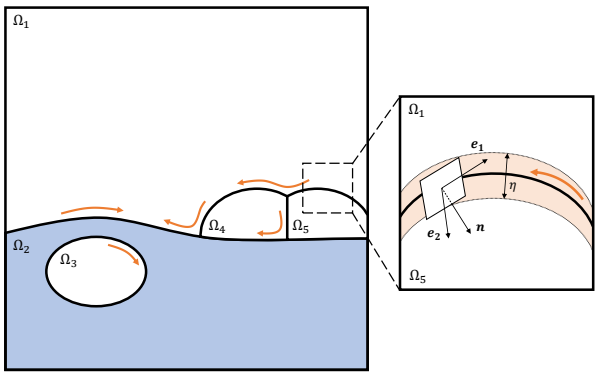


Fig. 2. Illustration of fluid domains. *Left:* The entire domain is divided into multiple volumetric regions Ω_i and the their interfaces $\partial\Omega_i$. *Right:* The interfaces are treated as thin films, characterized by the local frame $\boldsymbol{R} = [\boldsymbol{e}_1, \boldsymbol{n}]$ in 2D ($\boldsymbol{R} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{n}]$ in 3D) and thickness η, where the interfacial flow is solved.

including regional level set [6], [11], [27], particle level sets [7], multiple marker level set [28], Voronoi implicit interfaces method (VIIM) [12], [29], VOF [5]. By utilizing these implicit representations on the grid, we can effectively handle the evolving topology during the simulation and maintain stability during drastic topological changes including splitting, merging, and self-collision. To track the foam even in chaotic flow, where the surface tension plays a less important role, researchers employ techniques such as FLIP [26], [30], [31] and the lattice Boltzmann method [32], [33]. In addition, there are also grid-based approaches that specifically focus on sub-grid detail of foams, which exhibit the continuum property [34], the particulate property [9], [35], [36] and the whitewater effect [10], where Lagrangian structures are incorporated in the Eulerian framework to track the small-scale features in the simulation. By integrating Lagrangian foam particles, Patkar et al. [8] achieve a multi-scale foam simulation.

3 Physical Model

In this section, we present the physical model we employed to simulate the multiphase bubble and foam flow systems.

3.1 Domain Definition

Volumetric Regions: As depicted in Figure 2 (Left), we divide the entire domain Ω into separate volumetric regions. Each region Ω_i is occupied by a uniform single-phase fluid and identified by a unique index number i. We classify the regions based on the specific phase of fluid they contain, resulting in liquid regions Ω_L (e.g., droplet, liquid bulk) and gas regions Ω_G (e.g., foam, bubble, ambient air).

Thin Film: We define the thin film geometry on the interface in the following two situations: (1) the interface between two gas regions, where a thin fluid layer is confined between two air-liquid interfaces; (2) the interface between a gas region and a liquid region, which is covered by a thin liquid layer, e.g. the surfactant spreading over a water tank. In both cases, the film thickness (10^{-7}m) is significantly smaller than its characteristic length (10^{-2}m), allowing us to treat the local geometry as thin film lamellae.

As shown in Figure 2 (Right), we define the thin film geometry on the interfaces. To determine the continuous surface normal, we examine the surface from one specific region and use the outwards direction $\boldsymbol{n}$ as the normal vector. Then we spawn the local frames $\boldsymbol{R} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{n}]$ ($\boldsymbol{R} = [\boldsymbol{e}_1, \boldsymbol{n}]$ in 2D) from the normal, where $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ ($\boldsymbol{e}_1$ in 2D) are the tangential basis. The thickness of the film η is defined along the normal.

3.2 Volumetric Flow Model

We solve the inviscid volumetric multiphase flow in both liquid regions and gas regions based on the incompressible Euler equations

$$\begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} = -\dfrac{\boldsymbol{\nabla} p}{\rho_i} + \boldsymbol{g}, \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0, \end{cases} \quad \boldsymbol{x}\in\Omega_i, \;\; i\in[1, n_\Omega] \tag{1}$$

with the interface jump conditions

$$\begin{cases} [p] = \gamma\kappa, \;\; \boldsymbol{x}\in\partial\Omega_i\cap\partial\Omega_j \text{ and } \Omega_i\in\Omega_L \text{ and } \Omega_j\in\Omega_G, \\ [p] = 2\gamma\kappa, \;\; \boldsymbol{x}\in\partial\Omega_i\cap\partial\Omega_j \text{ and } \Omega_i\in\Omega_G \text{ and } \Omega_j\in\Omega_G, \\ [\boldsymbol{u}] = 0, \;\; \boldsymbol{x}\in\partial\Omega, \end{cases} \tag{2}$$

where $\boldsymbol{u}$ is the velocity, p is the pressure, g is the gravity, ρ_i is density of the corresponding volume, γ is the surface tension coefficient, and κ is the surface curvature on the gas-liquid interfaces. Equation (1) is derived from the Navier-Stokes equations by dropping the viscosity term and accounting for variations in fluid density across different regions. In Equation (2), the first row represents the pressure jump caused by the surface tension across the interface between the liquid and gas region. The second row corresponds to the interface between two non-coalescence gas regions. Since there are two gas-liquid interfaces between these gas regions, we account for the surface tension contribution twice in the pressure jump. The third row is the zero Dirichlet boundary condition of the velocity at the domain boundary.

3.3 Interfacial Flow Model

On a thin film or a liquid surface, we assume that the tangential material transport is independent of the volumetric multiphase flow. Thus, we decouple the tangential surface of the film from the volumetric dynamics and follow [4] to model the tangential flow as:

$$\begin{cases} \dfrac{D\boldsymbol{u}^\top}{Dt} = -\dfrac{2\bar{R}T}{\rho\eta}\nabla_s\Gamma + \dfrac{1}{\rho}\boldsymbol{g}^\top \\ \dfrac{D\Gamma}{Dt} = -\Gamma\nabla_s\cdot\boldsymbol{u} \\ \dfrac{D\eta}{Dt} = -\eta\nabla_s\cdot\boldsymbol{u} \end{cases} \quad \boldsymbol{x}\in\partial\Omega \tag{3}$$

where $\boldsymbol{u}^\top$ is the tangential velocity of the interface, $\bar{R}$ is the gas constant, T is the temperature, η is the thickness, Γ is the surfactant concentration. $\boldsymbol{g}^\top$ is the component of the gravity along the surface. The first row denotes the tangential force acting on the velocity, including the gravity and the force caused by the tangential gradient of surface tension. The following two rows correspond to the conservation equations

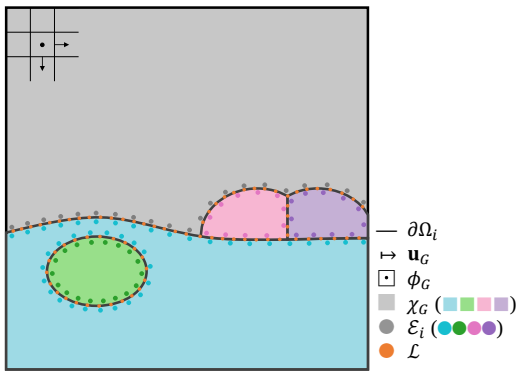


Fig. 3. The illustration of our MLSLS particle system. Our MLSLS particle system includes two major components: a set of MLSLS particles $\mathcal{E}$ and a background grid $\mathcal{G}$. (1) The particles $\mathcal{E}$ are divided into multiple groups $\mathcal{E}_i$ to track the surface of each region Ω_i. (2) On the background grid $\mathcal{G}$, we represent the implicit surface with a indicator map χ_G and a global level set ϕ_G. A velocity field $\boldsymbol{u}_G$ is stored on the staggered grid. An extra set of Lagrangian particles $\mathcal{L}$ is introduced on the interface to track the interfacial flow across different regions.

Symbol	Meaning
$\boldsymbol{\Omega}_i$	Region with index i
$\partial\boldsymbol{\Omega}_i$	Interface of Ω_i
$(\cdot)_{i/j}$	Attributes related to Ω_i or Ω_j
$\mathcal{E}$	MLSLS particles
$\mathcal{E}_i$	MLSLS particles of a region Ω_i
$\boldsymbol{R}$	Local frame
$(\cdot)_E$	Attributes related to $\mathcal{E}$
$\mathcal{G}$	Background grid
χ_G	Indicator map on $\mathcal{G}$
ϕ_G	Global level set on $\mathcal{G}$
$\boldsymbol{u}_G$	Velocity field on $\mathcal{G}$
ϕ_i	Temporary narrow band level set of Ω_i
$\mathcal{L}$	Extra Lagrangian particles on interface
$(\cdot)_L$	Attributes related to $\mathcal{L}$
$(\cdot)_c$	Values related to a grid cell c
$(\cdot)_{p/q}$	Values related to a particle p or q
$(\cdot)^{\top/\perp}$	Tangential or normal component of a vector

TABLE 1
A list of symbols and expressions in our paper.

for the surfactant concentration Γ and the film thickness η, respectively.

Note that this independent interfacial flow model relies on the assumption of a free-slip condition between the interfacial and volumetric flows. Under this assumption, the coupling of tangential velocities on the interfaces is neglected. While this assumption yields visually-appealing results in our experiments, it may introduce artifacts in scenarios characterized by widely varying tangential velocities across the interfaces, such as blowing bubbles.

4 Geometric Representation

The interface geometry we aim to represent is depicted in Figure 3 as the solid curves. It includes two types of boundaries: (1) the boundaries that separate fluid and air regions, such as the free surface between a liquid volume and the ambient air, or the surface of an underwater bubble, and (2) the liquid films between adjacent air regions, such as the membranes between neighboring bubbles. To conceptualize these boundaries, we can consider each piece as the joining of two boundaries from its adjacent regions. The interface geometry can then be described as the union of the boundaries of all regions in the domain, assuming these boundaries have negligible thickness. Motivated by this concept, we have developed a novel hybrid particle-grid structure to represent the interfaces of bubbles and foam in a multiphase flow system. In the following sections, we will introduce the discretization of this structure and outline the procedures for updating its geometry and topology. We refer readers to Table 1 for the relevant symbols.

4.1 Interface Discretization

4.1.1 MLSLS Particles

The fundamental concept of our particle representation is to allocate a dedicated particle system to track each region's boundary (on the interior side). In particular, each volumetric region Ω_i is associated with a distinct collection of particles $\mathcal{E}_i$. For example, as depicted in Figure 3, we employ five particle systems to trace the boundaries of five distinct regions within the domain: one for water, one for ambient air, one for an underwater bubble, and two for floating bubbles. The motion of particles representing a specific region's boundary reflects the motion of that region. It is ensured that the boundaries of different regions never intersect with each other. This concept draws inspiration from [4], where each bubble was represented by a separate set of SPH particles, and their interactions were governed by the localized interactions among particles from different sets.

The particles play a dual role in our geometric representation. First, they provide discrete moving least-squares stencils to differentiate the local surface. By applying the MLS method to these particles, we can fit a polynomial that describes the local geometry of the surface. This polynomial representation conveniently allows us to calculate various geometric properties, such as signed distance, mean curvature, and metric tensor, directly from local particles. These geometric properties are particularly useful for simulating the details of interfacial flow (see Section 5.2). Second, these particles serve as interface trackers, capturing the evolution of the level set associated with the region they belong to. In each time frame, we can conveniently extend the particles' local MLS approximation to a volumetric distance field, which can be further used to accommodate the volumetric multiphase flow solver in the global domain (see Section 5.1). Given their dual roles in moving least-squares and level-set tracking, we refer to these particles as Moving *Least-Squares/Level-Set (MLSLS)* particles.

4.1.2 Background Grid

In addition to the MLSLS particles, we also leverage a background staggered grid structure to store the volumetric fields extended from these particles. In particular, we construct two fields on the background grid, including an indicator map χ_G and a global level set ϕ_G, to represent the domain division and the implicit interface among a set of bubbles and their surrounding liquid and air mediums. The indicator map χ_G is a field of integer values, assigning a region index to each cell, indicating its corresponding region.

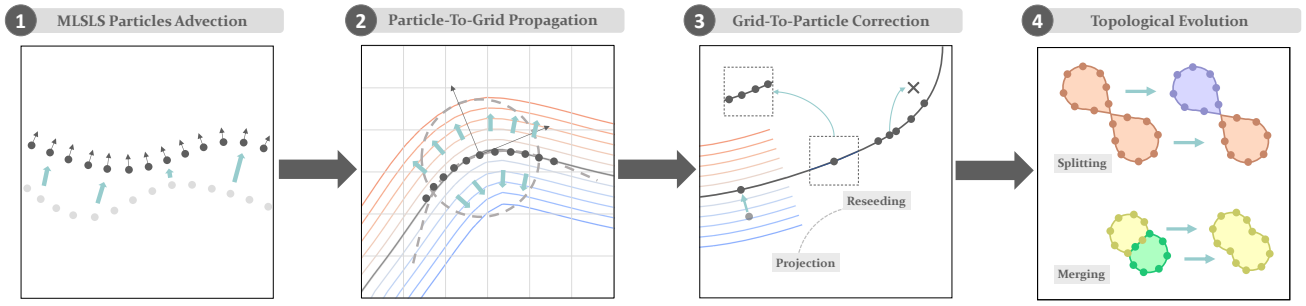


Fig. 4. The process of geometric evolution of our MLSLS particles system. (1) Advect MLSLS particles $\mathcal{E}$ using the velocity field $\boldsymbol{u}_G$; (2) Reconstruct the level-set-based implicit interfaces on the grid $\mathcal{G}$ using local MLS surfaces approximated on $\mathcal{E}$; (3) Correct particles $\mathcal{E}$ based on the grid $\mathcal{G}$ to achieve consistent interface representation between grid and particle and a optimal distribution. (4) Handle topological changes between regions including splitting and merging.

Without loss of generality, we assume that the global level set ϕ_G exclusively comprises negative values, symbolizing the negative distance from the cell to its nearest interface. Given a grid cell $\boldsymbol{x}_c$ lying within a region Ω_i, its indicator value would be $\chi_G(\boldsymbol{x}_c) = i$ and $\phi_G(\boldsymbol{x}_c)$ is the negative distance to the interface of $\partial\Omega_i$. Besides, we store velocities $\boldsymbol{u}_G$ on the staggered grid faces to specify fluid flow in the global domain.

Last, we want to emphasize that both the particle systems and the background grid are used to represent the same interface geometry, with different computational focuses. While the particle systems provide a local and codimensional representation of the interface through moving least-squares, the background grid offers a global and volumetric discretization via advancing the level-set field. These two components complement each other and together enable us to accurately capture and simulate the complex behavior of the geometric and topological evolution in a multiphase flow system.

4.2 Geometric Evolution

One of the fundamental challenges in our hybrid particle-grid representation is determining how the Lagrangian and Eulerian components co-evolve and influence each other. This section presents our geometric evolution scheme, which facilitates updating the particles and the grid. The core idea of our scheme revolves around a *bi-directional calibration* process. First, we propagate the local Moving Least-Squares (MLS) approximation of the interface geometry from the particles to the entire background grid, generating an unsigned distance field that spans the entire domain. Second, we leverage this distance field to redistribute the particles on the surface, ensuring an optimal distribution of particles without accumulation or sparsity.

This bi-directional calibration process consists of four stages, including *particle advection*, *particle-to-grid propagation*, *grid-to-particle correction*, and *topological evolution*, as depicted in Figure 4. (1) We begin with the *particle advection* stage, where MLSLS particles $\mathcal{E}$ are advected using the velocity field $\boldsymbol{u}_G$. (2) In the *particle-to-grid propagation* stage, we extend the interface defined by local MLS surfaces on the updated MLSLS particles $\mathcal{E}$ to the grid $\mathcal{G}$, which is done by reconstructing a set of regional level sets Ω_i based on the MLS surfaces approximated on MLSLS particles $\mathcal{E}_i$ and using them to reconstruct the implicit interface on the global level set ϕ_G and the indicator map χ_G. (3) In the *grid-to-particle correction* stage, we calibrate particles $\mathcal{E}$ based on the implicit interface defined on the $\mathcal{G}$, which is done by projecting MLSLS particles onto the implicit interface and optimizing their distribution through particle reseeding on the implicit interface. (4) In the *topological evolution* stage, we detect the splitting and merging events based on grid $\mathcal{G}$ and address the region update on particles $\mathcal{E}$ and grid $\mathcal{G}$. In the upcoming sections, we will provide detailed explanations of each stage and its implementation.

4.2.1 Particles Advection

For each region Ω_i, we advect the positions of its MLSLS particles $\mathcal{E}_i$ using the background velocity field $\boldsymbol{u}_G$. In addition, we also update each particle's normal vector following Ianniello and Di Mascio [37] as $D\boldsymbol{n}/Dt = -\nabla\boldsymbol{u}\cdot\boldsymbol{n}$. This predicted normal is utilized in the particle-to-grid propagation stage to generate the local frame required for MLS fitting. It will be replaced by the normal derived from the grid $\mathcal{G}$ in the grid-to-particle correction step;

4.2.2 Particle-To-Grid Propagation

In this stage, we extend the local MLS surface approximated on particles $\mathcal{E}$ to reconstruct an implicit interface representation on the grid $\mathcal{G}$. In particular, we will update the global level-set field ϕ_G and the indicator field χ_G on the grid based on the particle interfaces. To realize this goal, we first rebuild a regional level set ϕ_i within a narrow band for each region Ω_i based on the MLS surfaces approximated on particles $\mathcal{E}_i$. Then, we reconstruct the implicit interface by combining all regional level sets ϕ_i to update the global level set ϕ_G and the indicator map χ_G.

Regional Level Set Reconstruction: For each region, we rebuild a regional level set ϕ_i using MLSLS particles $\mathcal{E}_i$ within a narrow band of width $r_N = 4\Delta x$. We illustrate the algorithm in Algorithm 1. As shown in Figure 5, given a cell $\boldsymbol{x}_c$, this process begins by identifying its closest particle $\boldsymbol{x}_p$ and its normal $\boldsymbol{n}_p$ in $\mathcal{E}_i$ within a radius r_{CP}. Then, we construct a local MLS surface based on neighboring particles $\mathcal{E}_{NB}$ at $\boldsymbol{x}_p$. To ensure that only relevant particles contributing to the surface are considered, we selectively include particles q in $\mathcal{E}_{NB}$ whose normals $\boldsymbol{n}_q$ satisfying $\boldsymbol{n}_p \cdot \boldsymbol{n}_q > 0$. Next, we employ the Newton method to project

Algorithm 1 Reconstruct the regional level set ϕ_i using MLSLS particles $\mathcal{E}_i$

for each cell $\boldsymbol{x}_c$ **do**
  Find its closest particle $p \in \mathcal{E}_i$ at $\boldsymbol{x}_p$ with its normal $\boldsymbol{n}_p$
  if $|\boldsymbol{x}_c - \boldsymbol{x}_p| > r_{CP}$ **then**
    Continue
  end if
  Fit local MLS surface at $\boldsymbol{x}_p$ with neighbors $\mathcal{E}_{NB}$
  Project $\boldsymbol{x}_c$ onto the MLS surface as $\boldsymbol{x}_{proj}$
  Adjust $\boldsymbol{x}_{proj}$ if it is distant from the MLS surface
  Compute the normal $\boldsymbol{n}_{proj}$ at $\boldsymbol{x}_{proj}$ on the MLS surface
  if $|\boldsymbol{x}_c - \boldsymbol{x}_{proj}| < |\boldsymbol{x}_c - \boldsymbol{x}_p|$ **then**
    $\phi_i(\boldsymbol{x}_c) \leftarrow \text{Sign}(\boldsymbol{n}_p \cdot (\boldsymbol{x}_c - \boldsymbol{x}_{proj})) \cdot |\boldsymbol{x}_c - \boldsymbol{x}_{proj}|$
  else
    $\phi_i(\boldsymbol{x}_c) \leftarrow \text{Sign}(\boldsymbol{n}_p \cdot (\boldsymbol{x}_c - \boldsymbol{x}_p)) \cdot |\boldsymbol{x}_c - \boldsymbol{x}_p|$
  end if
end for

$\boldsymbol{x}_c$ onto the surface, obtaining the projected point $\boldsymbol{x}_{proj}$. Its normal $\boldsymbol{n}_{proj}$ is derived based on the MLS surface. In certain cases involving sharp features, the projected point may be distant from the actual surface. To address this, we apply an adjustment to constrain the projected point based on the neighboring particles. Finally, the level set value $\phi_i(\boldsymbol{x}_c)$ is updated based on the distance between the cell $\boldsymbol{x}_c$ and the projected point $\boldsymbol{x}_{proj}$. Further details can be found in Appendix B.

Global Level Set reconstruction: In this step, we work on the grid to correct the level sets, ensuring seamless domain division. Then we build the global level set ϕ_G and the indicator map χ_G based on the updated level sets.

We follow [7] to correct the regional level sets ϕ_i to avoid vacuum or overlaps between regions. Specifically, if a cell $\boldsymbol{x}_c$ is covered by multiple regional level sets within their narrow bands, we identify two level sets with the minimal value, denoted as ϕ_i and ϕ_j ($\phi_i(\boldsymbol{x}_c) < \phi_j(\boldsymbol{x}_c) < \forall_k \phi_k(\boldsymbol{x}_c)$) and correct these regional level sets by subtracting their mean value $0.5\,(\phi_i(\boldsymbol{x}_c) + \phi_j(\boldsymbol{x}_c))$ at $\boldsymbol{x}_c$. We then reconstruct the global level set by taking the minimum value among the regional level sets within the narrow band as $\phi_G(\boldsymbol{x}_c) = \min_i(\phi_i(\boldsymbol{x}_c))$.

With the CFL condition, we ensure that the particle displacements are always smaller than half the width of the narrow band r_N within a single time step. This ensures that the interfaces only exist within the narrow band after the advection and allows us to update the indicator map only in the narrow band using $\chi_G(\boldsymbol{x}_c) = \arg\min_i(\phi_i(\boldsymbol{x}_c))$. To obtain the global level set ϕ_G throughout the entire domain, we perform the fast marching algorithm within the interior of each region.

4.2.3 Grid-To-Particle Correction

To ensure geometric consistency between the particle sampled interfaces represented by particles $\mathcal{E}$ and the implicit interfaces defined on the grid $\mathcal{G}$, a two-step grid-to-particle correction is performed, including particle projection and particle reseeding. The particle projection ensures that all particles lie on the zero-contour of the regional level set,

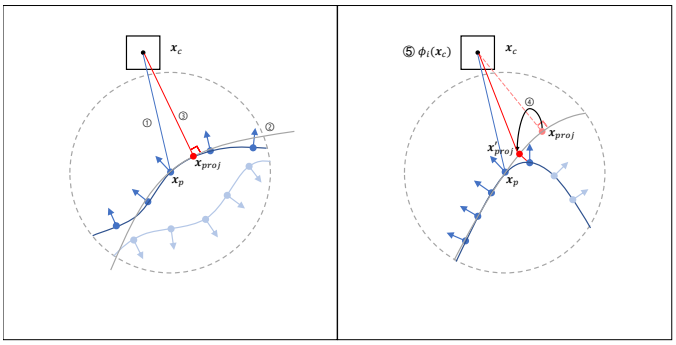


Fig. 5. From a cell $\boldsymbol{x}_c$ (square with a black dot), the regional level set value $\phi_i(\boldsymbol{x}_c)$ is reconstructed as follows: (1) Search the closest point $\boldsymbol{x}_p$ (blue line); (2) Fit the local MLS surface (grey line) with particle neighbors $q \in \mathcal{E}_{NB}$ (dark blue dots); (3) Project $\boldsymbol{x}_c$ onto the MLS surface as $\boldsymbol{x}_{proj}$ (red dot); (4) Adjust the outside projected point $\boldsymbol{x}_{proj}$ (black arrow in the right subfigure); (5) Update the level set value $\phi_i(\boldsymbol{x}_c)$.

while the particle reseeding aim to achieve an optimal sampling on the interface.

Particle Projection: In this step, we calibrate the particles $\mathcal{E}_i$ to align with the interface of the regional level set ϕ_i. We first perform the fast marching algorithm on corrected regional level sets ϕ_i within the narrow band. Then, MLSLS particles $p \in \mathcal{E}_i$ are projected onto zero-contours of ϕ_i using

$$\boldsymbol{x}_p = \boldsymbol{x}_p - \phi_i(\boldsymbol{x}_p) \cdot \frac{\nabla \phi_i(\boldsymbol{x}_p)}{|\nabla \phi_i(\boldsymbol{x}_p)|}. \tag{4}$$

We also update particle normals as the normalized gradient of the regional level sets ϕ_i.

Particle Reseeding: To maintain an optimal particle distribution and density of the surface, we implement a particle reseeding algorithm for MLSLS particles $\mathcal{E}_i$ in the region Ω_i. In the insertion stage, we employ a FLIP-like algorithm. For each region Ω_i, we check the cells close to the interface with $|\phi_i(\boldsymbol{x}_c)| < \Delta x$. If the particle count within the cell is below a predefined threshold $n_{seed}/2$, we remove the existing particles in that cell, randomly generate and insert n_{seed} particles, and project them onto the interface using ϕ_i. In practice, we use $n_{seed} = 8$ and 16 in 2D and 3D, respectively. In the removal stage, when two particles are in close proximity ($|\boldsymbol{x}_p - \boldsymbol{x}_q| < r_{RM}$), we remove the one with a larger number density (see in Appendix C.2), as it resides in a denser neighborhood.

4.2.4 Topological Evolution

Our particle-grid representation supports two types of topological change to simulate bubble split and merge. Given the grid-based representation's ability to handle topological changes effectively, we begin by detecting splitting and merging on the indicator map χ_G. Then, we update the grid representation accordingly and calibrate the particles to align with the updated interface on the grid.

Splitting: In the splitting step, we first perform the flood fill algorithm on the indicator map χ_G for each region in parallel to identify any potential splitting. If multiple connected volumes are detected in region Ω_i, we assign them as new separate regions. We then update the indicator map χ_G by filling the cells of Ω_i with the new region indices. The global level set ϕ remains unchanged during the splitting process.

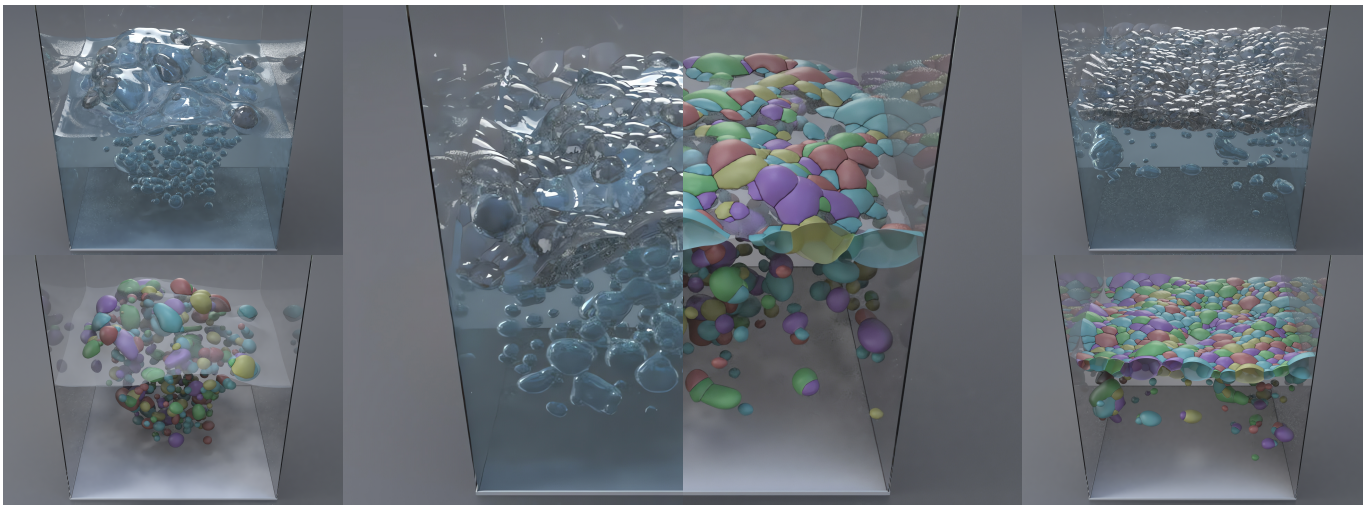


Fig. 6. Massive bubbles generated underwater, rising and clustering on the liquid surface.

The MLSLS particles $\mathcal{E}_i$ are reassigned to their nearest new region and projected onto their interfaces, which is obtained by extracting the corresponding connected part from the regional level set ϕ_i and performing a re-marching within the narrow band.

Merging: The merging of regions happens when two adjacent regions Ω_i, Ω_j $(i < j)$ contain the same phase of liquid, which is identified on the indicator map χ_G. We select the smaller index i as the merged index and update the indicator map χ_G by replacing j with i. In the merged region, we merge two level sets as $\phi_{ij}(\boldsymbol{x}) = \min(\phi_i(\boldsymbol{x}), \phi_j(\boldsymbol{x}))$ and then update the global level set ϕ_G as $\phi_G(\boldsymbol{x}) = \phi_{ij}(\boldsymbol{x})$ where $\chi_G(\boldsymbol{x}) = i$. To remove the particles on the interface between two merged regions $\partial\Omega_i \cap \partial\Omega_j$, we perform neighbor searches on the $\mathcal{E}_i$ and $\mathcal{E}_j$ with the radius $r_{MRG} = 0.75\Delta x$. Any particles that detect neighboring particles from the other region are then removed.

5 Simulation Framework

After presenting the geometry representation and evolution of our MLSLS particle system, we will now demonstrate its integration into the simulation framework.

As discussed in Section 3, we solve the volumetric multiphase flow and the interfacial flow separately. Within our grid-particle approach, we can seamlessly integrate our method into the existing simulation framework for both volumetric flow and interfacial flow. For the multiphase flow, which is discussed in Section 3.2, we discretized it on the background grid $\mathcal{G}$ following the traditional Eulerian framework. With the domain division and the implicit interface defined on the background grid $\mathcal{G}$, we solve a pressure projection derived from Equation (1). For the interfacial flow, we utilize Equation (3) to solve the tangential dynamics of the surfactants on MLSLS particles $\mathcal{E}_i$, along with a set of auxiliary Lagrangian particles $\mathcal{L}$. We tackle the tangential flow in a pure particle approach following [4], where the interface is approximated and differentiated on MLSLS particles $\mathcal{E}_i$ and flow tracked by Lagrangian particles $\mathcal{L}$.

We would like to emphasize our strategy of solving the surface tension in our method. We decouple it into its normal and tangential components, addressing them independently in two distinct flows. The normal component of the surface tension, contributing to the bubble deformation and liquid-bubble interaction, is typically incorporated into the volumetric Eulerian fluid solver as a pressure jump on grid faces. In the volumetric multiphase flow, we adopt a similar strategy. We enforce the normal surface tension on the grid face and approximate the curvature using the MLS method on Eulerian particles $\mathcal{E}$. In the interfacial flow, the tangential surface tension, as presented in Equation (3), is modeled as a tangential acceleration resulting from the surface gradient of the surfactant concentration within the thin film on the interface. We follow [4] to resolve the surface tension on MLSLS particles $\mathcal{E}$ with a set of SPH-based surface operators. Next, we will explore the specific details of the two flows.

5.1 Volumetric Multiphase Flow

Following the physical model discussed in Section 3.2, we solve the dynamics of the volumetric multiphase flow throughout the entire domain using the conventional Eulerian framework. With the discretization on the grid $\mathcal{G}$, we effectively solve the physical forces in Equation (1), including gravity, pressure, and surface tension. The gravity is explicitly applied to all grid faces. The pressure is resolved through the divergence-free projection, which considers discontinuous density across different regions, to achieve a divergence-free velocity field. To account for surface tension, we utilize MLSLS particles to approximate it and incorporate it into the pressure projection as a jump condition on the grid faces of the interfaces. Next, we will discuss the details of the divergence-free projection and surface tension components.

Divergence-Free Projection: We solve the pressure projection equation derived from Equation (1) to enforce the incompressibility across the entire domain

$$\begin{cases} \boldsymbol{u}_G^* = \boldsymbol{u}_G - \dfrac{\Delta t}{\rho}\nabla p + \delta_s(\boldsymbol{x})\dfrac{\Delta t}{\rho}k\gamma\kappa \\ \nabla \cdot \boldsymbol{u}_G^* = 0 \end{cases} \tag{5}$$

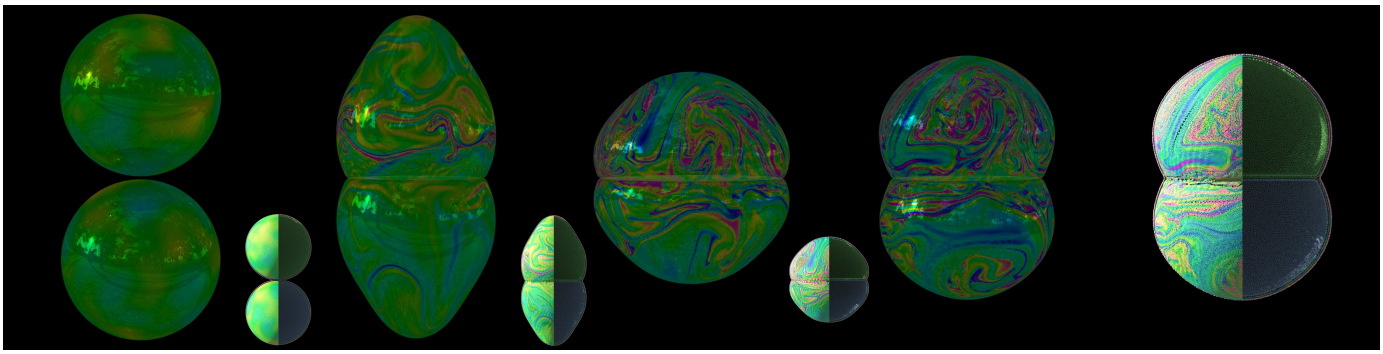


Fig. 7. When two bubbles collide, the interfacial flow across their interfaces results in a complex pattern, as depicted in the photo-realistic rendering. Additionally, a particle view is provided with colored Lagrangian particles $\mathcal{L}$ (left), and the MLSLS particles $\mathcal{E}$ of each region(right).

where Δt is the time step, $\boldsymbol{u}_G^*$ is the velocity field after pressure projection, p is the pressure, ρ is the fluid density at the grid faces, γ is the surface tension coefficient, κ is the surface curvature, and $\delta_s(\boldsymbol{x})$ is a Dirac function on the interface. k denotes the number of liquid-gas interfaces for surface tension. As discussed in Section 3.2, $k = 1$ when the interface is between liquid and gas regions and $k = 2$ when the interface is between two gas regions. We adopt the divergence control method proposed in [27] to ensure volume conservation.

To address the discontinuous distribution of the physical coefficients, e.g., fluid density, we follow [7] to model the mixed coefficient across the interface. We identify a grid face $\boldsymbol{x}_{c+1/2}$ as the interface when its neighboring cells are in different regions $\chi_G(\boldsymbol{x}_c) \neq \chi_G(\boldsymbol{x}_{c+1})$. Suppose $\chi_G(\boldsymbol{x}_c) = i$, $\chi_G(\boldsymbol{x}_{c+1}) = j$, we define $\theta = \phi_G(\boldsymbol{x}_c)/(\phi_G(\boldsymbol{x}_c) + \phi_G(\boldsymbol{x}_{c+1}))$ to represent the volume fraction of region j in the control volume of face $\boldsymbol{x}_{c+1/2}$. The interface position is then interpolated as $\boldsymbol{x}_\theta = (1-\theta)\boldsymbol{x}_c + \theta\boldsymbol{x}_{c+1}$. The mixed density on the face is computed as $(1-\theta)\rho_i + \theta\rho_j$, where ρ_i and ρ_j represent the densities of regions i and j.

Surface Tension: We incorporate the surface tension into the solver as pressure jumps and approximate the curvature based on MLSLS particles from adjacent regions on both sides of the interface. Suppose $\boldsymbol{x}_\theta$ lies between Ω_i and Ω_j, we estimate two curvature κ_i, κ_j from both sides of the interface by performing the MLS approximation on $\mathcal{E}_i$ and $\mathcal{E}_j$ at $\boldsymbol{x}_\theta$. The curvature at $\boldsymbol{x}_\theta$ is then determined as the average of κ_i and κ_j. By enforcing the surface tension on the grid face and considering the curvature from both sides of the interfacial grid face, our method effectively handles the non-manifold joint where multiple interfaces are involved, such as triplet joints between bubbles.

The surface tension coefficient is determined as $\gamma = \gamma_0 - \bar{R}T\Gamma$, where Γ represents the surfactant concentration on MLSLS particles $\mathcal{E}$, which we will discuss in Appendix C.2. For simulations without the interfacial flow, we simply use the default coefficient γ_0.

5.2 Interfacial Flow

We follow the physical model in Section 3.3 to solve the interfacial flow on MLSLS particles $\mathcal{E}_i$, which involving the evolution of surfactant concentration Γ and thin film thickness η on the interface. We discretize the model on our MLSLS particles, following the particle-only approach proposed in [4], where we approximate the MLS surface and build SPH-based surface differential operators on the MLS particles. The tangential dynamics is then solved on MLSLS particles $\mathcal{E}_i$ iteratively using the IISPH method. Additionally, we introduce an additional set of particles to track the surfactant distribution on the interface and transfer the quantities to MLSLS particles for solving.

MLSLS Discretization: By reformulating and discretizing Equation (3), we obtain an implicit equation of the interfacial dynamics of surfactant

$$\left(-\frac{1}{\Delta t\Gamma}\right)\Gamma^* + \left(\Delta t\frac{\bar{R}T}{\rho}\nabla\frac{1}{\eta}\right)\cdot\nabla_s\Gamma^* + \left(\Delta t\frac{\bar{R}T}{\rho}\frac{1}{\eta}\right)\nabla_s^2\Gamma^* = \nabla\cdot\boldsymbol{u}^\top - \frac{1}{\Delta t} + \Delta t\left(\nabla\frac{1}{\rho}\cdot\boldsymbol{g}^\top\right) \tag{6}$$

and a tangential velocity correction equation

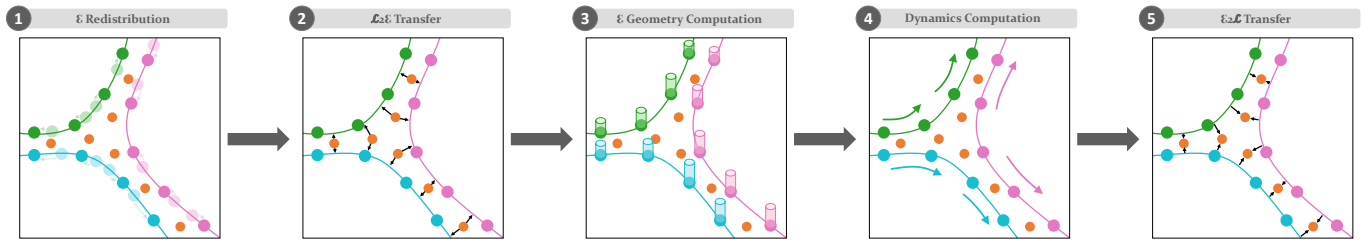


Fig. 8. The process of interfacial flow simulation. (1) Redistribute MLSLS particles $\mathcal{E}$ within each region. (2) Transfer physical quantities from Lagrangian particles $\mathcal{L}$ to MLSLS particles $\mathcal{E}$ in their closest two regions. (3) Compute interfacial geometry. (4) Solve tangential dynamics within each region separately. (5) Transfer quantities from MLSLS particles $\mathcal{E}$ to Lagrangian particles $\mathcal{L}$.

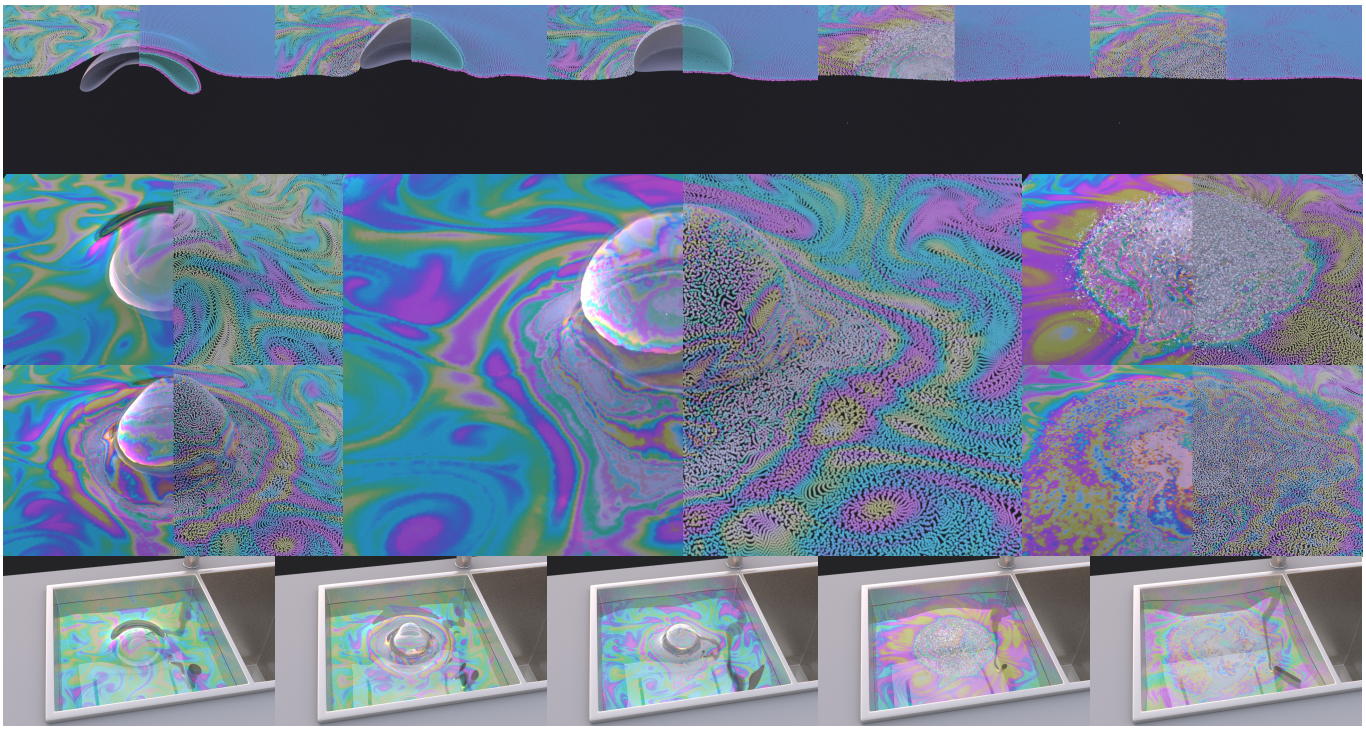


Fig. 9. The life-cycle of a bubble. In a liquid tank covered by a thin layer of the surfactant, a submerged bubble ascends, settles on the surface, and eventually ruptures. This process highlights the intricate coupling between the volumetric and surface dynamics in our method. *Top:* Sectional view showcasing the underlying Lagrangian particles $\mathcal{L}$ (upper left corner), MLSLS particles $\mathcal{E}_i$ (upper right corner), and bubble mesh rebuilt on the grid $\mathcal{G}$. The MLSLS particles $\mathcal{E}_i$ of the ambient air, bubble, and liquid bulk are colored in blue, purple, and green, respectively. *Middle:* Close-up view of the photo-realistic rendering (left) and Lagrangian particles $\mathcal{L}$ (right). *Bottom:* Long-shot view capturing the entire scenario.

$$\boldsymbol{u}^{\top *} = \boldsymbol{u}^{\top} - \frac{2\bar{R}T}{\rho\eta^*}\nabla_s\Gamma^* + \frac{1}{\rho}\boldsymbol{g}^{\top}. \tag{7}$$

where * denotes the updated quantities. Note that in these equations, the tangential velocity $\boldsymbol{u}^{\top}$ describes the tangential motion within the thin film and is independent of the volumetric velocity field.

Interfacial Enhancement: We solve the interfacial dynamics using a hybrid particle-particle approach on MLSLS particles $\mathcal{E}_i$ following MELP [4]. We introduce an extra set of Lagrangian particles $\mathcal{L}$ to track the distribution of interfacial physical properties. These Lagrangian particles $\mathcal{L}$ are positioned on the region interfaces and advect along the interfaces, carrying various physical quantities (see Appendix C.1). They interact with MLSLS particles $\mathcal{E}_i$ to exchange the physical properties on the interface in $\mathcal{L}2\mathcal{E}$ and $\mathcal{E}2\mathcal{L}$ transfer steps using the SPH-based interpolator.

Note that in contrast to MELP, where Lagrangian particles are bound to specific regions, our method allows them to freely move along the interfaces of any region. With both enclosed regions and the region of ambient air tracked, our interfacial enhancement operates under the assumption that each Lagrangian particle resides between the interfaces of two distinct regions. Each Lagrangian particle contributes to the interfacial flows of two regions and, in turn, is influenced by them. Therefore, at the beginning of the process, we transfer the quantities from Lagrangian particles $\mathcal{L}$ to the MLSLS particles $\mathcal{E}$ of their two closest regions. After computing interfacial dynamics on each region, we transfer the updated quantities backward and finally update the Lagrangian particles $\mathcal{L}$.

The overall process for solving the interfacial flow is depicted in Figure 8 and summarized as follows:

(1) $\mathcal{E}$ ***Redistribution:*** Solve a pseudo-pressure equation to redistribute $\mathcal{E}_i$ particles on the interfaces, to ensure a uniform distribution;

(2) $\mathcal{L}2\mathcal{E}$ ***Transfer:*** Transfer the physical quantities from Lagrangian particles $\mathcal{L}$ to MLSLS particles $\mathcal{E}_i$ of their two closest regions;

(3) Geometry Computation: Compute the geometry-related properties, including domain area a, thickness η, surface concentration of surfactant Γ on each region; fit MLS surfaces on $\mathcal{E}_i$ to build codimension-1 differential operators, including surface gradient ∇_s, surface divergence $\nabla_s\cdot$, and surface Laplacian ∇_s^2 on each region;

(4) Interfacial Flow Equation Computation: Solve Equation (6) and Equation (7) on MLSLS particles $\mathcal{E}_i$ using IISPH method on each region;

(5) $\mathcal{E}2\mathcal{L}$ ***Transfer:*** Transfer the updated tangential velocity $\boldsymbol{u}^{\top}$ and its divergence $\nabla_s \cdot \boldsymbol{u}^{\top}$ backward from MLSLS particles $\mathcal{E}_i$ to Lagrangian particles $\mathcal{L}$; update the thickness η for Lagrangian particles based on the transferred divergence.

For more detailed information, please refer to Appendix C.

5.3 Time Integration

The overall time integration of our approach includes the following steps:

(1) MLSLS Particles Advection: Advect MLSLS particles $\mathcal{E}$ (Section 4.2.1); Advect Lagrangian particles (Appendix C.1, optional);

(2) Particle-To-Grid Propagation: Reconstruct regional level sets ϕ_i based on MLS surfaces fitted on $\mathcal{E}$ and then reconstruct the implicit interface on the global level set ϕ_G and indicator map χ_G (Section 4.2.2);

(3) Grid-To-Particle Correction: Project particles $\mathcal{E}_i$ onto regional level sets ϕ_i and reseed particles $\mathcal{E}_i$ for all regions (Section 4.2.3);

(4) Topological Evolution: Address the topological changes including splitting and merging (Section 4.2.4);

(5) Velocity Advection: Advect velocity field $\boldsymbol{u}_G$ using the MacCormack method.

(6) Volumetric Multiphase Flow Solving: Solve the volumetric multiphase flow in Equation (1) on grid $\mathcal{G}$ with the surface tension approximated on MLSLS particles $\mathcal{E}$ (Section 5.1);

(7) Interfacial Flow Solving (Optional): Solve the interfacial flow in Equation (3) on MLSLS particles $\mathcal{E}$ using interfacial properties transferred from Lagrangian particle $\mathcal{L}$ (Section 5.2);

6 Results

We demonstrate the capabilities of our solver through a series of simulations. The configuration and running time of the simulations are provided in Table 3 which shows that all the simulations run ranging from 11 seconds (Four bubbles 3D) to 3.8 minutes (Rising bubbles 3D) per timestep, depending on the scene complexity. For visualization, we perform marching cube on regional level sets ϕ_i to extract the region surfaces and use ParaView for rendering all 2D simulations. 3D results are rendered using Houdini. We use ColorPy [38] to compute the interference color of the thin film from its thickness. Additional validations for surface tracking and curvature estimation are detailed in Appendix D and E.

6.1 Validation

Plateau border: We validate our method on bubbles driven by Plateau's law, which has been studied in previous works [1], [4]. According to Plateau's law, which states that soap films always meet in groups of threes, forming an angle (dihedral angle in 3D) of precisely 120 degrees between films. To verify this, we initialize four bubbles that collide and adhere to each other. We visualize the results in Figure 11 (2D) and Figure 12 (3D). The corresponding data are summarized in Table 2, which shows that the error of the measured dihedral angle remains below 6% for all experiments. This outcome validates the accuracy of our method in simulating surface-tension-driven interfaces.

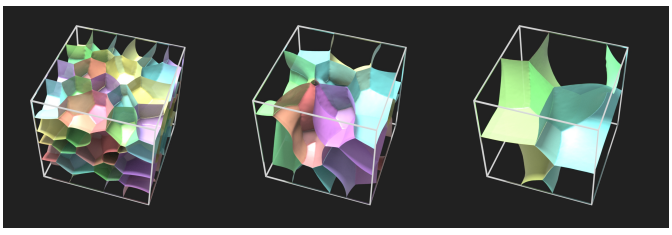


Fig. 10. Randomly initialized packed bubbles merging one by one. We color the surface mesh based on region indices.

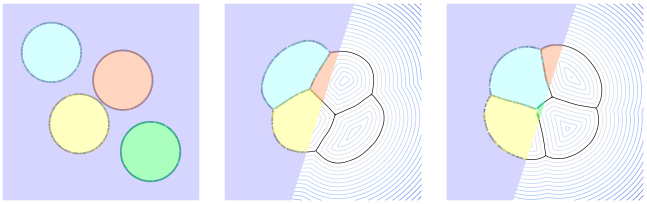


Fig. 11. Four 2D bubbles colliding together. We depict the indicator map χ (colored regions) and MLSLS particles $\mathcal{E}$ of each region (dark colored dots) on the left side; we also visualize the region surfaces (black lines) and the contour of the global level set ϕ (blue line) on the right side.

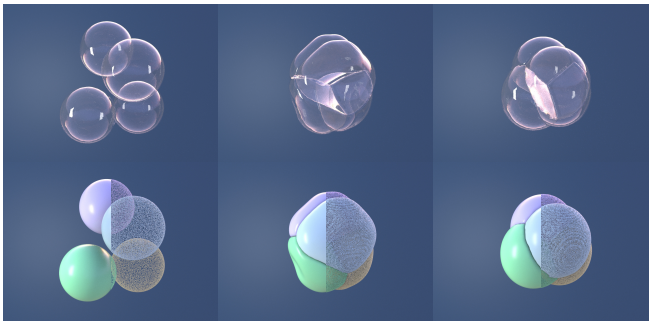


Fig. 12. Four 3D bubbles converging, adhering to each other, and forming the 120-degree angle at the joint. *Top:* Photorealistic rendering. *Bottom:* Colored surface mesh and MLSLS particles $\mathcal{E}$ of each region.

Merging bubbles: Next, we validate our approach by solving minimum surface problems as in [14], [29]. As shown in Figure 10, we initiate a configuration consisting of 128 bubbles of varying sizes, tightly packed within a cubic domain. At every 10 frames, we introduce perturbations to the system by merging the smallest bubble with one of its neighboring bubbles. As two bubbles merge, the whole system undergoes deformation driven by the surface tension, gradually transitioning towards a new equilibrium configuration that minimizes the overall surface area.

6.2 Examples

Droplet and bubbles: We first showcase our solver's capability in tackling multiphase systems consisting of both

Set-up	2D four bubbles					
Pairs	1-2	1-3(a)	2-3	1-4	3-4	1-3(b)
Angle	119.1	122.7	118.2	115.6	118.1	126.3
Error	0.7%	2.2%	1.5%	3.7%	1.6%	5.25 %
Set-up	3D four bubbles					
Pairs	1-2	1-3	1-4	2-3	2-4	3-4
Angle	115.0	121.8	120.3	121.7	124.5	113.8
Error	4.2%	1.5%	0.3%	1.4%	3.75%	5.2%

TABLE 2
Numerical results of Plateau Border example. In the 2D example, the labels 1-4 correspond to the green, yellow, blue, and orange bubbles shown in Figure 11. In the 3D example, we examine the dihedral angles between regions. The labels 1-4 correspond to green, yellow, purple, and blue bubbles shown in Figure 12.

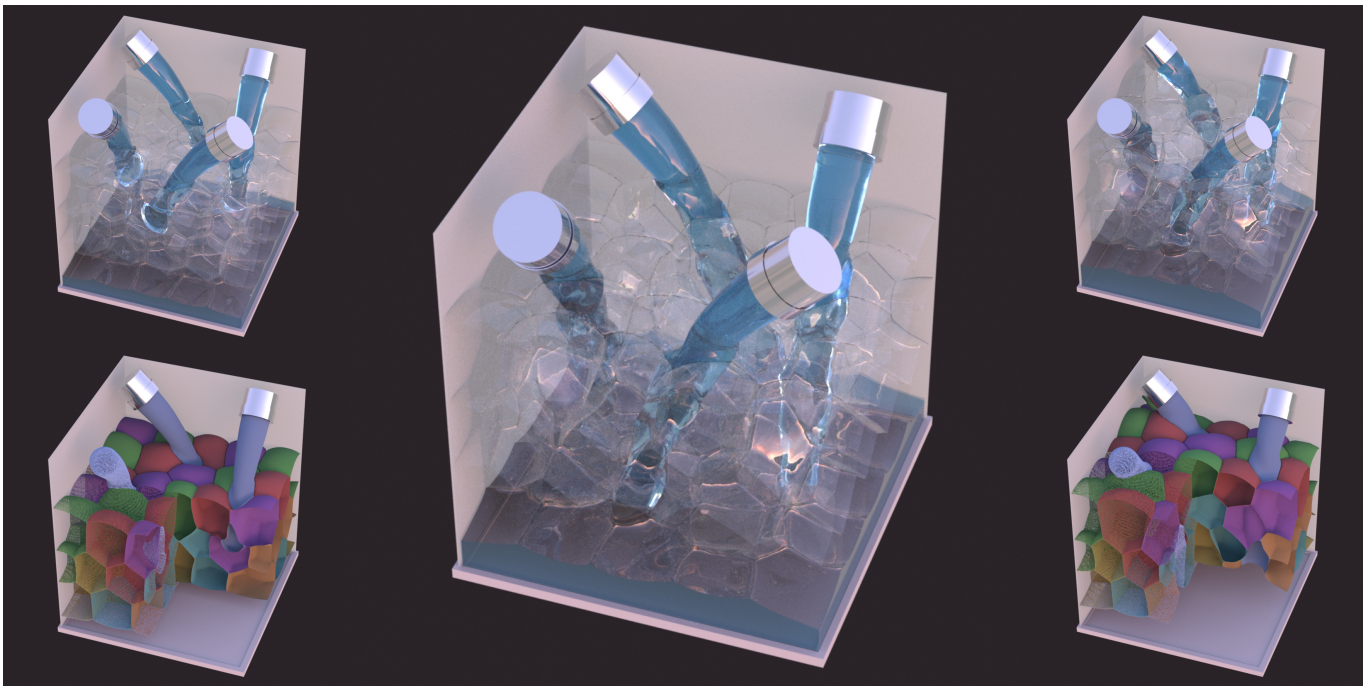


Fig. 13. Four jets emit above a stack of bubbles, flowing through the gaps between bubbles, resulting in the rise of the water line. *Top:* Photo-realistic rendering. *Bottom:* Colored surface mesh and MLSLS particles $\mathcal{E}$.

volumetric and interfacial components. As depicted in Figure 14, we begin with a liquid tank containing two submerged bubbles. Three other bubbles and a droplet are released and fall on the tank due to gravity. We show that our method successfully captures the topological changes during the simulation, including the merging of the droplet with the tank and the fragmentation of a bubble into smaller bubbles.

Jet on bubbles: Next, we show a similar example with a more drastic volumetric flow setting. As depicted in Figure 13 we initialize the scenario with a collection of tightly-packed bubbles in a cubic domain situated atop the surface of the liquid tank. Four pipes emit liquid jets above the bubbles, causing the liquid to flow through the gaps between bubbles and merge with the underlying liquid tank. As the simulation progresses, the water line steadily rises.

Rising bubbles: Next, we showcase our method's ability in tackling a large number of bubbles and their interactions and foam formulation in a multiphase environment. As shown in Figure 17 and Figure 6, we demonstrate the robustness and versatility of our method in capturing intricate foaming flow. We set up the simulation in a cubic aquarium by continuously generating underwater foams. The foams, driven by the buoyancy, rise, break through the liquid surface, adhere to other foam bubbles, and form stacked layers on top of the liquid. Our approach exhibits effectiveness in handling a large number of regions (approximately 300 regions in 2D, 800 regions in 3D) and the complex interaction and topological changes between them. Through the utilization of moving particles, we can accurately track and represent small-scaled foam even with a volume as small as several cells. The large-scale foaming flow has also been explored in a research paper by [5], where they achieved the simulation of 10000 bubbles by running on a supercomputer.

Bubble with interfacial flow: Next, we show our solver can facilitate complex interfacial flow details on a deforming surface, which is comparable to the previous literature of purely Lagrangian methods [2]–[4] . As shown in Figure 16, we set up an oscillated bubble with interfacial flow dynamics. We initialize the velocity field perturbed by Perlin noise and derive the tangential velocity of Lagrangian particles $\mathcal{L}$ from a vorticity field. Starting from a perturbed thickness field, driven by the surface tension, the bubble gradually oscillates and forms intricate vortex patterns on its surface. To enhance the swirling flow on the surface, we apply vorticity confinement to the interfacial velocity.

The life-cycle of a bubble: The life cycle of a bubble, which involves various visually captivating phenomena, has been extensively studied in prior research [39]–[41]. The complete life cycle of a bubble encompasses several stages [40]. Firstly, the bubble is generated underwater and begins to rise, interacting with the surrounding liquid along its

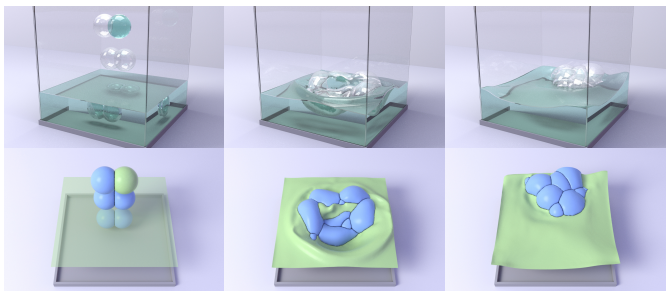


Fig. 14. The droplet, bubbles, and a liquid tank are initialized within a cubic domain, where they undergo collision, merging, and splitting. *Left:* Photo-realistic rendering. *Right:* surface mesh colored according to the fluid phase.

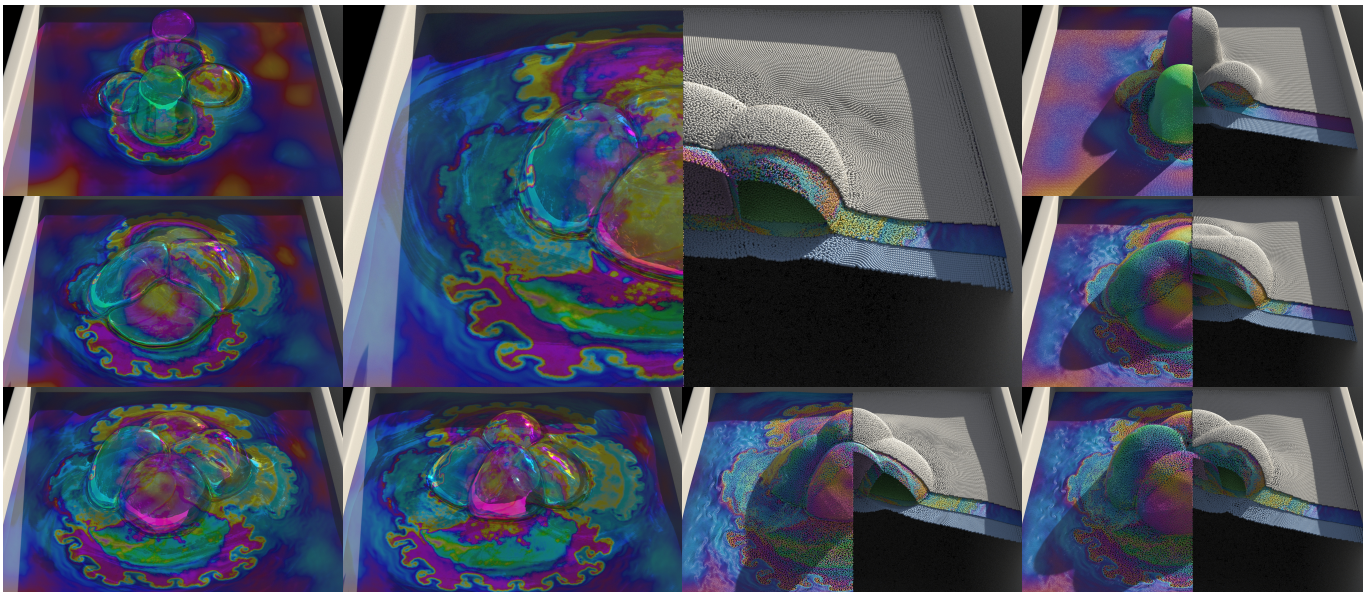


Fig. 15. Bubbles on a tank with multiregional interfacial flow. Four bubbles move and merge with the surfactant-coated surface of the liquid tank, where the surfactant flows across their interconnected interfaces and diffuses on the tank surface. Our method effectively addresses the multiregional volumetric and surface dynamics to reproduce this phenomenon. *Left:* The photorealistic rendering. *Right:* The colored Lagrangian particles $\mathcal{L}$ on the left half, and the sectional view of the underlying MLSLS particles $\mathcal{E}_i$ and Lagrangian particles $\mathcal{L}$ on the right half.

ascent. At the free surface, the thin film of the bubble starts to drain, gradually decreasing in thickness. Eventually, the film breaks at its thinnest point, causing the film to retract and fragment into numerous tiny droplets.

We reproduce this process as in Figure 9, where a bubble initially rises within a liquid bath covered by a surfactant layer. Upon reaching the surface, the bubble ascends through the liquid-air interface, resulting in the surfactant flowing atop the bubble. As the bubble settles on the liquid bath, its vertical oscillation influences the surrounding surfactant layer, leading to intricate color patterns. When the thickness of the bubble decreases below a threshold (2.0×10^{-8} m), we intentionally rupture the bubble. The Lagrangian particles on the bubble are then converted into "splash" particles which are simulated with the volumetric IISPH method [42]. The bubble bursting leaves a black hole in the surfactant layer on the liquid surface, which is rapidly filled due to the Marangoni effect. As the splash particles fall back into the liquid bath, they undergo a reverse conversion back into Lagrangian particles, retaining their original properties, which create dotted patterns on the surface.

Double bubbles with interfacial flow: We simulate binary collisions with bubbles of different surfactant concentrations to validate the multi-region interfacial flow simulation in Figure 7. The Lagrangian particles are initialized with attributes disturbed by Perlin noise. After two bubbles collide, the interfacial surfactant, driven by the concentration gradient, flows from the lower bubble to the upper one, across the non-manifold joint, fostering intricate patterns on the bubbles.

Bubbles on tank with interfacial flow: We further validate the multiphase simulation with two submerged bubbles and two additional bubbles released over a liquid tank. Both the bubbles and the tank carry surfactants of varying concentrations on their interfaces. As shown in Figure 15, gravitational forces cause the ascent and descent of the bubbles toward the liquid surface. This process enables the interconnection of their interfaces, thereby facilitating the multi-region interfacial flow of the surfactant. The concentration variation then drives surfactant flow across the interfaces, diffusing onto interfaces with low concentrations. Our method adeptly addresses both the volumetric and interfacial flows within this complex multiphase and multiregional system.

7 Discussion and Limitations

We propose a hybrid particle-grid approach that effectively captures a variety of complicated bubbles and foams phenomena. The key aspect of our approach is a novel particle representation that integrates the computational characteristics of moving least-squares particles and particle level-set methods. For each individual bubble, we adopt a carefully designed particle system to capture its interface evolution and handle the topological changes in foaming flow. These particles play two essential roles in our approach. From a geometric perspective, they serve as interface trackers, allowing us to accurately capture the structure evolution of the deformed surfaces and the topology of multiphase fluid domains. Meanwhile, the particle systems across all bubbles collectively establish an implicit surface representation on an unsigned level set, augmenting volumetric multiphase flow simulation. From a dynamics perspective, these particles provide robust discrete differential stencils on thin films to allow solving interfacial flow details. By integrating our particle representation with the volumetric multiphase flow framework, our method empowers the simulation of diverse and intricate bubble and foam phenomena, capturing the dynamic and complex structural evolution within volumetric fluids, as well as intricate interfacial flow occurring on interfaces.

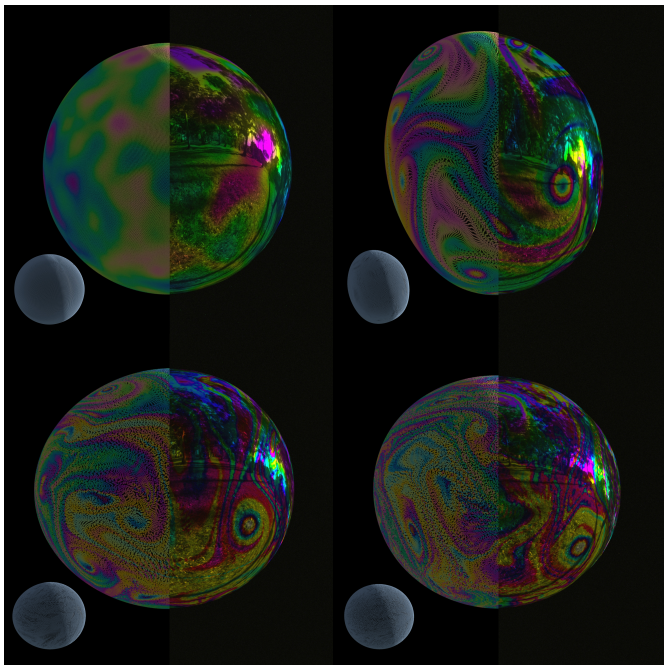


Fig. 16. A bubble showcasing surface-tension-driven oscillation and swirling interfacial flow. The photo-realistic rendering and colored Lagrangian particles $\mathcal{L}$ are visualized, with MLSLS particles $\mathcal{E}$ depicted at the bottom left corner.

Our method distinguishes itself from the existing literature on hybrid particle-grid methods in terms of the role particles play in the pipeline. Unlike previous methods, e.g., the particle level set method [43] or PIC/FLIP methods [44], where particles primarily serve as auxiliary structures to serve for interface and mass tracking, our MLSLS particles play their multifaceted, prominent roles in our framework. First, the dynamics of these particles drive the evolution of the bubble and foam interface, where the level set plays an auxiliary role in extending their values onto the background grid. The coupling between particles and the grid is embodied in the propagation-correction process (i.e., we use the grid information to correct particles, not vice versa), which is significantly different from the traditional hybrid schemes. Moreover, our MLSLS particles allow accurate simulation of surfactant flow details on deforming interfaces, which can be comprehended as a natural extension of the previous works on particle interface methods (e.g., SPH [45], MLS [16] and MELP [4] particles) to tackle nonmanifold and multiphase challenges. Overall, our method stands out from existing particle-tracking methods by enabling co-simulation of volumetric multiphase flow and interfacial flow, which allows us to faithfully reproduce a host of previously challenging bubble and foam phenomena that simultaneously exhibits complex volumetric and interfacial flow features.

Limitations: The following are the limitations of our approach. The first limitation is the instability caused by the surface tension. Our multiphase solver integrates the surface tension as an explicit pressure jump, which could lead to instability when dealing with fluids with a significant density difference and large surface tension. The second limitation lies in the limited ability to express diverse

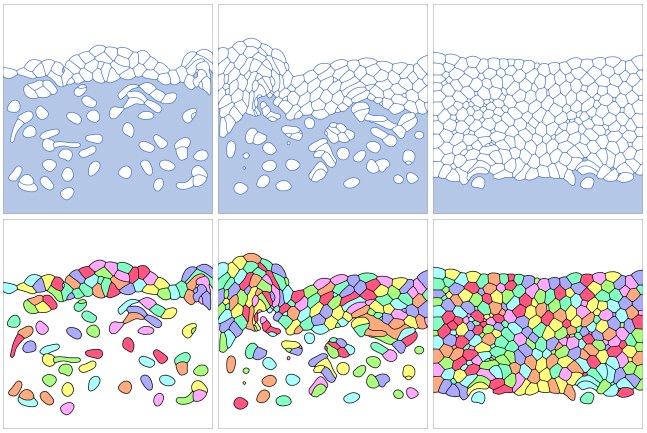


Fig. 17. A large number of submerged bubbles ascending to the water surface and stacking together. *Top:* Results colored according to the fluid phase. *Bottom:* Results colored according to region indices.

geometries, especially dimensional structures, within our discretization. Currently, our discretization is designed to handle volumetric regions with closed manifold surfaces. As a consequence, we cannot represent individual thin films or other open codimension-1 structures as separate entities within our simulations.

Future Work: Based on our current work, we propose two potential avenues for future exploration. First, we seek to investigate the coupling between the volumetric flow with the interfacial flow to reproduce more complicated codimensional scenarios, e.g., a thin liquid flows along the bubble gap and subsequently fills a liquid tank. These scenarios involve the interplay of codimensional structures not only at the interface itself but also at the non-manifold joint across codimensions. To tackle these challenges, a carefully designed strategy is required to implement the codimensional transitions and achieve physically-accurate coupling. Second, we aim to explore further the Lagrangian representation to facilitate the simulation of foaming flow that involves bubbles with significant differences in scale (e.g., with bubble sizes crossing several orders of magnitude). In this setting, Lagrangian particles are expected to not only evolve the interface's geometry but also track the subcell bubble volumes (e.g., see previous works in this direction [8], [36]). In particular, we plan to explore the adaptivity and flexibility of Lagrangian particles to track bubbles ranging from regular-sized to those across drastically different size scales. Moving forward, we intend to expand our simulations further to cover the scenario with a significant number of bubbles with varying sizes, aiming to reproduce more comprehensive and realistic phenomena, e.g., the foaming flow in turbulent breaking waves.

ACKNOWLEDGMENTS

We acknowledge the funding support from NSFC 62272305 and NSF 1919647, 2106733, 2144806, 2153560. We credit the Houdini Education licenses for the video rendering.

REFERENCES

[1] F. Da, C. Batty, C. Wojtan, and E. Grinspun, "Double bubbles sans toil and trouble: Discrete circulation-preserving vortex sheets for

Scene	Grid Resolution	Δx(m)	$\#\mathcal{L}$	$\#\mathcal{E}$	Time/Timestep(s)	Device Used
Four bubbles 3D	128^3	$1/128$	–	$388k$	11.0	B
Merging bubbles 3D	128^3	$0.1/128$	–	$2775k$	45.6	B
Droplet and bubbles 3D	128^3	$0.1/128$	–	$278k$	30.5	B
Jet on bubbles 3D	128^3	$0.1/128$	–	$3204k$	44.6	B
Rising bubbles 3D	256^3	$0.4/128$	–	$5222k$	228.2	A
Bubble with interfacial flow 3D	128^3	$1/128$	$200k$	$150k/80k$	21.2	B
The life-cycle of a bubble 3D	128^3	$0.4/128$	$200k(15k$ burst)	$289k/79k$	32.3	B
Double bubble with interfacial flow 3D	128^3	$1/128$	$1000k$	$160k/160k$	118.8	B
Bubbles on tank with interfacial flow 3D	128^3	$0.4/128$	$600k$	$240k/240k$	103.4	B

TABLE 3
Simulation configuration and timings. In the column "$\#\mathcal{E}$", the first number denotes the number of MLSLS particles $\mathcal{E}$ across all regions, while the second number represents the number of MLSLS particles $\mathcal{E}$ utilized for solving the interfacial flow. The data of "Merging bubbles 3D" and "Rising bubbles 3D" were recorded when there was the maximum number of regions. Specifically, "Merging bubbles 3D" had 128 regions, while "Rising bubbles 3D" had 767 regions. Device A is the computer with Intel(R) Core(TM) i9-9980X and Nvidia GeForce RTX 2080Ti, and Device B is the computer with AMD Ryzen 7 1700X and Nvidia GeForce GTX 1060Ti.

soap films and foams," *ACM Trans. Graph.*, vol. 34, no. 4, p. 149, 2015.

[2] S. Ishida, P. Synak, F. Narita, T. Hachisuka, and C. Wojtan, "A model for soap film dynamics with evolving thickness," *ACM Trans. Graph.*, vol. 39, no. 4, aug 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392405

[3] M. Wang, Y. Deng, X. Kong, A. H. Prasad, S. Xiong, and B. Zhu, "Thin-film smoothed particle hydrodynamics fluid," *ACM Trans. Graph.*, vol. 40, no. 4, jul 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459864

[4] Y. Deng, M. Wang, X. Kong, S. Xiong, Z. Xian, and B. Zhu, "A moving eulerian-lagrangian particle method for thin film and foam simulation," *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530174

[5] P. Karnakov, S. Litvinov, and P. Koumoutsakos, "Computing foaming flows across scales: From breaking waves to microfluidics," *Science Advances*, vol. 8, no. 5, p. eabm0590, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/sciadv.abm0590

[6] W. Zheng, J.-H. Yong, and J.-C. Paul, "Simulation of bubbles," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '06. Goslar, DEU: Eurographics Association, 2006, p. 325–333.

[7] F. Losasso, T. Shinar, A. Selle, and R. Fedkiw, "Multiple interacting liquids," *ACM Trans. Graph.*, vol. 25, no. 3, p. 812–819, jul 2006. [Online]. Available: https://doi.org/10.1145/1141911.1141960

[8] S. Patkar, M. Aanjaneya, D. Karpman, and R. Fedkiw, "A hybrid lagrangian-eulerian formulation for bubble generation and dynamics," in *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 105–114. [Online]. Available: https://doi.org/10.1145/2485895.2485912

[9] A. Stomakhin, J. Wretborn, K. Blom, and G. Daviet, "Underwater bubbles and coupling," in *ACM SIGGRAPH 2020 Talks*, ser. SIGGRAPH '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3388767.3407390

[10] J. Wretborn, S. Flynn, and A. Stomakhin, "Guided bubbles and wet foam for realistic whitewater simulation," *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530059

[11] B. Kim, "Multi-phase fluid simulations using regional level sets," in *ACM SIGGRAPH Asia 2010 Papers*, ser. SIGGRAPH ASIA '10. New York, NY, USA: Association for Computing Machinery, 2010. [Online]. Available: https://doi.org/10.1145/1866158.1866197

[12] R. I. Saye and J. A. Sethian, "Multiscale modeling of membrane rearrangement, drainage, and rupture in evolving foams," *Science*, vol. 340, no. 6133, pp. 720–724, 2013.

[13] R. I. Saye and J. A. Sethian, "Multiscale modelling of evolving foams," *J. Comput. Phys.*, vol. 315, pp. 273–301, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999116300158

[14] S. Ishida, M. Yamamoto, R. Ando, and T. Hachisuka, "A hyperbolic geometric flow for evolving films and foams," *ACM Trans. Graph.*, vol. 36, no. 6, p. 199, 2017.

[15] B. Zhu, E. Quigley, M. Cong, J. Solomon, and R. Fedkiw, "Codimensional surface tension flow on simplicial complexes," *ACM Trans. Graph.*, vol. 33, no. 4, p. 111, 2014.

[16] H. Wang, Y. Jin, A. Luo, X. Yang, and B. Zhu, "Codimensional surface tension flow using moving-least-squares particles," *ACM Trans. Graph.*, vol. 39, no. 4, aug 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392487

[17] J. Xing, L. Ruan, B. Wang, B. Zhu, and B. Chen, "Position-based surface tension flow," *ACM Trans. Graph.*, vol. 41, no. 6, nov 2022. [Online]. Available: https://doi.org/10.1145/3550454.3555476

[18] L. Wu, B. Wu, Y. Yang, and H. Wang, "A safe and fast repulsion method for gpu-based cloth self collisions," *ACM Trans. Graph.*, vol. 40, no. 1, dec 2020. [Online]. Available: https://doi.org/10.1145/3430025

[19] M. Li, D. M. Kaufman, and C. Jiang, "Codimensional incremental potential contact," *ACM Trans. Graph.*, vol. 40, no. 4, jul 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459767

[20] D. J. Hill and R. D. Henderson, "Efficient fluid simulation on the surface of a sphere," *ACM Trans. Graph.*, vol. 35, no. 2, apr 2016. [Online]. Available: https://doi.org/10.1145/2879177

[21] W. Huang, J. Iseringhausen, T. Kneiphof, Z. Qu, C. Jiang, and M. B. Hullin, "Chemomechanical simulation of soap film flow on spherical bubbles," *ACM Trans. Graph.*, vol. 39, no. 4, aug 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392094

[22] F. Da, D. Hahn, C. Batty, C. Wojtan, and E. Grinspun, "Surface-only liquids," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 78, 2016.

[23] L. Huang and D. L. Michels, "Surface-only ferrofluids," *ACM Trans. Graph.*, vol. 39, no. 6, nov 2020. [Online]. Available: https://doi.org/10.1145/3414685.3417799

[24] J.-M. Hong and C.-H. Kim, "Animation of bubbles in liquid," *Computer Graphics Forum*, vol. 22, no. 3, pp. 253–262, 2003. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.00672

[25] J.-M. Hong and C.-H. Kim, "Discontinuous fluids," *ACM Trans. Graph.*, vol. 24, no. 3, p. 915–920, jul 2005. [Online]. Available: https://doi.org/10.1145/1073204.1073283

[26] L. Boyd and R. Bridson, "Multiflip for energetic two-phase fluid simulation," *ACM Trans. Graph.*, vol. 31, no. 2, apr 2012. [Online]. Available: https://doi.org/10.1145/2159516.2159522

[27] B. Kim, Y. Liu, I. Llamas, X. Jiao, and J. Rossignac, "Simulation of bubbles in foam with the volume control method," *ACM Trans. Graph.*, vol. 26, no. 3, p. 98–es, jul 2007. [Online]. Available: https://doi.org/10.1145/1276377.1276500

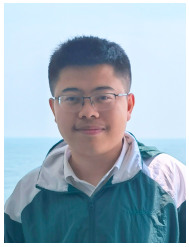
[28] N. Balcázar, O. Lehmkuhl, J. Rigola, and A. Oliva, "A multiple marker level-set method for simulation of deformable fluid particles," *International Journal of Multiphase Flow*, vol. 74, pp. 125–142, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301932215001019

[29] R. I. Saye and J. A. Sethian, "The voronoi implicit interface method for computing multiphase physics," *Proceedings of the National Academy of Sciences*, vol. 108, no. 49, pp. 19 498–19 503, 2011. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1111557108

[30] R. Ando, N. Thuerey, and C. Wojtan, "A stream function solver for liquid simulations," *ACM Trans. Graph.*, vol. 34, no. 4, jul 2015. [Online]. Available: https://doi.org/10.1145/2766935

[31] R. Goldade, M. Aanjaneya, and C. Batty, "Constraint bubbles and affine regions: Reduced fluid models for efficient immersed bubbles and flexible spatial coarsening," *ACM Trans. Graph.*, vol. 39, no. 4, aug 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392455
[32] Y. Guo, X. Liu, and X. Xu, "A unified detail-preserving liquid simulation by two-phase lattice boltzmann modeling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 5, pp. 1479–1491, 2017.
[33] W. Li, Y. Ma, X. Liu, and M. Desbrun, "Efficient kinetic simulation of two-phase flows," *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530132
[34] Y. Yue, B. Smith, C. Batty, C. Zheng, and E. Grinspun, "Continuum foam: A material point method for shear-dependent flows," *ACM Trans. Graph.*, vol. 34, no. 5, nov 2015. [Online]. Available: https://doi.org/10.1145/2751541
[35] D. Kim, O.-y. Song, and H.-S. Ko, "A practical simulation of dispersed bubble flow," *ACM Trans. Graph.*, vol. 29, no. 4, jul 2010. [Online]. Available: https://doi.org/10.1145/1778765.1778807
[36] O. Busaryev, T. K. Dey, H. Wang, and Z. Ren, "Animating bubble interactions in a liquid foam," *ACM Trans. Graph.*, vol. 31, no. 4, jul 2012. [Online]. Available: https://doi.org/10.1145/2185520.2185559
[37] S. Ianniello and A. Di Mascio, "A self-adaptive oriented particles level-set method for tracking interfaces," *Journal of Computational Physics*, vol. 229, no. 4, pp. 1353–1380, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999109005865
[38] M. Kness. (2008) Colorpy - a python package for handling physical descriptions of color and light spectra. [Online]. Available: http://markkness.net/colorpy/ColorPy.html
[39] H. Lhuissier and E. Villermaux, "Bursting bubble aerosols," *Journal of Fluid Mechanics*, vol. 696, p. 5–44, 2012.
[40] P. L. L. Walls, J. C. Bird, and L. Bourouiba, "Moving with Bubbles: A Review of the Interactions between Bubbles and the Microorganisms that Surround them," *Integrative and Comparative Biology*, vol. 54, no. 6, pp. 1014–1025, 08 2014. [Online]. Available: https://doi.org/10.1093/icb/icu100
[41] L. Bourouiba, "The fluid dynamics of disease transmission," *Annual Review of Fluid Mechanics*, vol. 53, no. 1, pp. 473–508, 2021. [Online]. Available: https://doi.org/10.1146/annurev-fluid-060220-113712
[42] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner, "Implicit incompressible sph," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 426–435, 2013.
[43] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, "A hybrid particle level set method for improved interface capturing," *Journal of Computational physics*, vol. 183, no. 1, pp. 83–116, 2002.
[44] Y. Zhu and R. Bridson, "Animating sand as a fluid," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 965–972, 2005.
[45] S. E. Hieber and P. Koumoutsakos, "A lagrangian particle level set method," *J. Comput. Phys.*, vol. 210, no. 1, p. 342–367, nov 2005. [Online]. Available: https://doi.org/10.1016/j.jcp.2005.04.013

Hui Wang Hui Wang received his B.S. degree and M.S. degree from Shanghai Jiao Tong University in 2018 and 2021. He is now a Ph.D. student in the School of Software at Shanghai Jiao Tong University, China since 2021. His research interests include physics-based animation and physics-based deep learning in computer graphics.



Zhi Wang Zhi Wang obtained his bachelor's degree from Shanghai Jiao Tong University in 2022. He is currently pursuing a master's degree at the Digital Art Laboratory, School of Software, Shanghai Jiao Tong University. His research interests include physics-based animation in computer graphics.

Shulin Hong Shulin Hong is a graduate student pursuing his master's degree at the Digital Art Laboratory in the School of Software Engineering at Shanghai Jiao Tong University. His research interests include computer graphics and the application of deep learning in graphics-related areas.

Xubo Yang Xubo Yang is currently a professor in the School of Software at Shanghai Jiao Tong University. He received his PhD degree in computer science from the State Key Lab of CAD & CG at Zhejiang University, in 1998. His research interests include computer graphics, virtual reality, and human computer interaction.

Bo Zhu Bo Zhu is an assistant professor in the School of Interactive Computing at Georgia Institute of Technology. He obtained his Ph.D. from Stanford and conducted his postdoctoral research at MIT CSAIL. His research interests include computer graphics, computational physics, computational fluid dynamics, and scientific machine learning. He mainly focuses on building computational approaches to simulate complex fluid systems.