

# A coupled particle level set and Moment-of-fluid method for simulating multiphase flows with phase change \*

Estebe, Cody  
Dept. of math, FL State Univ.

Liu, Yang  
Dept. of Mech. Engr., FL State Univ.

Sussman, Mark  
Dept. of math, FL State Univ.

Moradikazerouni, Alireza  
Dept. of Mech. Engr., FL State Univ.

Shoele, Kourosh  
Dept. of Mech. Engr., FL State Univ.

Ohta, Mitsuhiro  
Dept. of Mech. Sci, Tokushima University

June 12, 2024

**ABSTRACT:** An improved algorithm for computing multiphase flows is presented in which the multimaterial Moment-Of-Fluid (MOF) algorithm for multiphase flows, described by Li et al (2015), is enhanced with meshless particles. The new method is less “dissipative” than the Continuous Moment of Fluid method (“CMOF,” Ye et al (2023)) and inherently removes the “checkerboard instability” that is present when using the MOF method on surface tension driven multiphase (multimaterial) flows (Li et al (2015)).

Multiphase flow examples are shown in 2D, 3D axisymmetric “RZ,” and 3D coordinate systems. Examples include two material and three material multiphase flows: bubble formation, the impingement of a liquid jet on a gas bubble in a cryogenic fuel tank, freezing, melting, evaporation, and liquid lens dynamics.

---

\*this material is based upon work supported by National Aeronautics and Space Administration under grant number 80NSSC20K0352

# 5 1 Introduction

In Ye et al[?], an improved “Continuous Moment-of-Fluid” (CMOF) algorithm for computing solutions to multiphase (multimaterial) flows was presented for overcoming the interface curvature “checkerboard” problem present in the “Moment of Fluid” (MOF) [?] method. Also in Ye et al[?], a novel decision tree data science approach was presented for accelerating the CMOF slope reconstruction algorithm.

In this article, we present a “Particle Level Set Moment-of-Fluid” (PLSMOF) method for overcoming the interface checkerboard problem. This new approach we claim is more accurate and faster than the CMOF approach and does not require the expensive initial training step. We also claim that our new approach is faster and more accurate than the CLSVOF method[?] with the added capability for simulating general multimaterial ( $\geq 2$  materials) multiphase flows.

We demonstrate the benefits of our new algorithm, beyond that of the current state-of-the-art, on these benchmark multiphase flow problems: bubble formation[24] (see Section 5.2), freezing[26] (see Section 5.4), liquid lens[37] (see Section 5.3), and bubble dynamics in a Cryogenic fuel tank[6] (see Section 5.5).

Figure 1 (LEFT figure) illustrates the Moment-of-Fluid[15, 1, 2] reconstruction of a “saw tooth” function that has a wave length of  $\Delta x$ . The MOF reconstruction is very “noisy” in the sense that the highest frequency Fourier coefficients of the discrete Fourier transform of the reconstruction do not decay with repeated reconstructions. In other-words the amplitudes of the high frequency Fourier modes may not decay when exposed to the repeated process of (1) MOF advection, and (2) MOF reconstruction. The presence of  $O(\Delta x)$  wave length noise is a problem for applications involving surface tension driven flows. Standard Volume-of-Fluid techniques for extracting the curvature from volume fractions[56, 13] will extract a zero curvature from a “saw tooth” interface since the associated volume fraction field *and* centroid field varies in the  $y$  direction only. In other words, the underlying computational fluid dynamics surface tension force algorithm will never “see” the jagged interface and therefore will have no way for removing the noise. If there is viscosity, the presence of the noise is unphysical and the noise can lead to collateral damage (see Figure 19 and Table 4) to the overall flow field; the noise can result in loss of accuracy.

In order to overcome the “MOF checkerboard instability” issue, we have developed the “Continuous Moment of Fluid” (CMOF) interface reconstruction algorithm[63]. Referring to Figure 1, the repeated process of (1) CMOF advection, followed by (2) CMOF reconstruction, will quickly eliminate the noise. The difference between CMOF and MOF is that in the CMOF algorithm, the reference centroid is the material centroid relative to the encompassing  $3 \times 3 \times 3$  stencil of grids cells, rather than just the center cell. See Figure 2.

We remark that alternative approaches to using MOF for multiphase (multimaterial) flows, e.g the level set method[54, 50, 55], phase field method[28], front tracking method [65], or standard “PLIC” VOF methods[3, 48, 49], do not have the “MOF checkerboard instability” problem. These alternative approaches have other difficulties. The level set or front tracking multimaterial

50 approaches[54, 50, 55, 65] are not volume preserving methods. The phase field method[28] smears the interface over 3 grid cells or more. The standard, second order, “PLIC” VOF methods for  $M$  material multimaterial flows[3, 48, 49] minimize a cost function which has  $27(M - 1)$  degrees of freedom; i.e. the control variable space is  $27(M - 1)$  dimensions.

55 Our improved CMOF method, on the other-hand, admits a sharp reconstructed interface, is volume preserving, and the control variable space has only  $3(M - 1)$  dimensions. The CMOF reconstruction algorithm has been improved in this article by applying “Decision Tree” machine learning techniques in order to rapidly determine the optimal “CMOF” slope. We choose the Decision  
60 Tree algorithm[10] which is a “lossless” method if one chooses to store all of the training data samples in the tree structure. In Table 1, we summarize the existing state-of-the-art for numerical methods for multiphase (multimaterial) flows with surface tension.

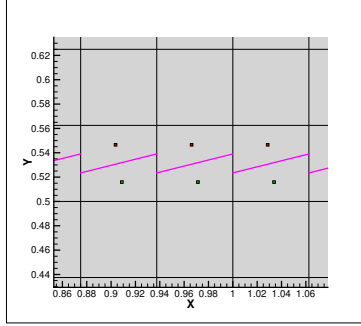


Figure 1: LEFT: Standard curvature discretization methods [56, 13, 45] will predict zero curvature for this saw-tooth interface since the volume fractions *and* centroids vary only in the  $y$  direction. Repeated Moment-of-Fluid (MOF) (i) reconstruction followed by (ii) the replacement of the reference centroid with the MOF actual centroid will not alter the initial interface (ignoring round-off error). RIGHT: the gaps in the initial saw-tooth interface are immediately removed with PLSMOF reconstruction since the level set function is derived as the exact signed distance function to the initial MOF reconstructed interface, and the particles are initialized from the level set function. Ensuing slopes are derived from a combination of the particles and the signed distance function (if particles available).

## 2 Background - deforming boundary problems in fluid mechanics

In the Introduction, we briefly demonstrate the benefits of our new method over the current state-of-the-art numerical methods for multi-phase flows in which surface tension forces are important. In this section, we give an overview of

Author(s)	Triple point reconstruction algorithm DOF <sup>1</sup>	Coupled with fluid	Volume preserving	Sharp interface	Curvature discretization
Smith et al. [54]	Level set	Yes	No	Yes	Level set
Ahn and Shakhov[1]	VOF-GRAD <sup>2</sup> VOF-LVIRA <sup>3</sup> MOF 27 or 3 $\times(M-1)$	No	Yes	Yes	N/A
Kim[31]	Phase field	Yes	Yes	No	Phase field
Dyadechko and Shashkov[16]	MOF $3(M-1)$	No	Yes	Yes	N/A
Caboussat et al. [11]	VOF-IP <sup>4</sup> $27(M-1)$	Yes	No	Yes	Convolution/Height function[18]
Schofield et al. [48, 49]	VOF-PD <sup>5</sup> $27(M-1)$	No	Yes	Yes	N/A
Sijoy and Chaturvedi[53]	VOF-PLIC <sup>6</sup> $27(M-1)$	Yes	Yes	Yes	N/A
Kucharik et al. [32]	VOF-PLIC VOF-PD MOF 27 or 3 $\times(M-1)$	Yes	Yes	Yes	N/A
Bonhomme et al. [8]	VOF $27(M-1)$	Yes	Yes	No	VOF
Starinshak et al. [55]	Interface level set	Yes	No	Yes	N/A
Vu et al. [65]	Front tracking	Yes	No	Yes	Front tracking
Pathak and Raessi[42]	VOF-PLIC $27(M-1)$	No	Yes	Yes	N/A
Shetabivash et al. [50]	Level Set	Yes	No	Yes	Finite Difference
Ancellin et al. [3]	VOF-PLIC $27(M-1)$	No	Yes	Yes	N/A
Huang et al. [28]	Phase field	Yes	Yes	No	Phase field
Present Article, Ye et al	CMOF <sup>7</sup> $3(M-1)$	Yes	Yes	Yes	VOF Height function

<sup>1</sup>: Degrees of Freedom for 3D reconstruction;  $M$  is the number of materials

<sup>2</sup>: Gradient based interface reconstruction

<sup>3</sup>: Least squares volume-of-fluid interface reconstruction algorithm

<sup>4</sup>: Interior-point method for the localization of the triple point

<sup>5</sup>: Power diagram

<sup>6</sup>: Piecewise linear interface construction

<sup>7</sup>: Continuous Moment of Fluid construction

Table 1: Recent numerical methods of flows with triple points in chronological order.

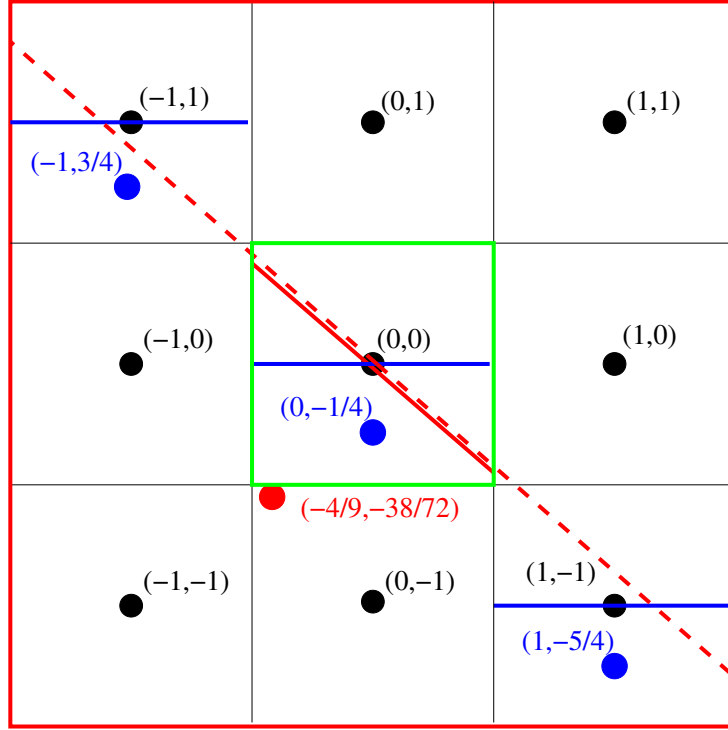


Figure 2: Illustration of the difference between Continuous Moment-of-Fluid (CMOF) and Moment-of-Fluid (MOF) reconstruction. The black circles represent cell centroids, blue circles represent “material 1” centroids, and the red circle represents the “CMOF” centroid associated with the center cell (in green) and calculated using the  $3 \times 3$  data surrounded by the CMOF cell (in red). The CMOF reconstructed interface (the red line in the center cell) is the line segment which minimizes the difference between the CMOF reference centroid (red circle) and the CMOF derived centroid (centroid associated with the dashed red line) subject to the constraint that the reference MOF volume fraction (material 1 volume fraction in the center cell) equals the actual CMOF reconstructed volume fraction which is the material 1 volume fraction below the solid red line in the center cell.

numerical methods for deforming boundary problems in computational fluid  
70 dynamics in general, and show where our new method fits in.

For many deforming boundary problems in fluid dynamics, “shock capturing” numerical methods are sufficient[22, 12, 64, 23, 52, 47]. Shock capturing methods have limitations. According to “Godunov’s theorem”[21], linear one-step second order accurate numerical methods for  $\phi_t + c\phi_x = 0$  cannot be monotonicity preserving unless  $|c|\Delta t/\Delta x$  is an integer. The concept of “monotonicity preserving” methods and its association to TVD (Total Variational Diminishing) methods was first drawn by Harten[23] who proved that (a) A monotone scheme is TVD and (b) a TVD scheme is monotonicity preserving. When Harten’s theory is taken together with Godunov’s theorem, one can see that “linear” second  
75 order TVD methods only exist under specialized situations. Harten’s and Godunov’s theory are manifested by the fact that stable shock capturing methods will invariably smear discontinuities over time.

In order to overcome the limitations of shock capturing methods, researchers have developed (i) front tracking methods[20, 61], (ii) front capturing methods[36,  
85 41, 59, 25, 9, 1, 2, 40, 44, 27], and (iii) shock fitting methods[46].

The above methods for overcoming the the time dependent “interface smearing” problem of the shock capturing methods, have been further improved by way of hybridization: the particle level set method[17], the hybrid front tracking and level set method[51], the coupled level set and volume of fluid method[58],  
90 and the coupled level set and moment of fluid method[5, 29].

The numerical method that we introduce in this article falls in the latter “hybrid” category in which we introduce the hybrid level set and Continuous Moment of Fluid (CMOF) method for multiphase/multimaterial flows. The level set method and the continuous moment-of-fluid method are hybridized as  
95 follows: (a) the smooth level set distance function representation is used to provide an initial guess for the CMOF slope reconstruction step (see Figure 3 and Section 4.2), and (b) the level set distance functions are in turn replaced by the exact signed distance to the continuous moment-of-fluid reconstructed interface (see Figure 3 and Section 4.3). The level set and CMOF representations  
100 are synchronized every time step.

### 3 Mathematical model

We simulate the flow of a multiphase system consisting of  $M_{fluid}$  fluid (deforming) materials and  $M_{rigid}$  non-deforming materials. The  $M_{fluid}$  materials tessellate the computational domain and in the case that a fluid material  $m_{fluid}$   
105 coincides with a rigid material,  $m_{rigid}$ , the rigid materials’ governing equations take precedence. Otherwise, in regions where fluid materials and rigid materials do not overlap, the fluid materials are governed by the incompressible Navier-Stokes equations of immiscible flows. We refer the reader to Figures 4, 5, and 6 for illustrations which distinguish between deforming (tessellating) materials and non-deforming (rigid) materials.  
110

▷ **Material domain and interface:** Mathematically, the domain of rigid

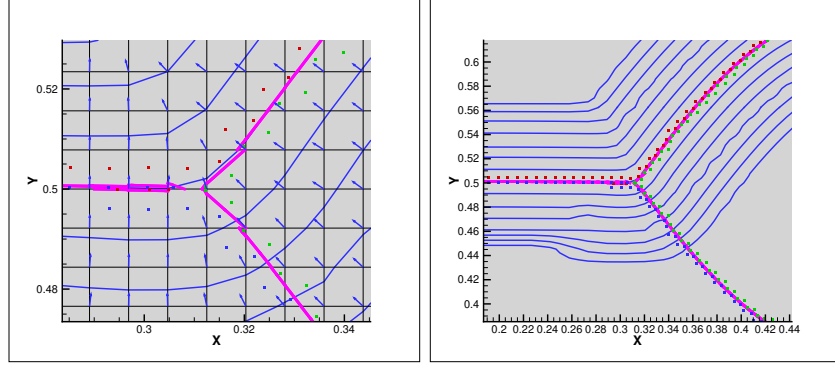


Figure 3: Illustration of the coupled level set and CMOF method corresponding to the liquid lens benchmark problem. LEFT: zoomed in view of (a) material 1 (signed distance function) level set contours (blue curves), (b) normal vectors corresponding to material 1 derived from the closest point map, (c) CMOF reconstructed interface for all 3 materials (pink curves), (d) material 1 centroids (red dots), (e) material 2 centroids (green dots), and (f) material 3 centroids (blue dots), RIGHT: zoomed out view. Notes:  $\Delta x = 1/128$ , contour spacing is 0.01, the initial location of the triple point ( $t = 0$ ) is intentionally offset from the grid lines (0.501, 0.501), tecplot level set data interpolated from the cell centers to the nodes.

*material*  $m_{rigid}$  is the region in which  $\phi_{m_{rigid}} > 0$ :

$$\phi_{m_{rigid}}(\mathbf{x}, t) = \begin{cases} > 0 & \mathbf{x} \in \text{material } m_{rigid}, \\ \leq 0 & \text{otherwise,} \end{cases} \quad (1)$$

$\mathbf{x}$  is the position vector in space and  $t$  is time. The domain of *material*  $m_{fluid}$  is the region in which  $\phi_{m_{fluid}} > 0$  and  $\phi_{m_{rigid}} < 0$ :

$$\phi_{m_{fluid}}(\mathbf{x}, t) = \begin{cases} > 0 & \mathbf{x} \in \text{material } m_{fluid} \cup m_{fluid,ghost}, \\ \leq 0 & \text{otherwise,} \end{cases} \quad (2)$$

The *interface* level set,  $\phi_{m1,m2}$  represents the interface between materials  $m1$  and  $m2$

$$\phi_{m1,m2}(\mathbf{x}, t) = \begin{cases} > 0 & \mathbf{x} \in \text{material } m1, \\ < 0 & \mathbf{x} \in \text{material } m2, \\ = 0 & \mathbf{x} \text{ along } (m1, m2) \text{ interface.} \end{cases} \quad (3)$$

The normal and curvature defined based on these level set functions are:

$$\mathbf{n}_{m1,m2} = \frac{\nabla \phi_{m1,m2}}{|\nabla \phi_{m1,m2}|}, \quad \kappa_{m1,m2} = \nabla \cdot \frac{\nabla \phi_{m1,m2}}{|\nabla \phi_{m1,m2}|}. \quad (4)$$

▷ **Conservation of mass:** We assume that each fluid material,  $m_{fluid}$ , is incompressible, so that the velocity field  $\mathbf{u} = (u, v, w)$  is divergence free within the bulk of each fluid material:

$$\nabla \cdot \mathbf{u} = 0. \quad (5)$$

In order to account for phase change or otherwise sources and sinks of mass, we have the following conditions on  $\nabla \cdot \mathbf{u}$ :

$$\nabla \cdot \mathbf{u} = \sum_{\text{sources}} \frac{\dot{m}_{\text{source}}}{\rho_{\text{source}}} \delta(\phi_{m_{\text{source}}}) - \sum_{\text{sinks}} \frac{\dot{m}_{\text{sink}}}{\rho_{\text{sink}}} \delta(\phi_{m_{\text{sink}}})$$

$H(\phi)$  is the Heaviside function which is defined as,

$$H(\phi) = \begin{cases} 1 & \phi > 0 \\ 0 & \phi \leq 0 \end{cases}$$

$\delta(\phi)$  is the Dirac Delta function,

$$\delta(\phi) = H'(\phi).$$

115 For boiling examples,  $\dot{m}$  is the mass flux of boiling liquid across the liquid/vapor interface,

$$\dot{m} = \frac{k_l \nabla T_l \cdot \mathbf{n}_{l,v} - k_v \nabla T_v \cdot \mathbf{n}_{l,v}}{L},$$

120 where  $k_l$  and  $k_v$  are the thermal conductivities in the liquid and ambient vapor regions respectively,  $\rho_l$  and  $\rho_v$  are the densities in the liquid and ambient vapor regions respectively,  $L$  is the latent heat of vaporization, and  $\mathbf{n}_{l,v}$  is the interface normal vector pointing from the ambient vapor region into the liquid,

$$\mathbf{n}_{l,v} = \frac{\nabla \phi_{l,v}}{|\nabla \phi_{l,v}|}.$$

▷ **Conservation of momentum:** The conservation of momentum for each material in its domain is given by

$$\begin{aligned} (\mathbf{u} \rho_m)_t + \nabla \cdot (\mathbf{u} \otimes \mathbf{u} \rho_m + p_m \mathbb{I}) = \\ \nabla \cdot (2\mu_m \mathbb{D}) + \rho_m \mathbf{g} (1 - \alpha_m (T_m - T_{0m})) \quad \text{if } \phi_m(\mathbf{x}, t) > 0, \end{aligned} \quad (6)$$

125 where  $p_m$ ,  $T_m$ ,  $\alpha_m$ , and  $\mu_m$  are pressure, temperature, coefficient of thermal expansion, and viscosity of material  $m$  respectively,  $\mathbf{g}$  is the gravitational acceleration vector, and  $\mathbb{D} = (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)/2$  is the rate of deformation tensor.



▷ **Conservation of energy:** The conservation of energy for each material in its domain is given by

$$(\rho_m C_{p,m} T_m)_t + \nabla \cdot (\mathbf{u} \rho_m C_{p,m} T_m) = \nabla \cdot (k_m \nabla T_m) \quad \text{if } \phi_m(\mathbf{x}, t) > 0, \quad (7)$$

where  $C_{p,m}$  and  $k_m$  are heat capacity and thermal conductivity of material  $m$  respectively, and  $T_m$  is the temperature.

130 ▷ **Interfacial jump condition:** Here, we write out the general equations for a deforming interface changing phase. We define  $m_s$  and  $m_d$  as the material id's associated with a "source" material (e.g. boiling liquid or freezing liquid) and "destination" material (e.g. vapor from boiling or ice from freezing) respectively. The location of the interface separating a  
135 material  $m_s$  region from a material  $m_d$  region is governed by the level set equation,

$$\phi_{m_s, m_d, t} + \mathbf{u}_{m_s} \cdot \nabla \phi_{m_s, m_d, t} = -\frac{\dot{m}}{\rho_s} |\nabla \phi_{m_s, m_d, t}| \quad (8)$$

$$\phi_{m_d, m_s, t} + \mathbf{u}_{m_s} \cdot \nabla \phi_{m_d, m_s, t} = \frac{\dot{m}}{\rho_s} |\nabla \phi_{m_d, m_s, t}| \quad (9)$$

An equivalent expression for the level set governing equations is:

$$\phi_{m_s, m_d, t} + \mathbf{u}_{m_d} \cdot \nabla \phi_{m_s, m_d, t} = -\frac{\dot{m}}{\rho_d} |\nabla \phi_{m_s, m_d, t}| \quad (10)$$

$$\phi_{m_d, m_s, t} + \mathbf{u}_{m_d} \cdot \nabla \phi_{m_d, m_s, t} = \frac{\dot{m}}{\rho_d} |\nabla \phi_{m_d, m_s, t}| \quad (11)$$

140 The interface jump conditions, between two materials  $m_s$  and  $m_d$ , for the velocity, pressure, and temperature, respectively, are,

$$\begin{aligned} \mathbf{u}_{m_s} \cdot \mathbf{n}_{m_s, m_d} - \mathbf{u}_{m_d} \cdot \mathbf{n}_{m_s, m_d} &= \dot{m} \left( \frac{1}{\rho_{m_d}} - \frac{1}{\rho_{m_s}} \right), \\ (p_{m_s} \mathbb{I} - p_{m_d} \mathbb{I}) \cdot \mathbf{n}_{m_s, m_d} &= -\sigma_{m_s, m_d} \kappa_{m_s, m_d} \mathbf{n}_{m_s, m_d} + \\ &\quad (2\mu_{m_s} \mathbb{D}_{m_s} - 2\mu_{m_d} \mathbb{D}_{m_d}) \cdot \mathbf{n}_{m_s, m_d}, \\ T_{m_s} &= T_{m_d}. \end{aligned} \quad (12)$$

$\kappa_{m_s, m_d}$  is the interface curvature and is defined as,

$$\kappa_{m_s, m_d} = \nabla \cdot \frac{\nabla \phi_{m_1, m_2}}{|\nabla \phi_{m_1, m_2}|}.$$

At a triple point junction a three-phase equilibrium known as the Neumann's triangle[14] determines the angles (see Figure 7.a):

$$\frac{\sin(\theta_1)}{\sigma_{23}} = \frac{\sin(\theta_2)}{\sigma_{13}} = \frac{\sin(\theta_3)}{\sigma_{12}}. \quad (13)$$

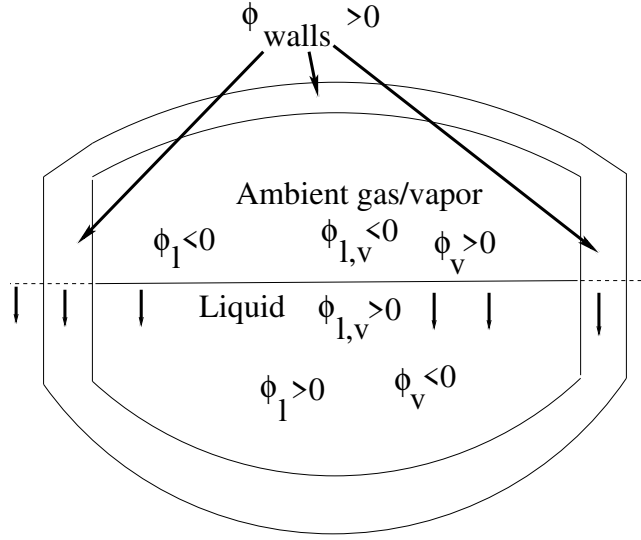


Figure 4: Level set functions for the simulation of evaporation in a Cryogenic Fuel tank. The liquid and vapor deforming materials tessellate the computational domain. The tank walls are rigid.  $\phi_{\text{walls}} > 0$  in the tank walls,  $\phi_{l,v} < 0$  and  $\phi_v > 0$  in the ambient vapor region, and  $\phi_{l,v} > 0$  and  $\phi_l > 0$  in the liquid region. The liquid/vapor interface is extended into the “tank walls region.” Following the surface tension model described in [33], page 471, “Stencil contains rigid boundary,” we calculate the curvature at the liquid/vapor/solid triple point by approximating  $\kappa = \nabla \cdot \mathbf{n}$  in which  $\mathbf{n}$  is assigned a strategic “ghost normal” in the solid region consistent with the contact angle condition. The interfacial curvature away from the liquid/vapor/solid triple point is computed using the Volume of fluid height function technique [56].

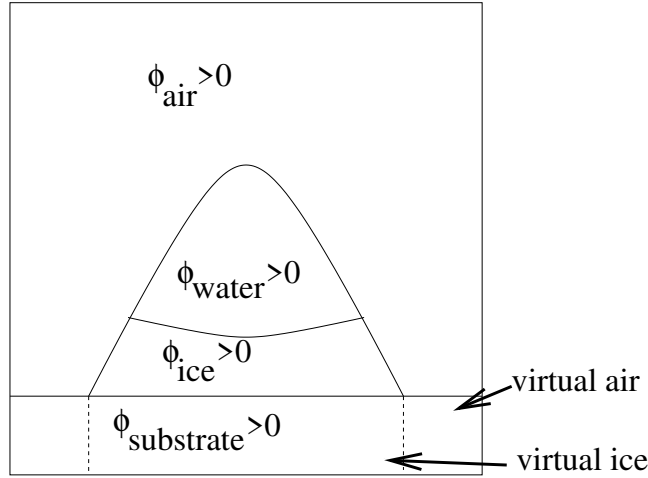


Figure 5: Level set functions for the simulation of the freezing of a water drop on top of a cold substrate. The temperature in the substrate,  $T_w$ , is prescribed. The ice, water, and air deforming materials tessellate the computational domain.  $\phi_{\text{substrate}} > 0$  in the cold substrate,  $\phi_{\text{ice}} > 0$  in the ice and  $\phi_{\text{water}} > 0$  in the water. The ice/air interface is extended into the substrate. Note: the ice is a deforming material solely due to phase change at the ice/water interface. Following the freezing model presented in [35], the ice and water materials are considered as the same material when calculating the surface tension force at the water/ice/air triple point. In other words, the interfacial curvature both at and away from triple points is computed using the Volume of fluid height function technique [56].

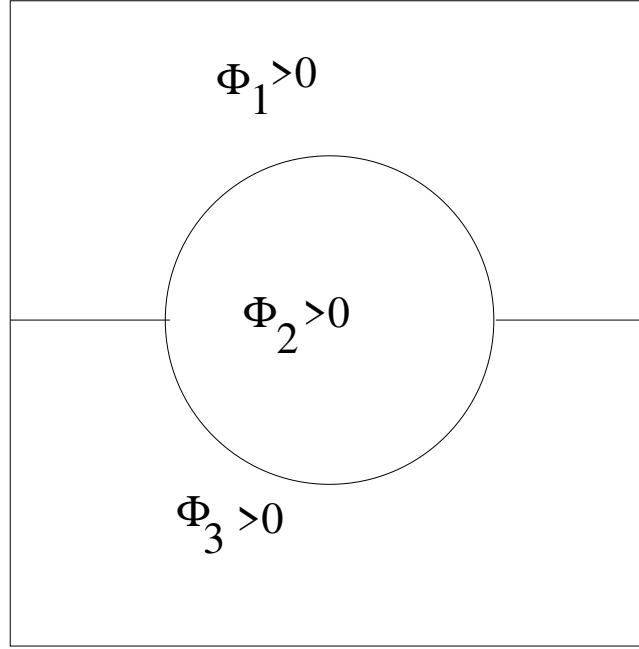


Figure 6: Level set functions for the simulation of the stretching of a “liquid lens” due to surface tension. The lens material  $\phi_2$  and surrounding materials,  $\phi_1$  and  $\phi_3$ , are all deforming fluids so that these three materials tessellate the computational domain. Following the surface tension model described in [33], page 472, “Stencil contains third fluid,” the surface tension force at the triple point(s) are calculated as  $\sum_{m=1}^M \gamma_m \kappa_m \nabla H(\phi_m)$  in which  $\kappa_m$  is approximated by way of finite differences of the level set function(s)  $\phi$ . The interfacial curvature away from triple points is computed using the Volume of fluid height function technique [56].

## 4 Numerical methods and algorithms

We describe our numerical method for the 2D uniform rectangular Cartesian grid case. In the problem domain, a computational cell,  $\Omega_{i,j}$ , is defined as,

$$\Omega_{i,j} = \left\{ \mathbf{x} : x \in \left[ x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2} \right], \right. \quad (14)$$

$$\left. y \in \left[ y_i - \frac{\Delta y}{2}, y_i + \frac{\Delta y}{2} \right] \right\} \quad (15)$$

where  $\mathbf{x}_{i,j} = \{x_i, y_j\}$  is the center of the cell  $\Omega_{i,j}$ . The domain of material  $m$  in a cell at time  $t^n$  is denoted by  $\Omega_{m,i,j}^n$ , and the zeroth and first order moments of the  $m$ th material distribution, corresponding to the volume fraction and centroid position, are defined as,

$$F_{m,i,j}^n = \frac{\int_{\Omega_{m,i,j}^n} d\Omega}{V_{i,j}}, \quad \mathbf{x}_{m,i,j}^{c,n} = \frac{\int_{\Omega_{m,i,j}^n} \mathbf{x} d\Omega}{V_{i,j,m}^n}, \quad (16)$$

where the computational cell volume is  $V_{i,j} = \int_{\Omega_{i,j}} d\Omega$ , and volume of the portion for material  $m$  in a computational cell is  $V_{i,j,m}^n = \int_{\Omega_{m,i,j}^n} d\Omega$ . The discretization for a typical point around the triple point is shown in Figure 7.

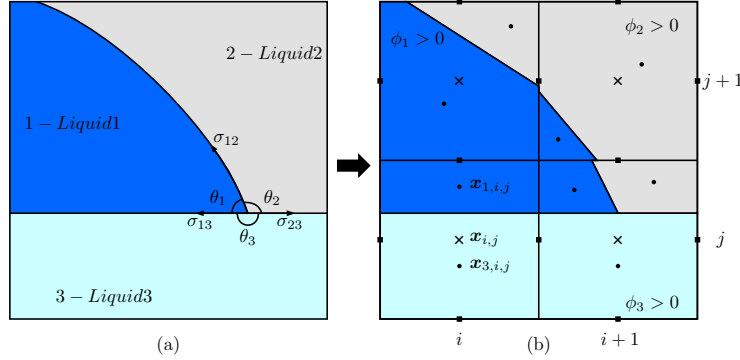


Figure 7: (a) Physical domain: a part of the problem domain containing a triple point. Surface tension forces are at equilibrium determining the contact angles. (b) Discretized domain: crosses show cell centers, circles represent cell centroids and horizontal and vertical rectangles depict the MAC grid points in each cell.

### 4.1 method overview: Staggered grid Projection Method

Referring to Figure 8 and also [43, 63], we discretize the velocity on the marker-and-cell (MAC) grid and we discretize pressure, temperature, level set function(s), volume fraction(s), and centroid(s) at the center of grid cells.

An outline of our operator split method is as follows ([63]):

▷ **Calculate the time step  $\Delta t = t^{n+1} - t^n$ :**

$$\Delta t_1 = \min_{d=1,\dots,D} \frac{\min \Delta x_d}{2D \max |u_d^n|} \quad D = 2 \text{ or } 3$$

$$\Delta t_2 = \min_{d=1,\dots,D} \frac{\min \Delta x_d}{D \max |u_d^{\text{phase change}, n-1}|}$$

$$\Delta t_3 = \min_{d=1,\dots,D} \min_{m_1, m_2=1,\dots,M} \Delta x_d^{3/2} \sqrt{\frac{\rho_{m_1} + \rho_{m_2}}{2\pi\sigma_{m_1, m_2}}} \quad (17)$$

$$\Delta t = \text{CFL} \min(\Delta t_1, \Delta t_2, \Delta t_3) \quad \text{CFL} = 1/2$$

160 ▷ **Cell Integrated Semi-Lagrangian (CISL) MAC velocity advection (see [43]):**

$$(\rho \mathbf{u})_t + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = 0, \quad (18)$$

▷ **CISL temperature advection (liquid and gas materials, see [63]):**

$$(\rho_m C_{p,m} T_m)_t + \nabla \cdot (\rho_m C_{p,m} \mathbf{u} T_m) = 0, \quad m = 1, \dots, M \quad (19)$$

▷ **CISL level set advection:**

$$\phi_{m,t} + \mathbf{u} \cdot \nabla \phi_m = 0, \quad m = 1, \dots, M$$

165 ▷ **CISL advection of the Volume fractions and Centroids (see [29, 30, 33, 43, 63]):**

$$F_{m,t} + \nabla \cdot (\mathbf{u} F_m) = 0, \quad m = 1, \dots, M \quad (20)$$

170 ▷ **Coupling the level set functions to the volume fractions and Centroids:** The (continuous) moment-of-fluid reconstructed (see Section 4.2) slope initial guess uses the level set functions, and the level set functions are in turn replaced by the exact signed distance (see Section 4.3) to the continuous moment-of-fluid reconstructed interface.

▷ **Phase change velocity from material  $m_s$  (source) to material  $m_d$  (destination) (see [63] and [34]):**

$$\mathbf{u}^{\text{phase change}} = -\frac{\dot{m}}{\rho_{m_d}} \mathbf{n}_{m_d}, \quad (21)$$

▷ **Phase change: Level Set Advection:**

$$\phi_{m,t} + \mathbf{u}^{\text{phase change}} \cdot \nabla \phi_m = 0, \quad m = m_s \text{ or } m_d.$$

175

- ▷ **Phase change: unsplit CISL advection of the Volume fractions and Centroids** (see [63, 34]):

$$F_{m,t} + \nabla \cdot (\mathbf{u}^{\text{phase change}} F_m) = 0, \quad m = m_s \text{ or } m_d.$$

180

- ▷ **Phase change: Coupling the level set functions to the volume fractions and Centroids:** The (continuous) moment-of-fluid reconstructed (see Section 4.2) slope initial guess uses the level set functions, and the level set functions are in turn replaced by the exact signed distance (see Section 4.3) to the continuous moment-of-fluid reconstructed interface.
- ▷ **Phase change: Mass source redistribution:** Redistribute  $\dot{m}$  to the source material,  $m_s$ , side (e.g. liquid if boiling or freezing) [63].
- ▷ **Thermal diffusion** (see [63] for algorithmic details):

$$\frac{(\rho C_{p,m})^{\text{mix},n+1}}{\Delta t^{\text{swept}}} (T_m^{n+1} - T_m^{\text{advection}}) = \nabla \cdot (k_m \nabla T_m^{n+1}) \quad (22)$$

- ▷ **Viscosity** (see [43] for algorithmic details):

$$\frac{\rho^{\text{mix},n+1}}{\Delta t} (\mathbf{u}^* - \mathbf{u}^{\text{advection}}) = \nabla \cdot (2\mu \mathbb{D}^*) - \rho^{n+1} (\alpha^{n+1} (T^{n+1} - T_0)) g \quad (23)$$

185

- ▷ **Pressure Gradient (liquid and vapor regions)** (see [30, 33, 43, 63] for algorithmic details):

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{\nabla p^{n+1}}{\rho^{\text{MAC,mix},n+1}} + \mathbf{g} - \frac{\sum_{m=1}^M \gamma_m \kappa_m \nabla H(\phi_m)}{\rho^{\text{MAC,mix},n+1}} \quad (24)$$

$$\nabla \cdot \mathbf{u}^{n+1} = \sum_{\text{sources}} \frac{\dot{m}_{\text{source}}}{\rho_{\text{source}}} \delta(\phi_{m_{\text{source}}}) - \sum_{\text{sinks}} \frac{\dot{m}_{\text{sink}}}{\rho_{\text{sink}}} \delta(\phi_{m_{\text{sink}}})$$

For the case in which two materials,  $m1$  and  $m2$ , are present in a given  $3 \times 3$  stencil,

$$\gamma_{m1} = \gamma_{m2} = \frac{\sigma_{m1,m2}}{2}$$

190

and for the case when three materials,  $m1$ ,  $m2$ ,  $m3$ , are present in a given  $3 \times 3$  stencil,

$$\gamma_{m1} = \frac{\sigma_{m1,m2} + \sigma_{m1,m3} - \sigma_{m2,m3}}{2}$$

$$\gamma_{m2} = \frac{\sigma_{m1,m2} + \sigma_{m2,m3} - \sigma_{m1,m3}}{2}$$

$$\gamma_{m3} = \frac{\sigma_{m1,m3} + \sigma_{m2,m3} - \sigma_{m1,m2}}{2}$$

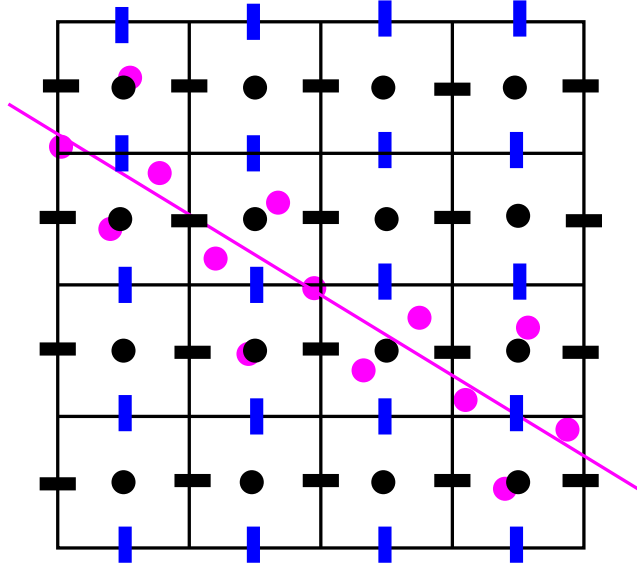


Figure 8: An illustration of the locations on our (staggered) computational grid in which fluid variables are approximated. Pressure, level set function(s), volume fraction(s), and centroids are discretized at the cell centers (filled in black circles). The horizontal velocity,  $u$ , is discretized at the midpoint of cell faces with constant  $x$  value (filled in black rectangles), and the vertical velocity,  $v$ , is discretized at the midpoint of cell faces with constant  $y$  value (filled in blue rectangles). The material centroids are denoted by the filled in magenta circles.



Remarks:

- 195 1. For step 2 “phase change” above, we use MOF interface reconstruction. Our rationale is that (1) the phase change velocity  $\mathbf{u}^{\text{phase change}}$  is not determined from the interface curvature, so that there is no instability risk due to MOF advection here, and (2) MOF is more accurate than CMOF in this scenario.
- 200 2. We use the multigrid preconditioned conjugate gradient method [60] to solve the large sparse matrix system that results from discretizing (22). The discretization of the Dirichlet interface temperature condition due to phase change uses the second order method described in [19].
- 205 3. We also use the multigrid preconditioned conjugate gradient (MGPCG) method [60] to solve the large sparse matrix system that results from discretizing (24). In order to simulate flows in sealed tanks or flows induced by the sealing of a valve [4] we have developed a general method for enforcing the solvability condition in each “enclosed” region. We identify each “enclosed” region using a shading method developed by [57].
- 210 4. In order to preserve mass for phase change problems, we must prescribe a stringent tolerance for the pressure equation (24). In some cases, due to round off error, the MGPCG method might stall. In order to overcome the effect of round-off error, we keep track of a history of the pressure admitting the least residual, and, when the solver has stalled, we restart the MGPCG process using the previous best guess pressure and increasing the number of multigrid relaxation steps by 1.
- 215

## 4.2 MOF and Continuous MOF interface reconstruction methods

The original MOF reconstruction method is local to the cell and uses the reference volume fraction,  $F_{\text{ref}} \equiv F_{m,i,j}^n$ , and reference centroid,  $\mathbf{x}_{\text{ref}}^c \equiv \mathbf{x}_{m,i,j}^n$  to find the linear (planer) interface reconstruction that has the volume fraction equal to  $F_{\text{ref}}$ , and has the least amount of error for centroid position. We find the actual volume fraction  $F_{\text{act}}(\mathbf{n}, b)$  and centroid  $\mathbf{x}_{\text{act}}^c(\mathbf{n}, b)$  for a reconstructed line(plane) with the normal  $\mathbf{n}$  and intercept  $b$  which minimizes

$$E_{\text{MOF}} = \|\mathbf{x}_{\text{ref}}^c - \mathbf{x}_{\text{act}}^c(\mathbf{n}, b)\|_2 \quad (25)$$

220 while  $F_{\text{act}}(\mathbf{n}, b) = F_{\text{ref}}$  (see Figure 9). Starting from a whole cell and repeating this process while only considering the uncaptured regions from previous steps, we can reconstruct the material interface for each phase in a multimaterial cell (see [33] for algorithm details). This tessellating procedure generates a volume preserving reconstruction at triple-points (see Figure 10).

225 The Continuous Moment-of-fluid method (CMOF) employs a similar procedure to find the interface reconstruction in a cell, but uses a different value for

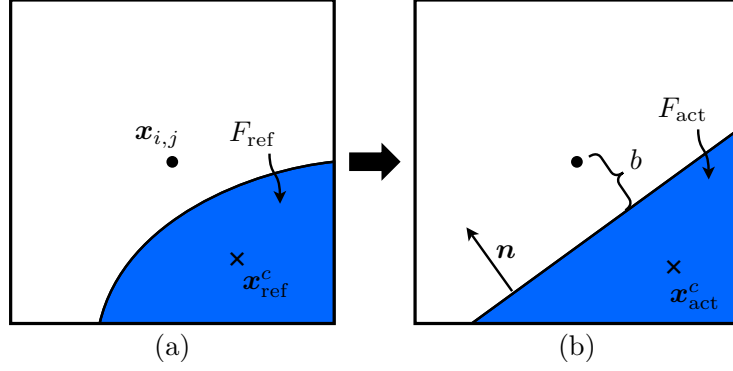


Figure 9: (a) Material domain at cell  $\{i, j\}$  for one of the phases, and corresponding reference volume fraction and centroid. (b) Piecewise linear reconstruction of the interface using MOF method. The line segment can be represented as  $\Omega_{i,j} \cap \{\mathbf{x} | \mathbf{n} \cdot (\mathbf{x} - \mathbf{x}_{i,j}) + b = 0\}$

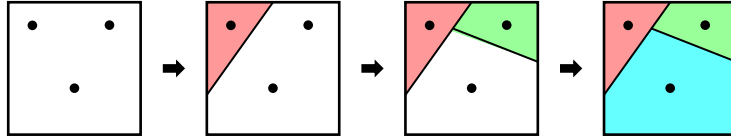


Figure 10: Volume-tessellating MOF reconstruction. Solid circles are centroids. White space is the unoccupied region, and three materials are shown with different colors.

the reference cell centroid  $\mathbf{x}_{\text{ref}}^c$ . We define a super cell

$$\Omega_{i,j}^s = \left\{ \mathbf{x} : x \in \left[ x_i - \frac{3\Delta x}{2}, x_i + \frac{3\Delta x}{2} \right], \right. \quad (26)$$

$$\left. y \in \left[ y_i - \frac{3\Delta y}{2}, y_i + \frac{3\Delta y}{2} \right] \right\} \quad (27)$$

with volume fraction and centroid

$$F_{m,i,j}^{s,n} = \frac{\sum_{i,j} F_{m,i,j}^n V_{i,j}}{\sum_{i,j} V_{i,j}}, \quad \mathbf{x}_{m,i,j}^{c,s,n} = \frac{\sum_{i,j} F_{m,i,j}^n \mathbf{x}_{m,i,j}^{c,n} V_{i,j}}{\sum_{i,j} F_{m,i,j}^n V_{i,j}}. \quad (28)$$

For the CMOF method we find the slope  $\mathbf{n}$  and intercept  $b$  such that  $F_{\text{act}}(\mathbf{n}, b) = F_{\text{ref}}$ ,  $E_{\text{MOF}} = \|\mathbf{x}_{\text{ref}}^c - \mathbf{x}_{\text{act}}^c(\mathbf{n}, b)\|_2$  is minimized, while  $F_{\text{ref}} \equiv F_{m,i,j}^n$ ,  $\mathbf{x}_{\text{act}}^c(\mathbf{n}, b)$  is measured relative to the super cell  $\Omega_{i,j}^s$ , and  $\mathbf{x}_{\text{ref}}^c \equiv \mathbf{x}_{m,i,j}^{c,s,n}$ . We refer the reader to Figure 2.

The initial starting guess for finding the optimal MOF or CMOF slope is determined from the optimal choice from the following (described for the 3D case):

$$1. (\Theta^{(1)}, \Phi^{(1)}) = \text{slope to angle} \left( \frac{\mathbf{x}_{\text{ref}} - \mathbf{x}_{\text{uncapt}}}{\|\mathbf{x}_{\text{ref}} - \mathbf{x}_{\text{uncapt}}\|} \right)$$

$$2. (\text{first cut only})$$

$$(\Theta^{(2)}, \Phi^{(2)}) = \text{slope to angle}(\mathbf{n}^{(\text{CLSVOF})}). \quad (29)$$

$\mathbf{n}^{(\text{CLSVOF})}$  is derived from the CLSVOF reconstructed slope [58].

$$3. (\text{first cut only})$$

$(\Theta^{(3)}, \Phi^{(3)})$  is determined by way of the following “regression” decision tree [10] machine learning process (see Figure 11) :

(a) at the very start of a simulation, randomly create  $n_{\text{sample}} = 100^3$  sample triplets for planes cutting through a given cell:

$$(\Theta^{\text{sample}}, \Phi^{\text{sample}}, F^{\text{sample}}).$$

(b) Each sample is associated with  $(\Theta^{\text{key}}, \Phi^{\text{key}}, F^{\text{key}})$  which is determined as

$$(\Theta^{\text{key}}, \Phi^{\text{key}}) = \text{slope to angle} \left( \frac{\mathbf{x}_{\text{ref}, \text{sample}} - \mathbf{x}_{\text{uncapt}}}{\|\mathbf{x}_{\text{ref}, \text{sample}} - \mathbf{x}_{\text{uncapt}}\|} \right) \\ F^{\text{key}} = F^{\text{sample}}$$

(c) Each associated triplet,  $(\Theta^{\text{key}}, \Phi^{\text{key}}, F^{\text{key}})$  with corresponding classification,

$$(\Theta^{\text{classify}}, \Phi^{\text{classify}}, F^{\text{classify}}) \equiv (\Theta^{\text{sample}}, \Phi^{\text{sample}}, F^{\text{sample}})$$

is added to the decision tree list.

- 250 (d) After all of the sample data has been classified, the decision tree list is split into two branches, determined by the median in the critical splitting direction ( $\Theta$ ,  $\Phi$ , or  $F$ ). See Figure 11. The critical direction maximizes the decrease in the classification variance between the parent and its associated two branches. (each branch having its own mean).
- 255 (e) The tree is split again into 4 branches, with the next critical direction determined to maximize the classification variance reduction. Note: the critical direction is the same for each branch on a given level.
- (f) the previous step is repeated recursively until each branch in the tree contains just one piece of data. The number of levels in the tree cannot exceed  $\text{ceiling}(\log_2 n_{\text{sample}})$ .
- 260 (g) for the prediction phase, the decision tree is traversed using the key,  $(\Theta^{\text{key}}, \Phi^{\text{key}}, F^{\text{key}})$  in which

$$(\Theta^{\text{key}}, \Phi^{\text{key}}) = \text{slope to angle} \left( \frac{\mathbf{x}_{\text{ref}} - \mathbf{x}_{\text{uncapt}}}{\|\mathbf{x}_{\text{ref}} - \mathbf{x}_{\text{uncapt}}\|} \right). \quad (30)$$

$$F^{\text{key}} = F_{\text{ref}}$$

$(\Theta^{(3)}, \Phi^{(3)})$  is the classification obtained from traversing the decision tree using the key from (30). The maximum number of decisions to be made in order to classify a key not contained in the training data set is  $\text{ceiling}(\log_2 n_{\text{sample}})$ .

265 Remarks:

- ▷ In 2D, at the very start of a simulation, one randomly creates  $n_{\text{sample}} = 100^2$  sample pairs for lines cutting through a given cell:  $(\Theta^{\text{sample}}, F^{\text{sample}})$ .
- ▷ In RZ, a 2D decision tree is created corresponding to each discrete value of  $r_i = (i + 1/2)\Delta x$ .
- 270 ▷ Analytical MOF reconstruction algorithms exist [38], but not analytical CMOF reconstruction algorithms.
- ▷ for the CMOF reconstruction algorithm,  $\mathbf{x}_{\text{ref}}$  and  $\mathbf{x}_{\text{uncapt}}$  are measured relative to the super cell  $\Omega_{i,j}^s$  (26).  $\mathbf{x}_{\text{uncapt}}$  is the centroid of the “uncaptured” (i.e. “unreconstructed”) region of  $\Omega_{i,j}^s$ . For the first cut,  $\mathbf{x}_{\text{uncapt}}$  is the centroid of  $\Omega_{i,j}^s$  (26).
- 275 ▷ if the super cell,  $\Omega_{i,j}^s$ , contains at most two materials, then our starting guess is guaranteed to lead to a second order reconstruction, regardless of the number of ensuing optimization iterations. This is because the CLSVOF slope (29) by itself leads to a second order method.
- 280 ▷ We have determined anecdotally that a machine learning sample size of  $N_{1D}^d = 100^d$  where  $d$  is the dimension of the optimization input data and  $N_{1D}$  represents a “sample size per dimension,” leads to

errors comparable to iterating to convergence. We have arrived at this sample size by studying the  $N_{1D}$  dependence of the symmetric difference error for “Zalesaks” problem (see Table 2).

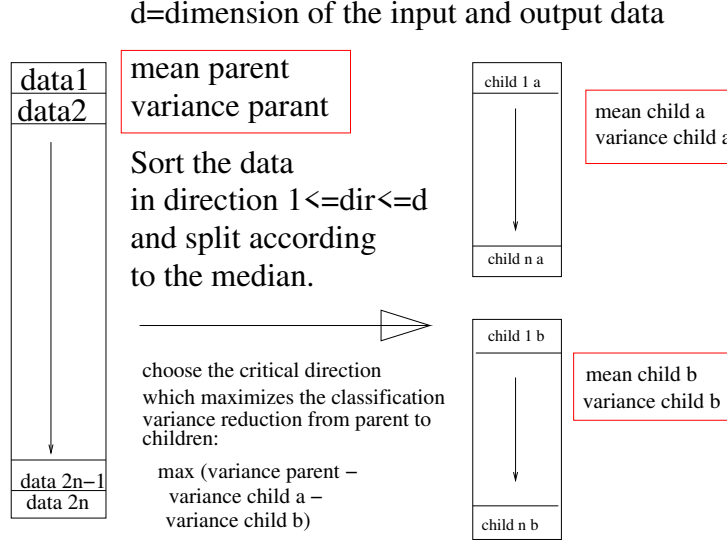


Figure 11: Illustration of regression decision tree splitting[10] for predicting the slope of the CMOF (or MOF) reconstructed interface given the reference centroid and volume fraction.

### 4.3 Reconstructing the distance function

Here we describe a distance function reconstruction algorithm that evaluate the exact signed distance function to the reconstructed interface.

Assuming that the reconstructed interface is available on the multimaterial cells, using a MOF or CMOF reconstruction, the algorithm first initialize the distance function to a large number with correct sign. Then in a narrow band the cells tagged for involvement in the redistancing procedure, that is, contributing information to and/or getting updated distance function value through this process. In the next steps, distances are evaluated from the center of an update cell to different possible interfacial points in its neighborhood, and material,  $\phi_m$ , and interface distance functions,  $\phi_{m1,m2}$ , are updated consequently.

- ▷ Set all distance functions,  $\phi_m$  and  $\phi_{m1,m2}$ , to a large negative value.
- ▷ Iterate over all cells.
- ▷ Using the reconstructed interface, change the sign of  $\phi_k$  and  $\phi_{k,m2}$  to positive if material  $k$  is occupying the center of a cell.

- 305      ▷ For a cell, check the neighbors for occupying materials in a  $3 \times 3 \times 3$  hypercube stencil around it. Count the materials  $m$  present if  $F_m > 0.5$  in at least one of the cells in the stencil, or if a material level set changes sign between two different cells in the stencil. If more than one material are present in the stencil, this cell is a *support* cell for the redistancing algorithm. Consequently, tag cells in the  $9 \times 9 \times 9$  hypercube around a support cell as *update* cells.
- 310      ▷ Iterate over update cells.
  - 315          ▷ Iterate over the support cells in the  $3 \times 3 \times 3$  hypercube around this update cell.
    - 320              ▷ Evaluate distance to the corners, face centers, and cell center in the support cell. This procedure gives the exact distance if cell boundaries are part of the material interface (Figure 12.a).
    - 325              ▷ Find the normal distance to each interface in the support cell. Update the distance only if the intersection point is within the support cell (Figure 12.b).
    - 330              ▷ Find the intersection of each pair of the interfaces in the support cell, and evaluate the distance to the triple-point. Check the materials around the intersection for updating the corresponding interface distance functions (Figure 12.c).
    - 335              ▷ Find the intersection of the interfaces with cell faces, evaluate the distance, and update the corresponding distance functions (Figure 12.d).
    - 340              ▷ For 3D cases only, find the intersection line of each pair of interfacial planes, and find the intersection of the cell boundaries with the line. Then, evaluate the distance to the intersection points, and update the corresponding distance functions (Figure 13).

Notes:

- 330      - In the redistancing algorithm, after finding the distance to an interface or intersection of interface with cell boundaries or other interfaces the materials around the intersection points are found by testing some points around the intersection to figure out which material they belong to. These points are picked by the combination of interface normal vectors involved in the intersection points and are shown with white diamonds in Figure 12 and Figure 13.
- 335      - Both distance function value and normal vector is updated if a distance measurement is found with smaller magnitude compared to the stored value in  $\phi_m$  and  $\phi_{m1,m2}$ .
- 340      - Applying the algorithm described above, an interface distance function  $\phi_{m1,m2}$  is not well defined when the update cell belongs to neither material  $m1$  nor material  $m2$  (Figure 14.a). An alternative measurement is used

to extend the interface distance functions (Figure 14.b). For a cell  $\{i, j\}$  where  $\phi_{m1,i,j} < 0$  and  $\phi_{m2,i,j} < 0$  we have:

$$\mathbf{x}_{m1} = \mathbf{x}_{i,j} - \phi_{m1,i,j} \mathbf{n}_{m1,i,j}, \quad (31)$$

$$\mathbf{x}_{m2} = \mathbf{x}_{i,j} - \phi_{m2,i,j} \mathbf{n}_{m2,i,j}, \quad (32)$$

$$\mathbf{q} = \mathbf{x}_{m1} - \mathbf{x}_{m2}. \quad (33)$$

$$\begin{cases} \phi_{m1,m2} = |\mathbf{q}| & \text{if } |\phi_{m1,i,j}| < |\phi_{m2,i,j}| \\ \phi_{m1,m2} = -|\mathbf{q}| & \text{if } |\phi_{m1,i,j}| \geq |\phi_{m2,i,j}| \end{cases} \quad (34)$$

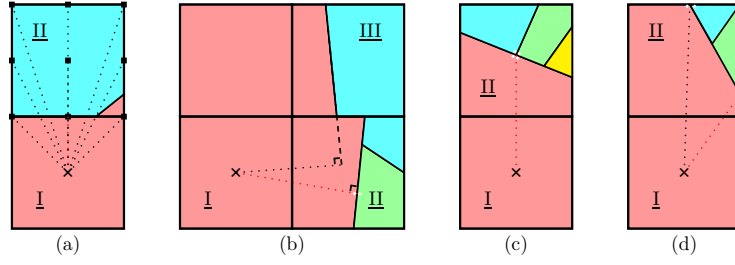


Figure 12: Redistancing procedures:(a) Update cell I, support cell II, distances are measured to the corners, face centers, and cell center in the support cell (dotted lines). Here, the distance to the bottom face center (red dotted line) is the true distance to the interface between two materials. (b) Update cell I, support cell II, normal distances (red dotted lines) is measured to the first interface. Since the intersection point is within the cell II, two materials around the interface is found using the testing points (white diamonds). Here, the signed distance and normal vector for the red and green material distance function and the red-green interface distance function are updated. However, when the case for update cell I and support cell III is considered, the normal distances (black dotted lines) is measured to the interface, and the intersection point is outside of cell III. Therefore, this measurement is not considered for redistancing update. (c) Update cell I, support cell II, the distance to the intersection of the first and second interfaces is measured (red dotted lines). The three materials around the triple-point is found checking the testing points (white diamonds). Here, the red, blue, and green material distance functions, and the red-blue, red-green, and blue-green interface distance functions are updated. (d) Update cell I, support cell II, distances to the intersection of the first interface with cell faces are measured (dotted lines). Here, for the red dotted line measurement, the red and green material distance functions, and the red-green interface distance functions are updated.

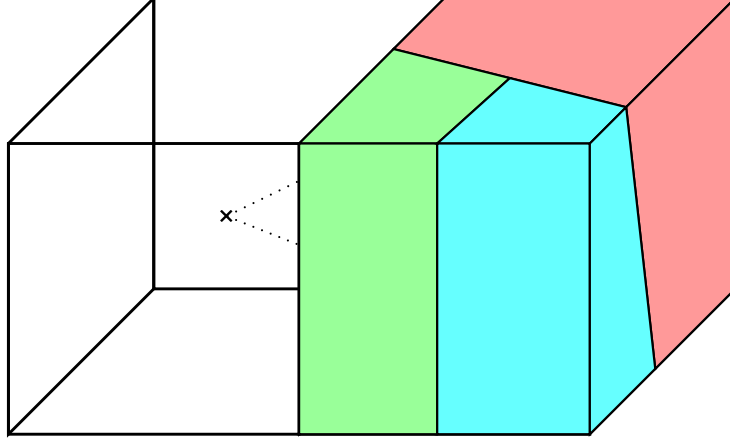


Figure 13: A specific distance measurement of the redistancing algorithm for a multimaterial cell in 3D. The intersection line (dashed line) of the two interface planes is found in the support cell. Then, the intersection of this line with cell face are found and the distance are evaluated from the center of the update cell to the intersection point. By testing the material around the intersection points (white diamonds) the corresponding material and interface distance functions are updated.

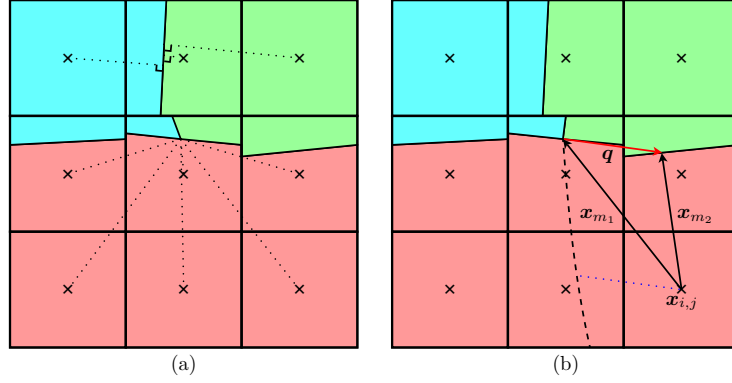


Figure 14: Redistancing of the interface distance functions. (a) Measurements are shown for updating the interface distance function between the blue and green materials. The measurements for the cell in red material leads to a singular point for the interface distance function at the triple point. (b) Evaluation of extended interface distance function between the blue and green materials (dashed line) for the cell  $\{i, j\}$ . Dotted blue line shows the corresponding evaluated distance.



## 345 5 Numerical tests and results

### 5.1 Zalesak problem

In this section we report results for “Zalesak’s” problem [67]. We have previously reported the symmetric difference error for this test problem when using MOF reconstruction[29]. Here, we compare MOF to CMOF, and decision tree machine learning to Gauss-Newton method for finding the optimal slope.

The problem setup for “Zalesak’s” problem is as follows:

- ▷ radius of notched disk=15 units.
- ▷ initial center of notched disk: (50,75)
- ▷ notch width: 5 units
- 355 ▷ notch height: 25 units
- ▷ prescribed velocity:  $u = -(\pi/314)(y - 50)$ ,  $v = (\pi/314)(x - 50)$

In Table 2, we report the symmetric difference error (35),

$$E_{sym} = |\Omega^{approx} \cap \Omega^{exact, complement}| + |\Omega^{approx, complement} \cap \Omega^{exact}|, \quad (35)$$

after one full rotation of the notched disk (time is 628.0 units). Figure 15 compares the MOF results to the CMOF results on a  $192^2$  grid. For all Gauss-Newton optimization results, we iterate until convergence with a tolerance of  $10^{-10}$ . For the decision tree machine learning optimization results, we choose the best starting guess between the “CLSVOF” slope, “decision tree slope,” and the “centroid slope.” There are no iterations necessary for the “decision tree optimization” results. In Table 3, we report the cost of the Gauss-Newton optimization algorithm vs. the “decision tree machine learning optimization.”

Table 2: Comparison of symmetric difference error: MOF versus CMOF and decision tree machine learning optimization (ML sample size equals  $100^2$  or  $50^2$ ) versus Gauss Newton (GN) optimization ( $t = 628$ ) for Zalesak’s problem.

$\Delta x$	$\Delta t$	MOF GN	MOF ML $100^2$	MOF ML $50^2$	CMOF GN	CMOF ML $100^2$	CMOF ML $50^2$
$\frac{100}{96}$	$\frac{628.0}{1155}$	6.8	8.1	9.6	19.0	18.9	18.9
$\frac{100}{192}$	$\frac{628.0}{2066}$	2.5	3.2	4.2	7.3	7.2	7.2

### 5.2 Bubble formation test problem; comparison of MOF with CMOF

In this section we report results for the simulation of bubble formation due to injection of gas through a nozzle at the bottom of the computational domain.

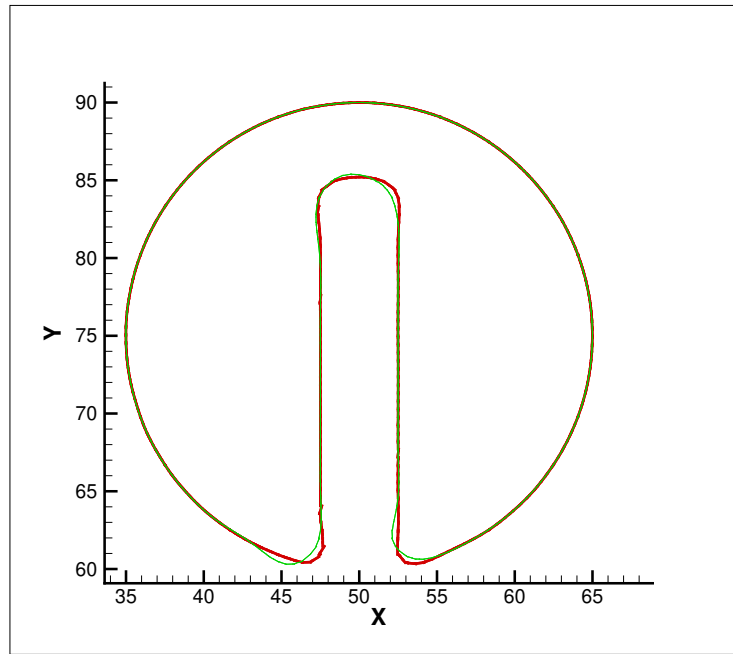


Figure 15: Zalesak's problem at  $t = 628.0$ . A comparison of MOF (red) with CMOF (green) (using "decision tree optimization") on a  $192^2$  grid.

Table 3: Comparison of reconstruction cost (average number of iterations, reconstruction time): MOF versus CMOF and decision tree machine learning optimization (ML, sample size equals  $100^3$ ) versus Gauss Newton (GN) optimization ( $t = 628$ ) for Zalesak’s problem.

$\Delta x$	$\Delta t$	MOF-GN	MOF-ML	CMOF-GN	CMOF-ML
100/96	628.0/1155	3.5, 0.008	0, 0.007	7.5, 0.05	0, 0.02
100/192	628.0/2066	3.7, 0.019	0, 0.016	6.2, 0.09	0, 0.04

370 The nozzle radius is 0.085cm. The effective fine grid resolution is  $\Delta x^{fine} = 0.010625\text{cm}$ .

See Figure 16 for the results using the CMOF method and Figure 17 for the results using MOF. The results that we report correspond to the Figure 1E experiments reported by [24] and the simulations reported in [39].

375 The expected effective bubble diameter, after the initial transients end, is  $d_{effective} = 0.499\text{cm}$ . We note in the results that there is unphysical noise and flotsam in the MOF interface due to problems that start at the nozzle. The CMOF method does not exhibit any noise.

### 5.3 Liquid lens test problem; comparison of MOF with CMOF

380 In this section we report results for the simulation of the stretching of a “liquid lens” due to surface tension forces at the junctions where three materials meet. We refer the reader to section 4.4 of [31] (phase field method) and section 4.4 of [33] (MOF method). In contrast to our previous simulations using the MOF method [33] for the Liquid Lens test, we intentionally add random noise to the  
385 initial centroids. Physically, due to the presence of surface tension and viscosity forces, the interface should become smooth, but the noise exhibited by the MOF results (see 19) never disappear.

The liquid lens test problem is setup as follows (see Figure 6): initially,  
390 there are three materials with material 2 occupying a circle of diameter 0.3 in the center of the domain, material 1 occupying the remaining top half of the computational domain, and material 3 occupying the remaining bottom half of the computational domain. All three materials have the same unit density and share the same viscosity of  $\mu = 1/60$ . The surface tension coefficients between  
395 materials 1 and 2 and between materials 2 and 3 were set to  $\sigma_{12} = 2/45$  and  $\sigma_{23} = 2/45$  respectively. The surface tension between materials 1 and 3 was set to  $\sigma_{13} = 5/90$ . Noise was artificially added to the centroids as follows,

$$\mathbf{x}^{ref,noise} = \mathbf{x}^{ref} + (2r - 1)\Delta x, \quad (36)$$

where  $r$  is a random variable sampled from a uniform distribution,  $0 \leq r \leq 1$ . The expected steady-state solution has material 2 being stretched into a lens  
400 shape with a major axis length of  $L_0^{exact} = 0.460$ . We refer the reader to Figures

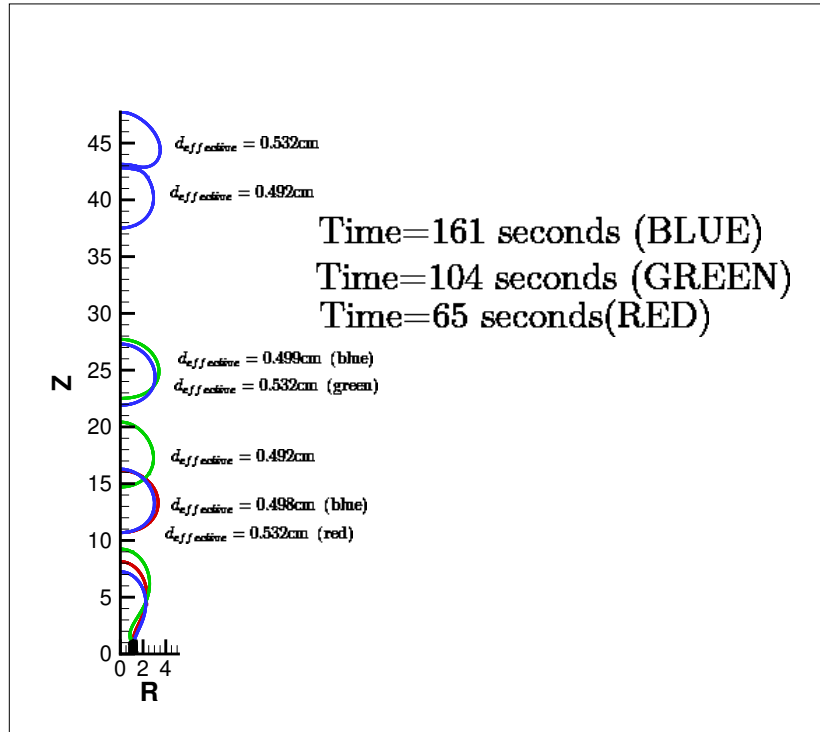


Figure 16: Computed results for the bubble formation problem using the CMOF method. The step numbers are 13400 ( $t = 65.2$ ), 21600 ( $t = 103.8$ ), and 34200 ( $t = 161.3$ ). The effective fine grid resolution is  $\Delta x^{fine} = 0.010625\text{cm}$

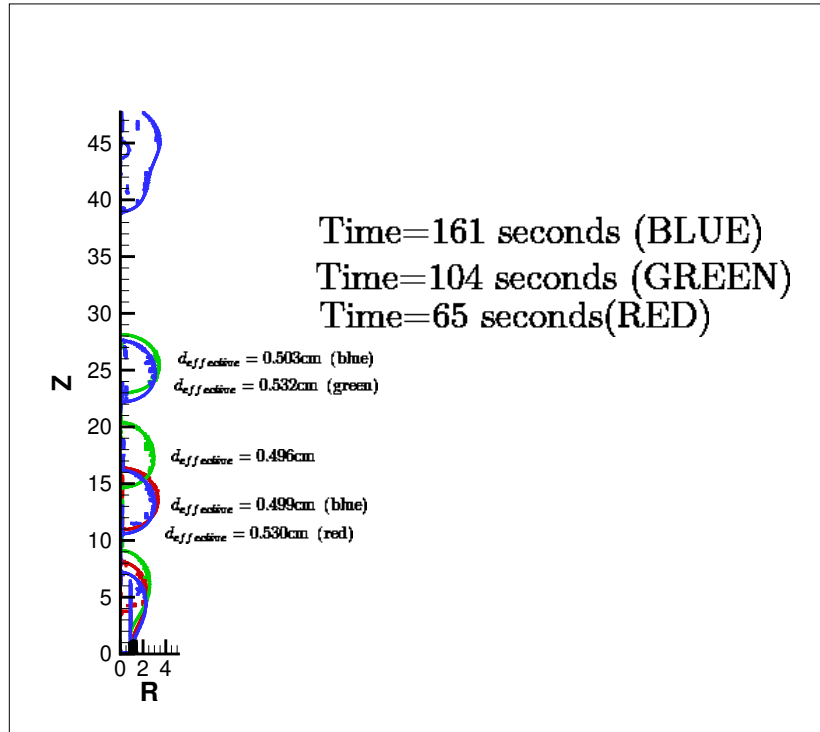


Figure 17: Computed results for the bubble formation problem using the MOF method. The step numbers are 13400 ( $t = 64.9$ ), 21600 ( $t = 103.4$ ), and 34800 ( $t = 161.1$ ). The effective fine grid resolution is  $\Delta x^{fine} = 0.010625\text{cm}$

18 (CMOF results), and 19 (MOF results), and Table 4 (comparison between MOF and CMOF for this liquid lens test problem).

Table 4: Comparison of MOF versus CMOF for the stretching of a liquid lens.  $L_0^{exact}$  is the expected steady state major axis length and  $L_0$  is the computed steady state major axis length. Material 2 is the lens material, material 1 is above the lens, and material 3 is below the lens. The initial centroids are randomly perturbed. For the MOF case, “flotsam” forms and migrates away from the “core” interface and is then truncated.

Method	$L_0$	$L_0^{exact}$	percent volume error $time = 4.0$
MOF	0.442	0.460	0.15 percent
CMOF	0.460	0.460	$7.0e - 8$ percent

See Figures 18 and 19.

#### 5.4 Freezing of a water droplet on a cold substrate; comparison with experiments and convergence study.

405

In this section we test our CMOF algorithm for simulating the freezing of a water droplet on a cold substrate. We refer the reader to Figure 5 which illustrates the initial conditions for our test. The temperature in the cold substrate is prescribed to be  $T_w$ . We shall compare our results with that which is reported in [26]. In Tables 5 and 6, we report the physical properties, in CGS units, used in our simulations.

410

Table 5: Physical properties of air, water, ice, and substrate for comparing with the icing experiments from [26].

variable	substrate	ice	water	air
density ( $\rho$ )	rigid	0.917	1.0	0.00129
viscosity ( $\mu$ )	rigid	rigid	$1.7e - 2$	$1.7e - 4$
spec. heat ( $C_p$ )	$T = T_w$	$2.03e + 7$	$4.21e + 7$	$1.0e + 7$
thermal cond. ( $k$ )	$T = T_w$	$2.2e + 5$	$0.55e + 5$	$0.026e + 5$
initial temp.	$T_w$	$T_w$	273.0	293.0

Table 6: Interfacial Physical properties for comparing with the icing experiments from [26].

variable	value
surface tension ( $\gamma$ )	72.8
physical latent heat ( $L$ )	$3.34e + 9$
modeling latent heat ( $L_{\text{model}}$ )	$65.0e + 9$
saturation temperature ( $T_{\text{sat}}$ )	273.0

Remarks:

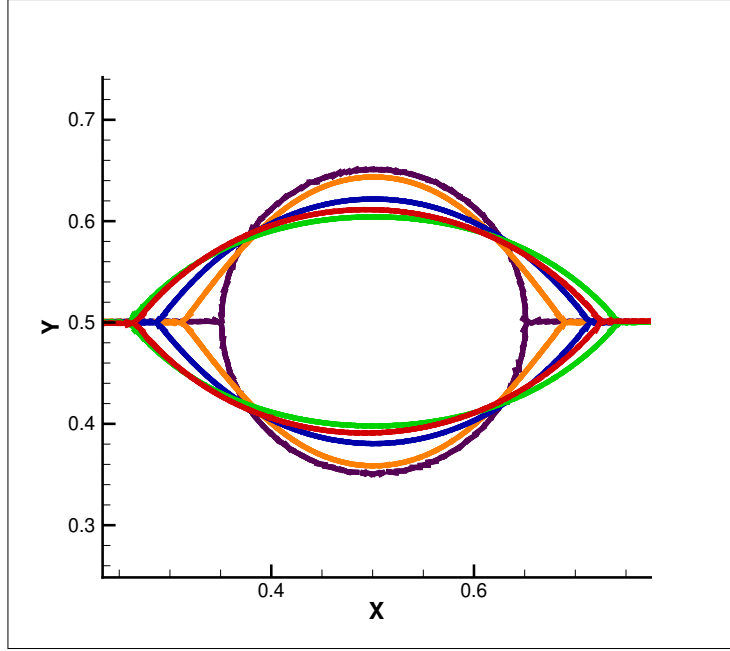


Figure 18: Stretching of a liquid lens computed using the CMOF method. Times displayed are 0.0 (brown), 0.187 (orange), 0.375 (blue), 0.749 (green), and 4.0 (red). Material 2 is the lens material, material 1 is above the lens, and material 3 is below the lens. The viscosity is the same for all 3 materials:  $\mu = 1/60$ .  $\sigma_{12} = \sigma_{23} = 2/45$ .  $\sigma_{13} = 5/90$ . The computational domain dimensions are  $1 \times 1$  and the effective fine grid resolution is  $128 \times 128$ . The initial centroids are randomly perturbed according to (36). The CMOF method, by design, will rapidly damp out the high frequency modes not recognizable by the curvature discretization algorithm.

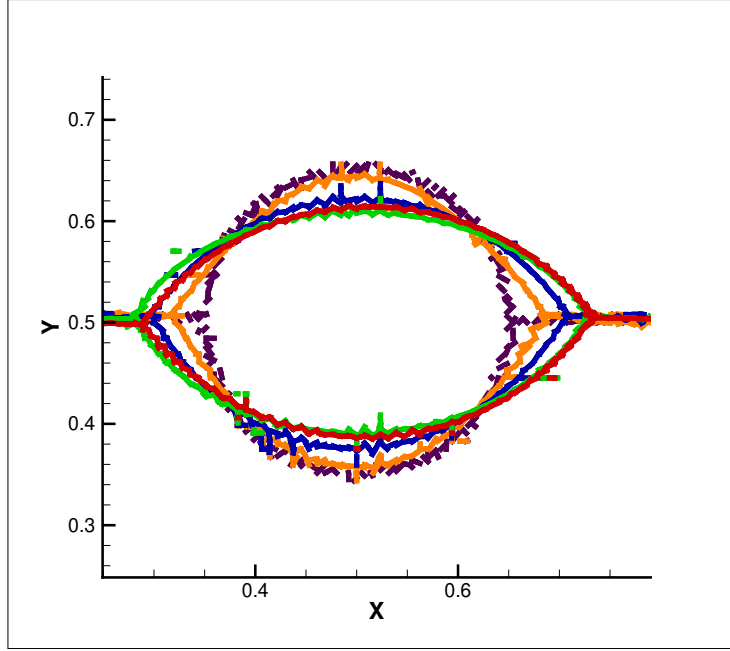


Figure 19: Stretching of a liquid lens computed using the MOF method. Times displayed are 0.0 (brown), 0.187 (orange), 0.375 (blue), 0.749 (green), and 4.0 (red). Material 2 is the lens material, material 1 is above the lens, and material 3 is below the lens. The viscosity is the same for all 3 materials:  $\mu = 1/60$ .  $\sigma_{12} = \sigma_{23} = 2/45$ .  $\sigma_{13} = 5/90$ . The computational domain dimensions are  $1 \times 1$  and the effective fine grid resolution is  $128 \times 128$ . The initial centroids are randomly perturbed according to (36). The MOF reconstruction method potentially generates “high frequency” signals unrecognized by conventional curvature discretization algorithms.



- ▷ In our icing simulations we intentionally add mass to the interface; we refer to sub-section 5.4.1 below for details of our added mass model. The surface tension time step constraint (17),

$$\Delta t < \Delta x^{3/2} \sqrt{\frac{\rho_L + \rho_G}{2\pi\sigma}},$$

is greatly relaxed by adding mass to the interface,

$$\Delta t < \Delta x^{3/2} \sqrt{\frac{\rho_{\text{added mass}}}{\pi\sigma}}. \quad (37)$$

We have heuristically chosen  $\rho_{\text{added mass}} = 1.0e + 6$  for simulations with  $L = 3.34e + 9$  and  $\rho_{\text{added mass}} = 1.0e + 9$  for simulations with  $L = 65.0e + 9$ . The criteria that we use in our choice of  $\rho_{\text{added mass}}$  is that the surface tension time step constraint (37) is maximally relaxed while the modified “numerical” characteristic surface tension wave speed,

$$\sqrt{\frac{\pi\sigma_{m1,m2}}{\Delta x\rho_{\text{added mass}}}}$$

is still maintained to be much faster than the characteristic speed associated with freezing (21). Similar strategies were taken in the simulation results reported by [50, 65, 62].

- ▷ In Table 7, we plot the time to freeze the droplet as a function of the wall temperature and using the physical value for latent heat. These results are consistent with the Stefan model (see Figures 22 and 23) but predict a freezing rate much faster than that which appears in the experiments [26]. In order to get agreement with experiments we artificially increased the latent heat (see  $L_{\text{model}}$  in Table 6) which then enabled us to get much closer agreement with experiments for all of the wall temperature cases. See Table 7.
- ▷ The “triple point” model for capillary forces is different from our previous research reported in [62]. Previously, we had prescribed an artificial “mushy zone” layer at the ice-water interface in which the ice was allowed to deform like a fluid. Also, previously, we had prescribed artificial surface tension coefficients for the ice-air and ice-water interfaces in order to enforce a static triple point contact angle condition. In the present work, we model the ice as a rigid solid, with the only exception that the ice-water interface is allowed to propagate due to phase change. The surface tension force at the triple point depends only on the water/gas surface tension coefficient. See Figure 5. Also, we refer the reader to the following article [35].

In order to verify that our numerical method is consistent with the “Stefan” freezing model, we compared our simulation results for the freezing of a planar

interface in the axisymmetric 3D  $R - Z$  coordinate system, with the results of the corresponding 1D Stefan problem in which a known “analytic” solution exists (see (38) and [66]):

$$\lambda e^{\lambda^2} \operatorname{erf}(\lambda) = \frac{cp(\theta_{wall} - \theta_{sat})}{h_{lg}\sqrt{\pi}} \quad (38)$$

$$x(t) = 2\lambda\sqrt{\alpha t} \quad \alpha = \frac{k}{\rho L}$$

In this test, the density ratio was set to one and the substrate temperature was prescribed as  $-6$  degrees Celsius. The latent heat was set to the physical value of  $3.34e + 9$  ergs per gram. Figure 22 shows our simulation results for this test problem and Figure 23 shows a comparison of the position of the water/ice front versus time between simulation (the curve labeled “Cylindrical\_ice”) and 1D analytical results (the curve labeled “1D\_ice”). The grid used in our simulation was  $64 \times 64$  grid cells. The maximum relative error between simulation and analytical solution is 1.1 percent.

The next test that we did was to check for convergence under grid refinement. In Figure 24 we compare the position of the water-ice interface (at  $r = 0$ ) for three different grid resolutions:  $32^2$ ,  $64^2$ , and  $128^2$ . For this test we used the physical value for Latent heat,  $L = 3.34e + 9$ , and the substrate temperature was maintained at  $-6$  degrees Celsius. In Figure 25, we display the interface(s) and temperature contours for the finest grid case,  $128^2$ . We note that the interface shape is as expected in that the water/air interface is smooth prior to the end stage of the liquid drop freezing process. The last frame shows the classical “tip” due to expansion of the ice.

Finally, in Table 7 we report on the time to completely freeze a water droplet as a function of the wall temperature (degrees Celsius) and Latent heat. We compare our simulation Results with experiments from [26]. We note that when the physical latent heat was used,  $3.34e + 9$ , the freezing times were much too fast. Since we get good agreement with the comparable 1D Stefan model, (see Figure 23, “Hemispherical ice” vs “1D ice”), we hypothesize that the discrepancy is due to incorrect modeling rather than a problem with our algorithm. Using our 1D model as a guide, we determined that increasing the latent heat to be  $65.0e + 9$  enabled us to have agreement for a wide range of prescribed substrate temperatures.

#### 5.4.1 Added Mass model for static freezing test problem

Our added mass model relies on the momentum equation being discretized in incremental gravity - pressure form. We replace (24) with

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{\nabla p^{\text{incremental}, n+1}}{\rho \text{MAC}_{\text{mix}, n+1}} - \sum_{m_1 < m_2} \mathbf{g} \cdot \mathbf{x} \frac{(\rho_{m_1} - \rho_{m_2}) \nabla H(\phi_{m_1})}{\rho \text{MAC}_{\text{mix}, n+1}} - \frac{\sum_{m=1}^M \gamma_m \kappa_m \nabla H(\phi_m)}{\rho \text{MAC}_{\text{mix}, n+1}} \quad (39)$$

where

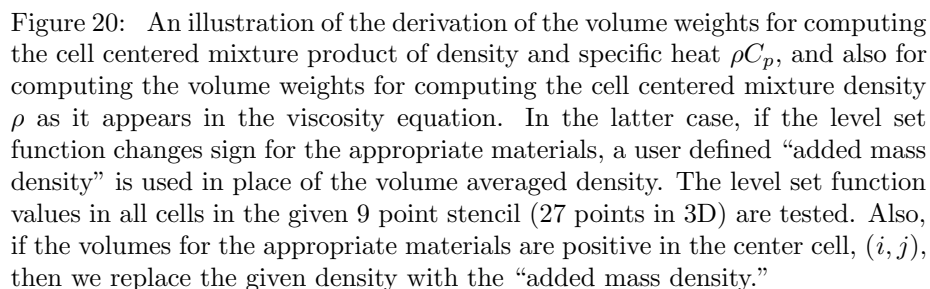
$$p = p^{\text{incremental}} + \rho \mathbf{g} \cdot \mathbf{x}$$

480 The cell-centered mixture of the density,  $\rho^{\text{mix}}$ , as it appears in the viscosity equation (23), is determined from volume weights in a given cell  $(i, j)$ . The volume weights are derived from the CMOF reconstructed interface (see [43] or Figure 20). If the level set function changes sign for an appropriate pair of materials, the user defined “added mass density,”  $\rho^{\text{added mass}}$ , is used in place  
485 of the volume averaged density. The level set function values in all cells in the given 9 point stencil (27 points in 3D) are tested. Also, if the volumes for the appropriate materials are positive in the center cell,  $(i, j)$ , then we replace the given density with the “added mass density.”

The “MAC” grid mixture of the density,  $\rho^{\text{MAC,mix}}$ , as it appears in the  
490 incremental pressure gradient equation (39), is determined from volume weights in a given MAC grid control volume  $(i + 1/2, j)$ . The volume weights are derived from the CMOF reconstructed interface (see [43, 63] or Figure 21). If the level set function changes sign for the appropriate materials, a user defined “added mass density,”  $\rho^{\text{added mass}}$ , is used in place of the volume averaged density.  
495 The level set function values in cells  $(i, j)$  and  $(i + 1, j)$  (for the  $(i + 1/2, j)$  face) are tested. Also, if the “face” volumes for the appropriate materials are positive in the MAC grid control volume,  $(i + 1/2, j)$ , then we replace the given density with the “added mass density.”

Table 7: Time to freeze a water droplet with initial volume  $4.95e - 5\text{cm}^3$  as a function of the wall temperature (degrees Celsius) and Latent heat. Simulations sharply “capture” the triple point dynamics at the ice, water, and air interface using the hybrid PLS-MOF method. Results are compared with experiments from [26]. Much closer agreement is made with experiments when the latent heat is prescribed as  $65.0e + 9$  as opposed to when the latent heat is the physical value  $3.34e + 9$ .

Wall temperature	Latent heat	time (computed)	time (exp)
-2.0	$3.34e + 9$	1.7	32.5
-3.0	$3.34e + 9$	1.1	21.0
-4.0	$3.34e + 9$	0.8	16.0
-5.0	$3.34e + 9$	0.6	13.5
-2.0	$65.0e + 9$	31.8	32.5
-3.0	$65.0e + 9$	20.3	21.0
-4.0	$65.0e + 9$	14.9	16.0
-5.0	$64.0e + 9$	11.7	13.5



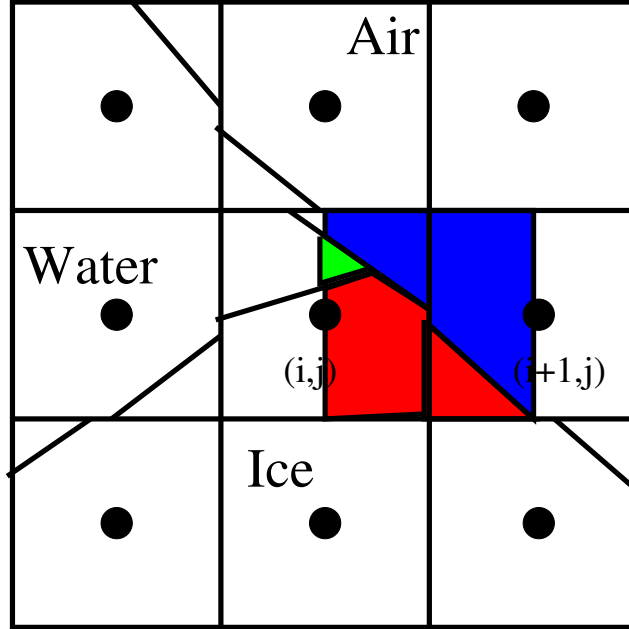


Figure 21: An illustration of the derivation of the volume weights for computing the “MAC” grid mixture of the density,  $\rho^{\text{MAC,mix}}$ , as it appears in the pressure gradient equation (24).  $\rho^{\text{MAC,mix}}$  is determined from volume weights in a given MAC grid control volume  $(i + 1/2, j)$ . The volume weights are derived from the CMOF reconstructed interface. If the level set function changes sign for the appropriate materials, a user defined “added mass density” is used in place of the volume averaged density. The level set function values in cells  $(i, j)$  and  $(i + 1, j)$  (for the  $(i + 1/2, j)$  face) are tested. Also, if the “face” volumes for the appropriate materials are positive in the MAC grid control volume,  $(i + 1/2, j)$ , then we replace the given density with the “added mass density.”

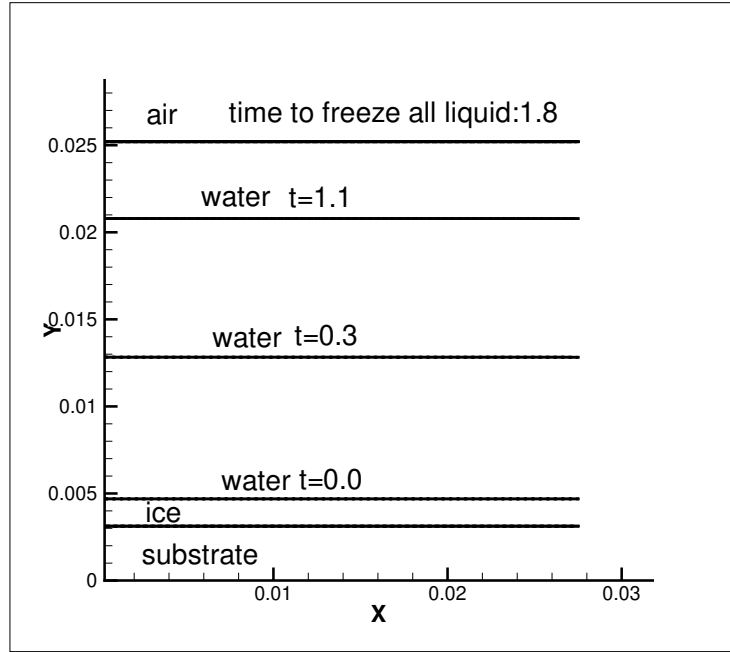


Figure 22: Freezing of a planar interface in R-Z coordinates. The substrate temperature is maintained at  $-2$  degrees Celsius. The Latent Heat is  $3.34e + 9$  ergs per gram and the grid size is  $64 \times 64$ .

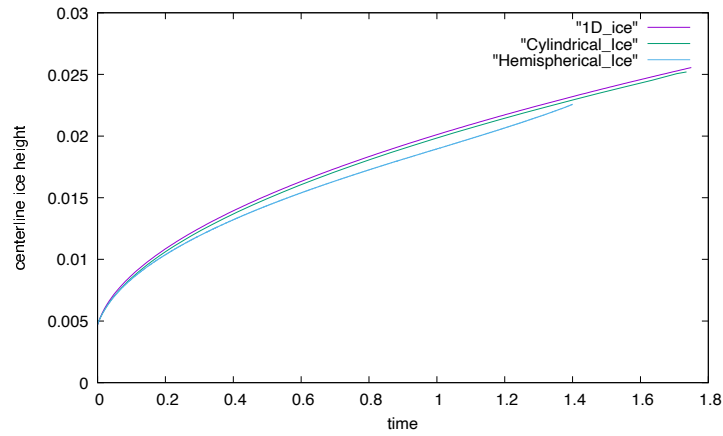


Figure 23: Verification of the Stefan model for freezing. The rate of freezing of a Hemispherical droplet is within 5 percent of the “comparable” cylindrical droplet freezing case. The Cylindrical freezing rate agrees within 1.1 percent of the 1D analytical model.

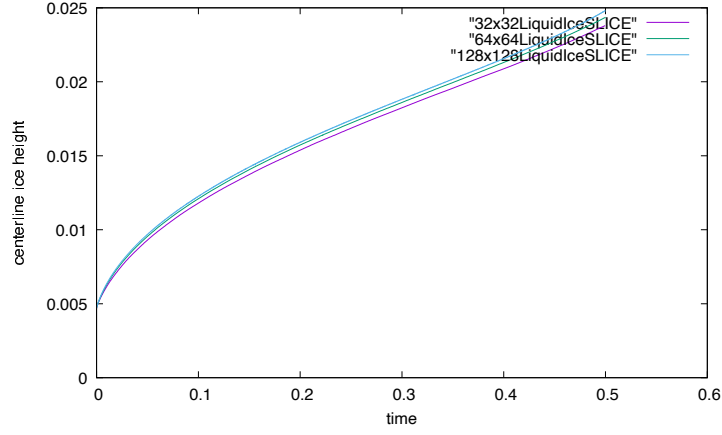


Figure 24: Convergence study for the solidification of a liquid drop on a cold substrate. The position of the liquid-ice interface is plotted versus time. The substrate temperature is maintained at  $-6$  degrees Celsius. The Latent Heat is  $3.34e + 9$  ergs per gram and the grid sizes are  $32 \times 32$ ,  $64 \times 64$ , and  $128 \times 128$ .

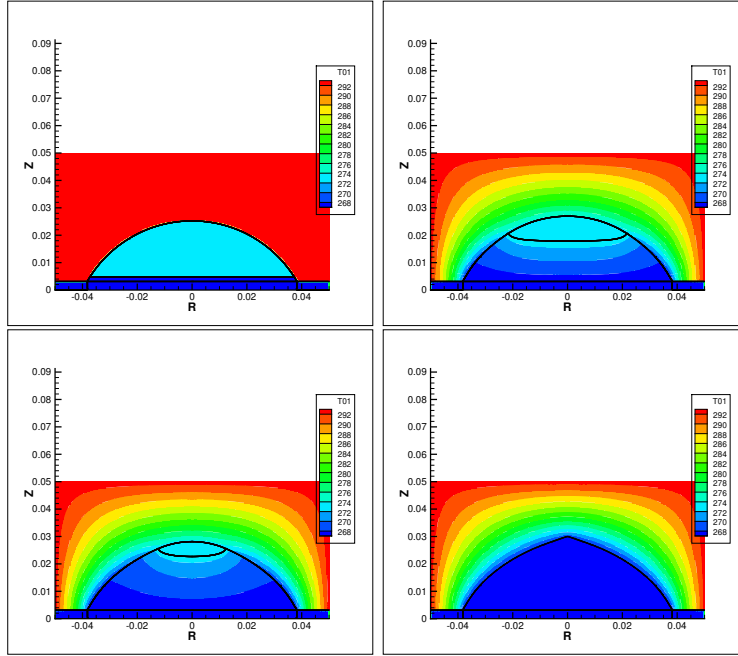


Figure 25: 4 snapshots of the liquid, gas, ice interfaces over time: left to right, top to bottom:  $t = 0$ ,  $t = 0.27$ ,  $t = 0.44$ , and  $t = 0.59$ . The substrate temperature is maintained at  $-6$  degrees Celsius. The Latent Heat is  $3.34e + 9$  ergs per gram and the grid size is  $128 \times 128$ .

## 5.5 Comparison of simulations with experiments for the break-up of a gas bubble due to a liquid jet in micro-gravity.

There were a number of experiments performed in microgravity (see [7] and [6]) in which a liquid jet was created in an experimental tank and the jet impinged upon a spherical gas bubble in the upper part of the tank. It was found by [7] and [6] that if the jet strength exceeded the following Weber number threshold,

$$We_j > 14,$$

the jet would penetrate the bubble, otherwise the jet might cause the bubble to become asymmetric,

$$1.5 \leq We_j \leq 14$$

or the jet would propel the bubble towards the top of the tank, without the bubble breaking up,

$$We_j \leq 1.5.$$

$We_j$  is defined as follows,

$$We_j = \frac{\rho_l V_0^2 R_0^2}{\sigma D_j},$$

where  $\rho_l$  is the liquid density,  $V_0$  is the liquid jet velocity at the nozzle,  $R_0$  is radius of the nozzle,  $\sigma$  is the surface tension coefficient, and  $D_j$  is the diameter of the jet when it impinges the bubble.

By trial and error, we have found that the Weber number cutoff in our simulation is:

$$We_j^{cutoff} = 5.0.$$

We refer the reader to the following figures: (A) a figure corresponding to a Weber number below the cutoff (see Figure 26,  $We_j = 4.875$ ) and (B) a figure corresponding to a Weber number above the cutoff (see Figure 27,  $We_j = 5.25$ ).

In order to test our new Decision Tree machine learning algorithm for finding the optimal CMOF slope, we recomputed the  $We_j = 5.25$  case except using the decision tree algorithm ( $100^3$  sample size) in order to generate a trial CMOF slope. In Table 8 and Figure 28 we compare results computed using our new decision tree machine learning algorithm with the results in which we iterate the Gauss-Newton method to convergence. We note that the overall savings per time step when using the decision tree reconstruction algorithm was 20 percent.

## 6 Conclusions

In this article the following results were reported for the first time:



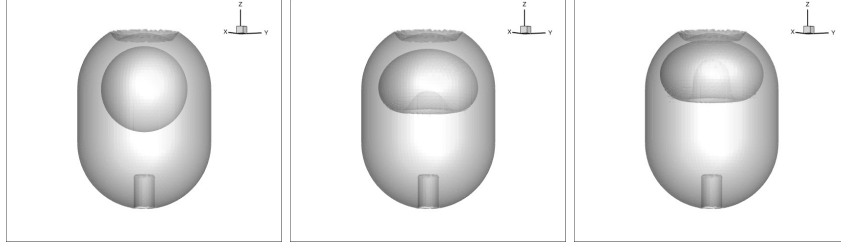


Figure 26: The deformation of a “spherical ullage” due to a liquid jet.  $We_j = 4.875$ . Times=0.0, 8.45, and 15.7. The jet does not penetrate the ullage in this case. The computational grid size was  $64 \times 64 \times 64$ .

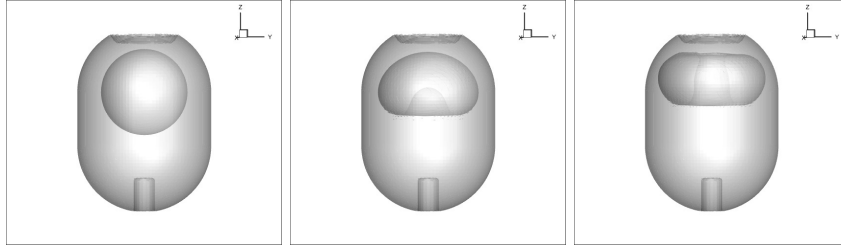


Figure 27: The deformation of a “spherical ullage” due to a liquid jet.  $We_j = 5.25$ . Times=0.0, 7.84, and 15.5. The jet penetrates the ullage in this case. The computational grid size was  $64 \times 64 \times 64$ .

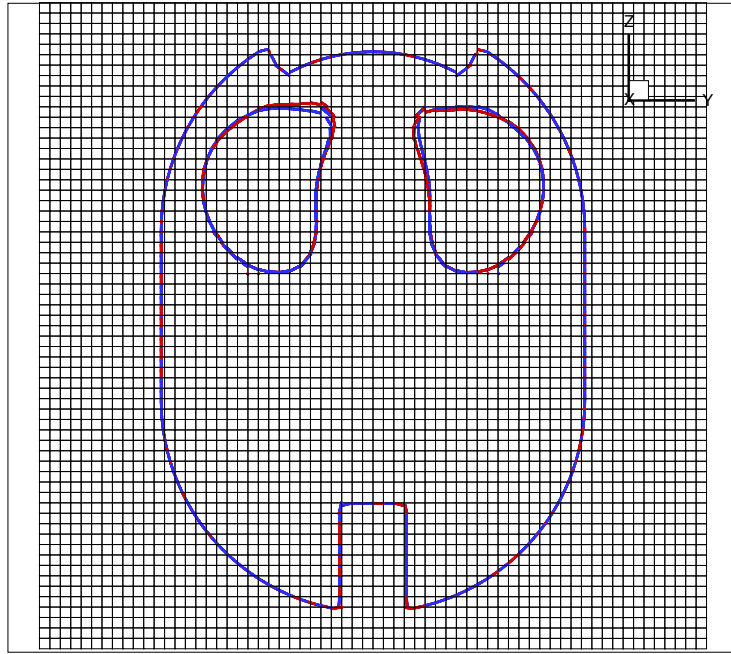


Figure 28: Comparison of the vapor-liquid interface just after break-up: decision tree machine learning optimization (ML) (blue) versus Gauss Newton optimization (GN) (Red) ( $t = 15.5$ ) for the break-up of a “spherical ullage” due to an impinging jet.

Table 8: Comparison of reconstruction cost (average number of iterations, reconstruction time): decision tree machine learning optimization (ML) versus Gauss Newton (GN) optimization ( $t = 15.5$ ) for the break-up of a “spherical ullage” due to an impinging jet.

Space resolution	Gauss Newton	Decision Tree
$64 \times 64 \times 64$	5.5, 3.9	0, 0.7

- 530
535
540
545
550
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
555
5

- [3] ANCELLIN, M., DESPRÉS, B., AND JAOUEN, S. Extension of generic two-component vof interface advection schemes to an arbitrary number of components. *Journal of Computational Physics* (2022), 111721.
- 560 [4] ARIENTI, M., AND SUSSMAN, M. An embedded level set method for sharp-interface multiphase simulations of diesel injectors. *International Journal of Multiphase Flow* 59 (2014), 1–14.
- [5] ASURI MUKUNDAN, A., MÉNARD, T., BRÄNDLE DE MOTTA, J. C., AND BERLEMONT, A. A hybrid moment of fluid–level set framework for simulating primary atomization. *Journal of Computational Physics* 451 (2022), 110864.
- 565 [6] BENTZ, M., KNOLL, R., HASAN, M., AND LIN, C. Low-g fluid mixing-further results from the tank pressure control experiment. In *29th Joint Propulsion Conference and Exhibit* (1993), p. 2423.
- [7] BENTZ, M., MESEROLE, J., AND KNOLL, R. Jet mixing in low gravity-results of the tank pressure control experiment. In *28th Joint Propulsion Conference and Exhibit* (1992), p. 3060.
- 570 [8] BONHOMME, R., MAGNAUDET, J., DUVAL, F., AND PIAR, B. Inertial dynamics of air bubbles crossing a horizontal fluid–fluid interface. *Journal of Fluid Mechanics* 707 (2012), 405–443.
- 575 [9] BRACKBILL, J. U., KOTHE, D. B., AND ZEMACH, C. A continuum method for modeling surface tension. *Journal of computational physics* 100, 2 (1992), 335–354.
- [10] BREIMAN, L., FRIEDMAN, J., OLSHEN, R., AND STONE, C. Classification and regression trees. monterey, ca: wadsworth & brooks/cole advanced books & software, 1984.
- 580 [11] CABOUSSAT, A., FRANCOIS, M. M., GLOWINSKI, R., KOTHE, D. B., AND SICILIAN, J. M. A numerical method for interface reconstruction of triple points within a volume tracking algorithm. *Mathematical and Computer Modelling* 48, 11 (2008), 1957–1971.
- 585 [12] COLELLA, P., AND WOODWARD, P. R. The piecewise parabolic method (ppm) for gas-dynamical simulations. *Journal of computational physics* 54, 1 (1984), 174–201.
- [13] CUMMINS, S. J., FRANCOIS, M. M., AND KOTHE, D. B. Estimating curvature from volume fractions. *Computers & structures* 83, 6-7 (2005), 425–434.
- 590 [14] DE GENNES, P.-G., BROCHARD-WYART, F., AND QUÉRÉ, D. *Capillarity and wetting phenomena: drops, bubbles, pearls, waves*. Springer Science & Business Media, 2013.

- 595 [15] DYADECHKO, V., AND SHASHKOV, M. Moment-of-fluid interface reconstruction. *Los Alamos report LA-UR-05-7571* (2005).
- [16] DYADECHKO, V., AND SHASHKOV, M. Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics* 227, 11 (2008), 5361–5384.
- 600 [17] ENRIGHT, D., FEDKIW, R., FERZIGER, J., AND MITCHELL, I. A hybrid particle level set method for improved interface capturing. *Journal of Computational physics* 183, 1 (2002), 83–116.
- [18] FRANCOIS, M. M., CUMMINS, S. J., DENDY, E. D., KOTHE, D. B., SICILIAN, J. M., AND WILLIAMS, M. W. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics* 213, 1 (2006), 141–173.
- 605 [19] GIBOU, F., FEDKIW, R. P., CHENG, L.-T., AND KANG, M. A second-order-accurate symmetric discretization of the poisson equation on irregular domains. *Journal of Computational Physics* 176, 1 (2002), 205–227.
- 610 [20] GLIMM, J., ISAACSON, E., MARCHESIN, D., AND MCBRYAN, O. Front tracking for hyperbolic systems. *Advances in Applied Mathematics* 2, 1 (1981), 91–119.
- [21] GODUNOV, S. Ph.d. dissertation: Different methods for shock waves moscow state university, 1954.
- 615 [22] GODUNOV, S. K., AND BOHACHEVSKY, I. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematičeskij sbornik* 47, 3 (1959), 271–306.
- [23] HARTEN, A. High resolution schemes for hyperbolic conservation laws. *Journal of computational physics* 135, 2 (1997), 260–278.
- 620 [24] HELSBY, F., AND TUSON, K. Behaviour of air bubbles in aqueous solutions. *Research* 8 (1955), 270–275.
- [25] HIRT, C. W., AND NICHOLS, B. D. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics* 39, 1 (1981), 201–225.
- 625 [26] HU, H., AND JIN, Z. An icing physics study by using lifetime-based molecular tagging thermometry technique. *International Journal of Multiphase Flow* 36, 8 (2010), 672–681.
- [27] HUANG, Z., LIN, G., AND ARDEKANI, A. M. Consistent, essentially conservative and balanced-force phase-field method to model incompressible two-phase flows. *Journal of Computational Physics* 406 (2020), 109192.
- 630

- [28] HUANG, Z., LIN, G., AND ARDEKANI, A. M. A consistent and conservative phase-field model for thermo-gas-liquid-solid flows including liquid-solid phase change. *Journal of Computational Physics* 449 (2022), 110795.
- 635 [29] JEMISON, M., LOCH, E., SUSSMAN, M., SHASHKOV, M., ARIENTI, M., OHTA, M., AND WANG, Y. A coupled level set-moment of fluid method for incompressible two-phase flows. *Journal of Scientific Computing* 54, 2-3 (2013), 454–491.
- 640 [30] JEMISON, M., SUSSMAN, M., AND ARIENTI, M. Compressible, multiphase semi-implicit method with moment of fluid interface representation. *Journal of Computational Physics* 279 (2014), 182–217.
- [31] KIM, J. Phase field computations for ternary fluid flows. *Computer methods in applied mechanics and engineering* 196, 45 (2007), 4779–4788.
- 645 [32] KUCHARIK, M., GARIMELLA, R. V., SCHOFIELD, S. P., AND SHASHKOV, M. J. A comparative study of interface reconstruction methods for multi-material ale simulations. *Journal of Computational Physics* 229, 7 (2010), 2432–2452.
- 650 [33] LI, G., LIAN, Y., GUO, Y., JEMISON, M., SUSSMAN, M., HELMS, T., AND ARIENTI, M. Incompressible multiphase flow and encapsulation simulations using the moment-of-fluid method. *International Journal for Numerical Methods in Fluids* 79, 9 (2015), 456–490.
- [34] LIU, Y., SUSSMAN, M., LIAN, Y., HUSSAINI, M. Y., VAHAB, M., AND SHOELE, K. A novel supermesh method for computing solutions to the multi-material stefan problem with complex deforming interfaces and microstructure. *Journal of Scientific Computing* 91, 1 (2022), 1–40.
- 655 [35] LYU, S., WANG, K., ZHANG, Z., PEDRONO, A., SUN, C., AND LEGENDRE, D. A hybrid vof-ibm method for the simulation of freezing liquid films and freezing drops. *Journal of Computational Physics* 432 (2021), 110160.
- 660 [36] MARKSTEIN, G. Interaction of flow pulsations and flame propagation. *Journal of the Aeronautical Sciences* 18, 6 (1951), 428–429.
- [37] MIAO, F., WU, B., SUN, Z., AND PENG, C. Calibration method of the laser beam based on liquid lens for 3d precise measurement. *Measurement* 178 (2021), 109358.
- 665 [38] MILCENT, T., AND LEMOINE, A. Moment-of-fluid analytic reconstruction on 3d rectangular hexahedrons. *Journal of Computational Physics* 409 (2020), 109346.
- 670 [39] OHTA, M., KIKUCHI, D., YOSHIDA, Y., AND SUSSMAN, M. Robust numerical analysis of the dynamic bubble formation process in a viscous liquid. *International Journal of Multiphase Flow* 37, 9 (2011), 1059–1071.

- [40] OLSSON, E., AND KREISS, G. A conservative level set method for two phase flow. *Journal of computational physics* 210, 1 (2005), 225–246.
- [41] OSHER, S., AND SETHIAN, J. A. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics* 79, 1 (1988), 12–49.
- [42] PATHAK, A., AND RAESSI, M. A three-dimensional volume-of-fluid method for reconstructing and advecting three-material interfaces forming contact lines. *Journal of Computational Physics* 307 (2016), 550–573.
- [43] PEI, C., VAHAB, M., SUSSMAN, M., AND HUSSAINI, M. Y. A hierarchical space-time spectral element and moment-of-fluid method for improved capturing of vortical structures in incompressible multi-phase/multi-material flows. *Journal of Scientific Computing* 81, 3 (2019), 1527–1566.
- [44] QIU, R., HUANG, R., XIAO, Y., WANG, J., ZHANG, Z., YUE, J., ZENG, Z., AND WANG, Y. Physics-informed neural networks for phase-field method in two-phase flow. *Physics of Fluids* 34, 5 (05 2022). 052109.
- [45] REMMERSWAAL, R. A., AND VELDMAN, A. E. Parabolic interface reconstruction for 2d volume of fluid methods. *Journal of Computational Physics* 469 (2022), 111473.
- [46] SALAS, M. D. Shock fitting method for complicated two-dimensional supersonic flows. *AIAA Journal* 14, 5 (1976), 583–588.
- [47] SAUREL, R., AND ABGRALL, R. A multiphase godunov method for compressible multifluid and multiphase flows. *Journal of Computational Physics* 150, 2 (1999), 425–467.
- [48] SCHOFIELD, S. P., GARIMELLA, R. V., FRANCOIS, M. M., AND LOUBÈRE, R. Material order-independent interface reconstruction using power diagrams. *International journal for numerical methods in fluids* 56, 6 (2008), 643.
- [49] SCHOFIELD, S. P., GARIMELLA, R. V., FRANCOIS, M. M., AND LOUBÈRE, R. A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations. *Journal of Computational Physics* 228, 3 (2009), 731–745.
- [50] SHETABIVASH, H., DOLATABADI, A., AND PARASCHIVOIU, M. A multiple level-set approach for modelling containerless freezing process. *Journal of Computational Physics* 415 (2020), 109527.
- [51] SHIN, S., AND JURIC, D. A hybrid interface method for three-dimensional multiphase flows based on front tracking and level set techniques. *International Journal for Numerical Methods in Fluids* 60, 7 (2009), 753–778.

- [52] SHU, C.-W., AND OSHER, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of computational physics* 77, 2 (1988), 439–471.
- [53] SIJOY, C., AND CHATURVEDI, S. Volume-of-fluid algorithm with different modified dynamic material ordering methods and their comparisons. *Journal of Computational Physics* 229, 10 (2010), 3848–3863.
- [54] SMITH, K. A., SOLIS, F. J., AND CHOPP, D. A projection method for motion of triple junctions by level sets. *Interfaces and Free Boundaries* 4, 3 (2002), 263–276.
- [55] STARINSHAK, D. P., KARNI, S., AND ROE, P. L. A new level set model for multimaterial flows. *Journal of Computational Physics* 262 (2014), 1–16.
- [56] SUSSMAN, M. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics* 187, 1 (2003), 110–136.
- [57] SUSSMAN, M., AND OHTA, M. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM Journal on Scientific Computing* 31, 4 (2009), 2447–2471.
- [58] SUSSMAN, M., AND PUCKETT, E. G. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *Journal of Computational Physics* 162, 2 (2000), 301–337.
- [59] SUSSMAN, M., SMEREKA, P., AND OSHER, S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics* 114, 1 (1994), 146–159.
- [60] TATEBE, O. The multigrid preconditioned conjugate gradient method. In *NASA. Langley Research Center, The Sixth Copper Mountain Conference on Multigrid Methods, Part 2* (1993).
- [61] UNVERDI, S. O., AND TRYGGVASON, G. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of computational physics* 100, 1 (1992), 25–37.
- [62] VAHAB, M., PEI, C., HUSSAINI, M. Y., SUSSMAN, M., AND LIAN, Y. An adaptive coupled level set and moment-of-fluid method for simulating droplet impact and solidification on solid surfaces with application to aircraft icing. In *54th AIAA Aerospace Sciences Meeting* (2016), p. 1340.
- [63] VAHAB, M., SUSSMAN, M., AND SHOELE, K. Fluid-structure interaction of thin flexible bodies in multi-material multi-phase systems. *Journal of Computational Physics* 429 (2021), 110008.
- [64] VAN LEER, B. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov’s method. *Journal of computational Physics* 32, 1 (1979), 101–136.



- [65] VU, T. V., TRYGGVASON, G., HOMMA, S., AND WELLS, J. C. Numerical investigations of drop solidification on a cold plate in the presence of volume change. *International Journal of Multiphase Flow* 76 (2015), 73–85.
- 750 [66] WELCH, S. W., AND WILSON, J. A volume of fluid based method for fluid flows with phase change. *Journal of computational physics* 160, 2 (2000), 662–682.
- [67] ZALESAK, S. T. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of computational physics* 31, 3 (1979), 335–362.