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

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

APPENDIX A
MOVING LEAST-SQUARES METHOD

A.1 Approximating the MLS surface

Given the a point $\mathbf{x}$ and its local frame $\mathbf{R}$, we split the local coordinate $\boldsymbol{\xi}$ into the tangential part $\boldsymbol{\xi}^\top$ and the normal part $\xi^\perp$. For example, in 2D, the tangential and the normal part are $[\xi_1]$, $[\xi_2]$. In 3D, they are $[\xi_1, \xi_2]^T$ and $[\xi_3]$.

In the MLS method, the local surface is approximated as $h(\boldsymbol{\xi}^\top) = \boldsymbol{\beta} \cdot \mathbf{b}(\boldsymbol{\xi}^\top)$, where $\boldsymbol{\beta}$ is the coefficients approximated by the MLS, $\mathbf{b}(\boldsymbol{\xi}^\top)$ returns the 2nd polynomial basis vector. $\boldsymbol{\beta} = [\beta_1, \beta_2, \beta_3]^T$, $\mathbf{b}(\boldsymbol{\xi}^\top) = [1, \xi_1, \xi_1^2]^T$ in 2D and $\boldsymbol{\beta} = [\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6]^T$, $\mathbf{b}(\boldsymbol{\xi}^\top) = [1, \xi_1, \xi_2, \xi_1^2, \xi_1\xi_2, \xi_2^2]^T$ in 3D.

A.2 Projecting point on the MLS surface

Given a point $\mathbf{x}$ with its local coordinate $[(\boldsymbol{\xi}^\top)^T, \boldsymbol{\xi}^\perp]^T$, its projected point $[(\boldsymbol{\xi}_{proj}^\top)^T, h(\boldsymbol{\xi}_{proj}^\top)]^T$ on the MLS surface can be found by solving the minimization problem:

$$\min |\boldsymbol{\xi}^\top - \boldsymbol{\xi}_{proj}^\top|^2 + (\boldsymbol{\xi}^\perp - h(\boldsymbol{\xi}_{proj}^\top))^2 \tag{1}$$

We solve this problem by the Newton method, where the gradient and the second derivative of the target function are obtained based on the polynomial coefficients $\boldsymbol{\beta}$.

The gradients are $\nabla h(\xi^\top) = (\beta_2 + 2\beta_3\xi_1)$ in 2D and $\nabla h(\xi^\top) = (\beta_2 + 2\beta_4\xi_1 + \beta_5\xi_2, \beta_3 + 2\beta_6\xi_2 + \beta_5\xi_1)^T$ in 3D. And the second-order derivatives are $\nabla \circ \nabla h(\xi^\top) = 2\beta_3$ in 2D and $\nabla \circ \nabla h(\xi^\top) = (2\beta_4\xi_1 + \beta_5\xi_2, \beta_3 + 2\beta_6\xi_2 + \beta_5\xi_1)^T$ in 3D where $\circ$ is the Hadamard product, which means $(A \circ B)_{ij} = (A)_{ij}(B)_{ij}$. The normal of the projected point $\boldsymbol{n}_{proj}$ is spawned orthogonal to the tangential vectors $\nabla h(\xi^\top)$.

APPENDIX B
REGIONAL LEVEL SET RECONSTRUCTION

To reconstruct the regional level set ϕ_i based on the MLSLS particles $\mathcal{E}_i$, we carry out the following steps.

Closest Particle Search: Given a grid cell $\boldsymbol{x}_c$, we begin by searching for its closest particle $p \in \mathcal{E}_i$ to $\boldsymbol{x}_c$ within a searching radius $r_{CP} = r_N + 0.5\Delta x$ that is slightly wider than the narrow band. If no neighbor particle is found within the range, we consider the cell to be located far from the interface Ω_i and skip further processing for that cell. Then, we construct the local frame $\boldsymbol{R}_p$ based on the normal vector $\boldsymbol{n}_p$ on the particle $\boldsymbol{x}_p$.

Local MLS Surface Fitting: Next, we approximate the local MLS surface at $\boldsymbol{x}_p$ according to [16]. We collect the neighbor particles $\mathcal{E}_{NB}$ that lie within a specified support range $r_{MLS} = 2\Delta x$ centered at $\boldsymbol{x}_p$. Note that to exclude the particles that do not belong to the same surface, we only consider particles q whose normals $\boldsymbol{n}_p$ satisfy $\boldsymbol{n}_q \cdot \boldsymbol{n}_p > 0$. Given the particle $\boldsymbol{x}_p$ and its local frame $\boldsymbol{R}_p$, we use $\boldsymbol{\xi}^\top$ to represent the local tangential coordinate and $\xi^\perp$ to represent the normal one. The local surface is approximated as $h(\boldsymbol{\xi}^\top) = \boldsymbol{\beta} \cdot \boldsymbol{b}(\boldsymbol{\xi}^\top)$, where $\boldsymbol{\beta}$ returns the coefficients fitted by MLS and $\boldsymbol{b}(\boldsymbol{\xi}^\top)$ is the polynomial basis vector (full expression provided in Appendix A.1). We obtain $h(\boldsymbol{\xi}^\top)$ by minimizing the MLS objective function on neighbor particles $\mathcal{E}_{NB}$.

MLS Surface Projection: To obtain the closest point on the MLS surface, we project the cell center $\boldsymbol{x}_c$ onto the MLS surface as $\boldsymbol{x}_{proj}$ ($\boldsymbol{\xi}_{proj}$ as the corresponding local coordinate). Given a point $\boldsymbol{x}_c$ with its local coordinate $[(\boldsymbol{\xi}_c^\top)^T, \boldsymbol{\xi}_c^\perp]^T$, its projected point on the MLS surface $[(\boldsymbol{\xi}_{proj}^\top)^T, h(\boldsymbol{\xi}_{proj}^\top)]^T$ is found by solving the minimization problem

$$\min |\boldsymbol{\xi}_c^\top - \boldsymbol{\xi}_{proj}^\top|^2 + |\xi_c^\perp - h(\boldsymbol{\xi}_{proj}^\top)|^2. \tag{2}$$

We solve this problem using the Newton method, where the first- and second-order derivatives of the target function are obtained based on $h(\boldsymbol{\xi}^\top)$ (see Appendix A.2 for the detailed derivation).

Projected Point Adjustment: In certain cases, especially when dealing with sharp features, the projected point may lie far away from the actual surface (as shown in Figure 5 (right)). To address this, we confine the point in a vicinity defined by the neighbor particles $\mathcal{E}_{NB}$ on the local tangent space. Following each iteration of the Newton method, we project all the neighboring particles $\boldsymbol{\xi}_q^\top$ onto the vector

• *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*

$\boldsymbol{\xi}_{proj}^{\top} - \mathbf{0}$ in the local tangent space. The projected point $\boldsymbol{\xi}_{proj}$ is valid when it satisfies

$$\boldsymbol{\xi}_{proj}^{\top} < \max_{q \in \mathcal{E}_{NB}} \boldsymbol{\xi}_{q}^{\top} \cdot \frac{\boldsymbol{\xi}_{proj}^{\top}}{|\boldsymbol{\xi}_{proj}^{\top}|}. \tag{3}$$

Otherwise, we simply truncate the tangential local coordinate $\boldsymbol{\xi}_{proj}^{\top}$ at the largest projected length and update the local coordinate in the normal direction as $\boldsymbol{\xi}_{proj}^{\perp} = h(\boldsymbol{\xi}_{proj}^{\top})$.

Level Set Value Update: According to the definition of the level set, the distance from the cell to the surface $|\boldsymbol{x}_c - \boldsymbol{x}_{proj}|$ corresponds to the absolute value of ϕ_i at cell c. If this distance is smaller than $|\boldsymbol{x}_c - \boldsymbol{x}_p|$, the projected point is accepted, and we update the level set value as $\phi_i(\boldsymbol{x}_c) = \text{Sign}(\boldsymbol{n}_p \cdot (\boldsymbol{x}_c - \boldsymbol{x}_{proj}))|\boldsymbol{x}_c - \boldsymbol{x}_{proj}|$. Otherwise, we simply use the closest particle $\boldsymbol{x}_p$ to update the level set value.

APPENDIX C
INTERFACIAL FLOW

C.1 Lagrangian Particles

Definition: Following [4], we introduce an extra set of Lagrangian particles $\mathcal{L}$. These Lagrangian particles $\mathcal{L}$ are positioned on the region interfaces and advect along the interfaces, carrying various physical quantities including mass m_L, volume V_L, surfactant c_L, tangential velocity $\boldsymbol{u}_L^{\top}$, thickness η_L.

Advection: Since the interfacial flow is decoupled with the volumetric multiphase flow, we independently advect Lagrangian particles $\mathcal{L}$ with the velocity of both flows. We begin by advecting them with the normal component of the grid velocity $(\boldsymbol{u}_G(\boldsymbol{x}_L) \cdot \boldsymbol{n}_L)\boldsymbol{n}_L$. Their normals $\boldsymbol{n}_L$ are dynamically computed as the MLS surface normal, approximated on MLSLS particles $\mathcal{E}_i$ from their closest neighboring region Ω_i. Subsequently, we advect them using their thin film tangential velocity $\boldsymbol{u}_L^{\top}$. We then determine their closest neighboring region Ω_j based on their advected position and project them onto the MLS surface fitted on MLSLS particles $\mathcal{E}_j$. This process facilitates Lagrangian particles to adhere to interfaces while also allowing them to move freely across interfaces of different regions.

C.2 Interfacial Dynamics Computation

We solve Equation (3) following MELP [4], where Lagrangian particles $\mathcal{L}$ and MLSLS particles $\mathcal{E}$ collaborate to simulate the interfacial flow. The procedure is illustrated below:

$\mathcal{E}$ **Redistribution**: To ensure a uniform sampling of $\mathcal{E}_i$, we perform a redistribution step on them. We set up a pseudo-pressure equation

$$\overline{\Delta t^2}\delta^* \nabla_s^2 C = \bar{\delta} - \delta^* \tag{4}$$

where ∇_s^2 is the surface Laplacian operator introduced in [4], C is the pseudo pressure, δ^* and $\bar{\delta}$ are the current number density and the average number density on MLSLS particles $\mathcal{E}_i$, $\overline{\Delta t}$ is a temporal step size. We solve this equation using the Implicit Incompressible SPH (IISPH) [42] and update the position of MLSLS particles $\mathcal{E}_i$ with $\overline{\Delta t^2}\beta\nabla_s C$ where β is a step size.

Note that this step serves a distinct purpose compared to the particle reseeding step (in Section 4.2.3). The particle reseeding step ensures a sufficient and appropriate sampling of MLSLS particles to represent a smooth surface, but does not guarantee uniform distribution. In contrast, this step is specifically implemented to adjust MLSLS particles towards a uniform distribution, therefore providing an accurate computational scheme of interfacial SPH necessitated by interfacial flow dynamics.

$\mathcal{L}2\mathcal{E}$ **Transfer**: We search for the two closest neighboring regions Ω_i, Ω_j for each Lagrangian particle $\mathcal{L}$. Then, we transfer the mass m, surfactant c, volume V and tangential momentum $\boldsymbol{p}^{\top}$ from $\mathcal{L}$ to $\mathcal{E}_i$ and $\mathcal{E}_j$ separately. The tangential momentum of $\mathcal{L}$ is calculated as $\boldsymbol{p}_L^{\top} = \boldsymbol{u}_L^{\top} m_L$. Additionally, we construct affine momentum $\hat{\boldsymbol{p}}^{\top}$ for APIC and reconstruct the velocity of MLSLS particles as $\boldsymbol{u}_E^{\top} = (\boldsymbol{p}_E^{\top} + \hat{\boldsymbol{p}}_E^{\top})/m_E$.

Geometry Computation: After the transfer, we compute the control area a_E, mean curvature H_E, and metric tensor g_E on $\mathcal{E}_i$ using the MLS method. With a_E, the thickness of $\mathcal{E}_i$ is updated as $\eta_E = V_E/a_E$ and the surfactant concentration is updated as $\Gamma_E = c_E/a_E$. The mean curvature H_E and metric tensor g_E are utilized to construct the codimension-1 SPH-based differential operators following [4], including the surface gradient ∇_s, surface divergence $\nabla_s \cdot$, and surface Laplacian ∇_s^2.

Interfacial Flow Equation Solving: Based on [4], we solve the following equation of surfactant concentration Γ_E^* derived from Equation (3) on MLSLS particles $\mathcal{E}$ using the IISPH method.

$$\begin{aligned}&\left(-\frac{1}{\Delta t \Gamma_E}\right)\Gamma_E^* + \left(\Delta t \frac{\bar{R}T}{\rho}\nabla\frac{1}{\eta_E}\right)\cdot\nabla_s\Gamma_E^* + \left(\Delta t\frac{\bar{R}T}{\rho}\frac{1}{\eta_E}\right)\nabla_s^2\Gamma_E^* \\ &= \nabla\cdot\boldsymbol{u}_E^{\top} - \frac{1}{\Delta t} + \Delta t\left(\nabla\frac{1}{\rho}\cdot\boldsymbol{g}^{\top}\right)\end{aligned} \tag{5}$$

With Γ_E^* in hand, we update the tangential velocity as

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

where η_E^* is the thickness updated using Equation (3).

$\mathcal{E}2\mathcal{L}$ **Transfer**: We transfer the updated tangential velocity $\boldsymbol{u}_E^{\top *}$ and its divergence $\nabla_s \cdot \boldsymbol{u}_E^{\top *}$ from MLSLS particles $\mathcal{E}_i$ and $\mathcal{E}_j$ back onto Lagrangian particles $\mathcal{L}$ as $\boldsymbol{u}_{Li}^*$, $\boldsymbol{u}_{Lj}^*$.

If the two transferred velocity are similar $|\boldsymbol{u}_{Li}^* - \boldsymbol{u}_{Lj}^*|/\min(|\boldsymbol{u}_{Li}^*|, |\boldsymbol{u}_{Lj}^*|) < \epsilon_{\boldsymbol{u}}$, the Lagrangian velocity is determined as their average $\boldsymbol{u}_L^T = 0.5(\boldsymbol{u}_{Li}^* + \boldsymbol{u}_{Lj}^*)$. Otherwise, the Lagrangian particle $\mathcal{L}$ might be located at a non-manifold joint, where the interfacial velocities of two adjacent interfaces are diverse. In this case, we determine the Lagrangian velocity based on a probabilistic scheme, to accept one of $\boldsymbol{u}_{Li}^*$, $\boldsymbol{u}_{Lj}^*$.

The scheme is derived from the intuition that the flux at the non-manifold joint is split onto two interfaces. By assuming the same local thickness at $\mathcal{L}$, the flux on each interface is proportional to the interfacial velocity magnitude. Therefore, we determine the velocity for $\mathcal{L}$ with the probability:

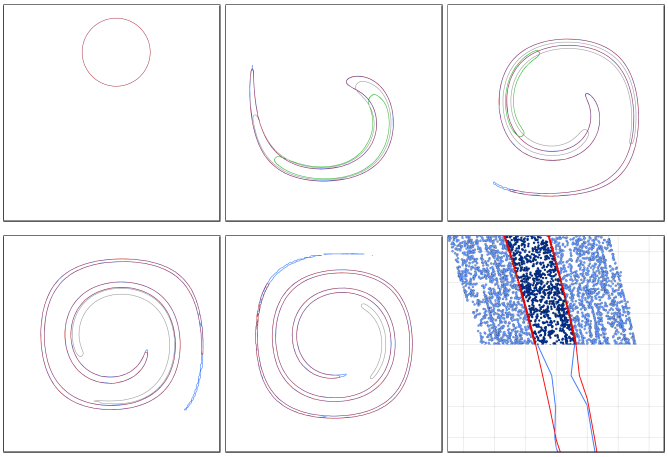


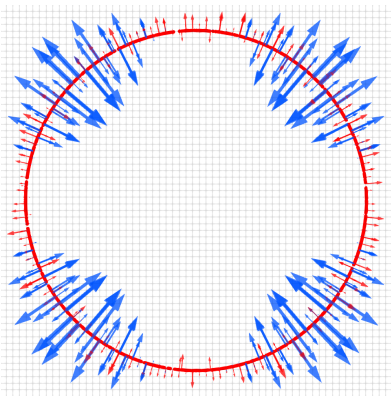
Fig. 1. The deformation of a 2D circle within a vortex flow field employing different multi-region tracking methods. We depict the interfaces tracked by multi-region tracking methods, including the standard level set method(grey), Voronoi implicit interface method [29](green), multiple particle level set method [7](blue), and our proposed method (red). In the lower right corner, a zoom-in view at the 150th frame illustrates the interfaces (solid lines) and underlying particles (dots) utilized in the multiple particle level set method(blue) and our proposed method (red).

$$\begin{cases} P(\boldsymbol{u}_L^T := \boldsymbol{u}_{Li}^*) = \dfrac{\boldsymbol{u}_{Li}^*}{|\boldsymbol{u}_{Li}^* + \boldsymbol{u}_{Lj}^*|} \\ P(\boldsymbol{u}_L^T := \boldsymbol{u}_{Lj}^*) = \dfrac{\boldsymbol{u}_{Lj}^*}{|\boldsymbol{u}_{Li}^* + \boldsymbol{u}_{Lj}^*|} \end{cases}. \tag{7}$$

Finally, we update the thickness η_L for $\mathcal{L}$ with the average velocity divergence, following the third row in Equation (3).

APPENDIX D
SURFACE TRACKING VALIDATION

We conducted a comparative analysis to validate the accuracy of surface tracking. Our proposed method was compared to other multi-region tracking methods in the "vortex-in-a-box" problem [43], where a circle undergoes stretching under a non-constant vorticity velocity field. The results obtained from the standard level set method, Voronoi implicit interface method [29], multiple particle level set method [7], and our proposed method are presented in Figure 1. Note that for the latter three methods, both the interfaces of the inner region (the interior of the circle) and the outer region (the ambient surroundings) are tracked, with only the interface from the inner region being visualized. Our method, leveraging the MLSLS particles, exhibits better tracking accuracy when compared to purely grid-based methods such as the standard level set method and Voronoi implicit interface method [29]. In comparison to the multiple particle level set method [7], although our method sacrifices some sub-grid details(see the thin tails in Figure 1), it achieves enhanced smoothness while utilizing fewer underlying particles. This advantage arises from the particle placement strategy and the interface approximation model based on MLS polynomials, as opposed to the model based on randomly sampled individual particles in particle level set methods.



Method	avg(ϵ)	var(ϵ)
MLS Particle	0.0417	0.0047
Level Set	0.0877	0.0309

Fig. 2. The error analysis of curvature estimation. *Top:* Distribution of curvature errors ϵ estimated on interfaces using particle MLS method and level set method. Both MLSLS particles (red dots) on interfaces and grid edges (grey lines) are depicted. Arrows visualize signed errors, including those of the particle MLS method (red arrows) and the level set method (blue arrows). *Bottom:* The relative average and variance of the error ϵ using MLS particle-based approximation and level set-based approximation.

APPENDIX E
CURVATURE ESTIMATION VALIDATION

We conduct a comparative analysis between our MLS particle-based curvature estimation and the level set-based approach on a 2D circle. The MLS particles are randomly sampled and on the interface with the same density used in the simulation, while the level set values are initialized as the signed distance to the circle. In the level set-based approach, we approximate the curvature as $\nabla \cdot (\nabla\phi/|\nabla\phi|)$ using the central difference method. We then estimate the curvature on the interfaces using both methods and analyze the resulting error ϵ by comparing them with the analytical curvature of the circle. The error distribution, relative average, and variance are illustrated in Figure 2. The results indicate that in comparison to the level set-based method, our MLS approximation achieves superior accuracy overall. Additionally, it is observed that the level set-based method exhibits an error distribution sensitive to orientation, manifesting an increase in error when the interface is not aligned with the grid edge. In contrast, such sensitivity is not observed in the MLS-based method.

APPENDIX F
TIME BREAKDOWN

We showcase the detailed time breakdown of some characteristic experiments in Figure 3. In our current implementation, the "Volumetric Flow Solving" and "Velocity Advection" steps are executed in a highly parallel manner on the Eulerian grid using CUDA, whereas other steps are parallelized using OpenMP.

It's worth noting that the step of particle-grid interaction, including "Particle-To-Grid Propagation" and "Grid-To-Particle Correction", is notably influenced by the number of regions. For instance, in the "Rising Bubbles" scene with

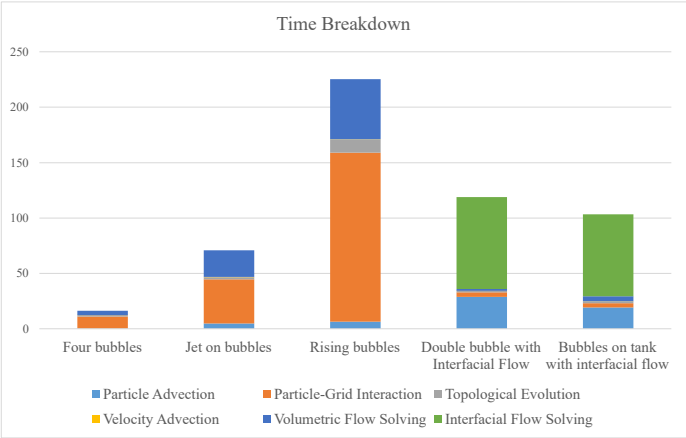


Fig. 3. The detailed time breakdown for characteristic scenes.

767 regions, approximately 67% of the total time is dedicated to the particle-grid interaction. Furthermore, when the interfacial flow simulation is involved, both the ”Particle Advection” and ”Interfacial Flow Solving” steps require additional time for geometric tracking and dynamic solving.