FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

A HYBRID PARTICLE AND COUPLED MOMENT-OF-FLUID LEVEL SET METHOD FOR MULTIPHASE FLOWS

By

CODY ESTEBE

A Dissertation submitted to the
Department of Mathematics
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2024

Cody Estebe defended this dissertation on April 2, 2024.
The members of the supervisory committee were:

Mark Sussman
Professor Directing Dissertation

Xiaoqiang Wang
University Representative

Feng Bao
Committee Member

Sanghyun Lee
Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the dissertation has been approved in accordance with university requirements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

A new algorithm that hybridizes massless marker particles with the coupled level set moment-of-fluid method to improve the interface reconstruction in multiphase flow problems is proposed. This new method aims to take advantage of both near-interface and on-interface Lagrangian marker particles in a similar fashion to the Particle Level Set (PLS) Method, first introduced by Enright et al. [28]. In contrast to Enright et al. who used particles to improve the mass conservation of the level set method, in this thesis the utility of the particles are studied to improve the slope reconstruction for the multimaterial moment of fluid (MOF) method [54]. Additionally, prior to this, PLS methods have not been implemented on N-phase flow problems involving more than two materials. The proposed method is volume-preserving.

Furthermore, we hypothesize that deriving slopes from particles overcomes the surface tension checkerboard instability of the MOF method first discussed by Ye et al. [116]. Ye et al. developed the "Continuous Moment of Fluid" (CMOF) method in order to overcome this checkerboard instability. It is asserted in this thesis that augmenting the interface representation with meshless particle produces similar results to that of the CMOF method, at a lowered computational cost.

The new coupled particle level set and moment of fluid method (PLSMOF) is demonstrated on benchmark tests in 2d, 3d axisymmetric "R-Z", and 3d problems including rigid body motion, two phase flow, and incompressible multiphase flow problems undergoing phase change involving several different materials with 'triple-points'. Comparisons are made between the new PLSMOF method proposed herein, the CMOF method, and the MOF method for multimaterial multiphase flows.

CHAPTER 1

INTRODUCTION

1.1 Background

Efficient and accurate numerical models for problems of incompressible flow involving multiple phases or materials are important in fields of engineering where simulation of complex phenomena involving phase change between multiple materials in the context of problems such as combustion, cavitation, and icing is needed. Physical phenomena such as surface tension, transfer of energy and momentum, and phase change drive the behavior of these problems and require schemes that can account for such interactions and capture the resulting behavior of the system.

Broadly, multiphase methods are categorized as either 'interface-tracking' or 'interface-capturing' schemes depending on how the front is represented and the problem discretized. Tracking methods involve a Lagrangian representation- either in the form of volume or interface tracking. Volume tracking schemes such as the marker-and-cell (MAC) method [37] fill the fluid volume with Lagrangian marker particles which can then be used to reconstruct an interface. Interface tracking schemes on the other hand, such as Tryggvason et al.'s front-tracking method [104, 46] or Peskin's immersed boundary method [73, 74] use a Lagrangian representation of the interface itself to explicitly track its movement. Another category of interface tracking methods are body-fitted methods. In contrast to the tracking schemes which embed the Lagrangian interface into the grid, body fitted methods such as Arbitrary Lagrangian-Eulerian (ALE) [9, 57] and Finite Element Methods (FEM) [96, 124] involve a fitting of the computational grid to the interface (i.e. the grid is aligned such that the interface coincides with a grid line). Interface tracking schemes more accurately represent the interface than volume tracking as it is directly tracked, but they introduce a significant complexity in dealing with particle connectivity or grid reconstruction under topological changes (such as separating or merging interfaces) the handling of which may introduce non-physical distortions.

In contrast, front-capturing methods use an Eulerian grid-based approach to representing the interface. These schemes do not directly track the interface but rather 'capture' it implicitly. Front-capturing methods automatically handle the topological changes that present difficulties to

front-tracking schemes. Popular approaches to front-capturing schemes include the Volume-of-Fluid method [41], Level Set method [69, 100], Moment-of-Fluid method [27, 26, 53, 65, 62] and phase-field method [108, 111, 49, 44]. Among front-capturing methods, interfaces can be represented in a sharp manner (discussed further below) or diffusely, as is done with the phase-field approach. Sharp interfaces more accurately capture discontinuous parameters at the interfaces between fluids, but the discontinuity requires special treatment in order to not introduce unwanted physical behavior such as spurious oscillations. Diffuse interface representations in which the jump is smoothed at the fluid interface with artificial viscosity avoids this erroneous numerical behavior at the cost of being less representative of the physical interface [63].

A number of hybrid approaches have been proposed to address the shortcomings of the varying schemes such as the Coupled Level Set Volume of Fluid (CLSVOF) [99, 51] or the Particle Level Set (PLS) [29, 110] method which is both front-tracking and front-capturing. The new method proposed within this thesis is in essence a hybridization of hybrid approaches. The methods it builds from are outlined briefly below.

1.1.1 Volume of Fluid Method

The Volume of Fluid Method (VOF) of Hirt and Nichols [41] discretizes the domain by assigning 'volume fractions' to the grid which indicate the percent of fluid within each cell. Volume fractions are advected and redistributed by an upwind donor/acceptor cell transfer prodcedure. By its construction, VOF explicitly conserves volume quantities, though it is susceptible to interface smearing. Instead of the interface being solely represented by the volume fractions of mixed cells, geometric VOF techniques have been introduced prevent numeric diffusion, though at an added complexity cost due the geometric representation of the interface. The Simple Line Interface Calculation (SLIC) [66, 43] approximates the interface with cell-aligned linear segments, positioned so as to capture the representative VOF volume, producing a low-order 'stepped' representation to the interface. The Piecewise-Linear Interface Construction (PLIC) [117, 75] also reconstructs the interface in linear segments, however these piecewise segments are instead oriented to best fit the fluid body. Geometric VOF methods have been popular within CFD with advancements such as Weymouth and Yue's conservative VOF [113] which avoids over/under shoots or truncations of the

volume fractions during advection, Le Chenadec and Pitsch's [52] treatment to handle surface tension in the presence of large density discontinuities without introducing instabilities from the VOF construction, and Fröde et al.'s [33] exactly volume conserving unsplit method using a Lagrangian cell correction.

In the context of multiphase flows, it is necessary to find the curvature of the interface. The most common approach for the curvature calculation in the context of VOF is the height function technique [38, 23, 2]. The curvature is computed along one axis (determined by the interface normal) by the means of height functions, whose 'height' is given within a local stencil by the sum of connected volume fractions for a given cell column. A finite difference approximation is then applied to this representation to obtain an interfacial curvature. Modifications to the original height function technique have been proposed for higher order approximations [98] and to handle underresolved regions [77].

1.1.2 Level Set Method

The level set method [69, 100, 68] implicitly represents interfaces by means of a signed distance function. The interface is defined as the zero level set, immersed into the computational domain, with grid values indicating the signed distance to it. This representation easily handles topological changes as with VOF and the interface curvature is simple to obtain. However, as the level set is advected, it undergoes deformation and does not remain a distance function. Consequently the level set must therefore be reinitialized in order to not introduce instabilities. Typically, the level set is reinitialized after every advection step. Unfortunately, the reinitialization procedure itself can negatively impact the conservion of mass, particularly in regions of high curvature where the interface may be underresolved.

A number of different approaches to the reinitalization of the level set function have been proposed. The PDE-based redistancing approach [100] iterates until the signed distance property (the Eikonal equation $|\nabla\phi| = 1$) is restored. However, this technique negatively impacts the volume conservation, particularly in regions of high curvature. Other level set redistancing methods include the crossing time [1] approach, in which the interface is advected outwardly and the psuedo-timestep in which the interface reaches a cell is taken as the updated value. The fast marching and fast sweeping methods [88, 121] can also be used to reconstruct the interface under the Eikonal equation. Another approach is to use a polynomial approximation to the zero level set and use a quasi newton

method to find the closest point of each cell to the reconstructed interface [19, 81, 78, 39, 22], though additional care is required in these techniques to ensure the closest point lands on the appropriate interface.

In order to address the mass conservation issues of the level set method, it is common to couple it with a volume preserving scheme as was done with the Coupled Level Set Volume-of-Fluid (CLSVOF) method [99] and similarly Coupled Level Set Moment-of-Fluid method [45]. In these couplings, the VOF/MOF handles the interface capturing from which a level set is geometrically redistanced from the VOF/MOF reconstructed interface, the smooth level set function is in turn used to handle the physical parameters.

1.1.3 Moment-of-Fluid Method

The Moment-of-Fluid method, introduced by Dydachko and Shashkov [27, 26], builds upon that of VOF. It is also volume-preserving but doesn't rely on neighboring cells (as PLIC VOF) for its reconstruction procedure. In MOF, both the the volume fraction and the volume centroid information are advected and then used to reconstruct the interface (using a non-linear optimization procedure). The involvement of the volume centroid is motivated by the need to minimize the artificial fluid movement that results from an inexact reconstruction. Lemoine et al. [53, 62] introduced an alternative procedure to the original minimization step for the surface reconstruction (using an analytic reconstruction). For two phases it replaces the original reconstruction procedure, and for more than two materials, it is coupled with the minimization algorithm for speedup. This accelerated method is used for the implemented MOF routines in this paper.

It is important to note that the because the MOF reconstruction procedure is entirely localized within the cell, it is susceptible to 'checkerboard' instabilities when noise is present, as observed in [116], and may not sufficiently capture the damping behavior of surface tension forces.

Of interest within this thesis is the comparison between three multimaterial coupled level set moment-of-fluid methods:

1. The level set MOF in which the original local (MOF) reconstruction remains entirely local within each cell [45, 54]

2. The Continuous Moment-of-Fluid method (CMOF), which uses a supercell containing neighboring cells (thereby removing the checkerboard instability of MOF) to reconstruct the inter-

face (either with a machine-learning derived slope reconstruction procedure or Gauss-Newton optimization) [116].

3. And the new coupled particle level set and moment of fluid method (PLSMOF) described in chapter 3. This proposed method replaces the slope reconstruction procedure used in MOF and CMOF. It uses Lagrangian particles in a neighborhood to likewise address the MOF checkerboard instability.

1.1.4 Particle Hybridization

Mixed tracking/capturing schemes that are coupled in order to overcome the shortcomings of their non-coupled counterparts have become increasingly popular. Of particular interest are methods in which the Lagrangian components are used without any connectivity as was done with Chen et al.'s Surface Marker Micro Cell method [18].

Enright et al. [28, 29] first introduced coupling the level set method with Lagrangian particles in order to improve the problems of mass conservation the level set method faces during reinitialization. The Particle Level Set (PLS) method uses a particle based correction procedure in order to reinitialize the level set function. Particles are seeded within a narrow band on either side of the interface, given a corresponding sign and distance value, and advected independently of the grid-based level set. The escaped interior/exterior particles are then used to correct the level set during the redistancing step and the underlying level set is used to correct trailing particles in regions of merging characteristics. The addition of these particles brings two primary advantages: the particle connectivity of front-tracking schemes is not involved and while not it is not an explicitly conservative scheme, mass is well-maintained due to the added particles.

A number of schemes stemming from the original PLS method have been introduced. Such methods include the One Layer Particle Level Set (OPLS) method [122, 123] and the Marker Level Set (MLS) method [61] which also couple marker particles to the level set method, but instead seed the Lagrangian particles directly on the interface in a single layer, requiring substantially fewer particles than the original PLS. A fully Lagrangian Particle Level Set method was also proposed in [40] which represents the level set in an entirely Lagrangian manner. Particles have also been coupled with VOF to aid with the reconstruction and curvature calculations [47, 115].

Relevant to the proposed method is the curvature adaptive Particle Level Set method [107], which proposes a seeding method that increases particle density in regions where local curvature

is large. And the Multiple Marker Level Set method [8], which introduces the use of particles for multiple materials (two-phase, multiple level sets), where each particle is uniquely associated with a corresponding material level set.

The PLSMOF method proposed herein uses particles seeded both directly on and in a region near the interface. It is implemented within an AMR framework and particles are 'owned' separately by each material. Though the particles themselves are mesh-free, a consequence of the seeding procedure described in section 3.2, grid refinement not only increases the Eulerian grid resolution, but also the particle density.

1.1.5 Multimaterial/Multiphase Triple Point

Special treatments must be used in conjunction with the above methods when applied to multiphase flow problems to account for physical characteristics such as surface tension and momentum conservation in order to avoid introducing instability, oscillations, or other unwanted behavior stemming from the numerical representation. [13, 82, 120, 4] among many others propose treatments for the surface tension and momentum conservation in the context of Level Set and VOF schemes. Treatments for the surface tension include Brackbill et al.'s Continuum Surface Force (CSF) model [13, 32] which uses treats the interface in a continuous manner, and Fedkiw et al.'s Ghost Fluid (GFM) [30] which instead treats the interfaces sharply. Sharp-interface representations are popular due to increased accuracy over the diffuse representations at similar resolutions, with a number of alternative sharp-interface/modified GFM treatments proposed [30, 35, 80, 101, 55, 32]. Some newer approaches such as Physics-Informed Neural Networks (PINN) [79] can instead use training data in order to capture any governing physical laws as an alternative to the modeling approaches.

All three methods compared within this dissertation (level set MOF, CMOF, and PLSMOF) use the same numerical treatments (outlined in chapter 3.3) with regards to the physical governing equations. They are constructed using the sharp-interface treatment of [54, 105] and modified Ghost Fluid Method (for the surface tension force). It is important to note that the compared methods differ only in regards to the interface reconstruction and redistancing processes. The comparison of the different reconstruction treatments is made in chapter 4.

There is a large span of literature which concerns the simulation of multiphase flow, however many of these are for two materials only. In the context of three materials, a new level of complexity is added as 'triple-points' emerge in which multiple interfaces may meet at junctions where three (or

more) materials are present. A comparison of the literature is included in Ye et al. [116] between recent numerical methods that are able to handle these triple points:

For level set methods, there are [91, 93, 89]. For phase-field method, [49, 44, 64]. For Front-tracking, [106]. For VOF methods, [16, 32, 84, 85, 90, 12, 71, 4]. And for MOF, [3, 27, 50, 116].

Of these triple-point approaches, [90, 4, 50, 116] are simultaneously coupled with fluid, volume preserving, and with a sharp interface treatment. The multimaterial MOF reconstruction uses a nested dissection (see Fig. 3.5) to reconstruct the triple point, tessalating the region with linear segments that cut through the uninitialized cell. Because of this, the contact angles within the triple-point containing cell will not exactly match the physical angles, though this does not prevent the simulation from relaxing to the appropriate steady-state (see chapter 4.4). Alternative to the nested dissection triple point reconstruction is the second order power method, as proposed in [86], however it is not straightforward to extend to 3d problems.

1.1.6 Phase Change

In addition to the interfacial motion of multiple interacting phases, we are interested in capturing the phase change occuring between different materials. It is necessary that the phase change be implemented such that the interface remains consistent with the exchange of mass between the materials. Past research of evaporation and boiling in multiphase flows include the Schrage model [87, 94], in which both temperature and pressure are assumed continuous everywhere and heat transfer between liquid/vapor regions is not modeled. Can and Prosperetti's model [17], in which a constant pressure assumption is used for vapor-liquid flows. The velocity and temperature fields in the vapor are ignored to simplify computation and the dynamical effects of the vapor approximated by spatially uniform pressure. And sharp interface phase change models that depend on the mass fractions have been proposed by Son et al.[92], Gibou et al. [34], Villegas et al. [80], Kassemi et al.[48], and Palmore & Desjardins [70] with varying simplifying assumptions. Recent methods using sharp interface phase change models include [112, 58], using VOF, and [34, 102, 80], using the Ghost Fluid approach.

The equations governing phase change used in this paper are implemented as in Ye et al. [116] and Vahab et al. [105]. The liquid materials are considered incompressible and the vapor regions as compressible (though acoustic waves are ignored to avoid the stringent time step constraint, similarly as in [48]). The density of vapor regions is assumed to be spatially uniform (though not

the temperature). The mass flux treated as in [70]. The governing equations involving phase change are descibed in chapter 2.1, and a brief overview of the numerical algorithm for the phase change is given in chapter 3.3.

1.1.7 Adaptive Mesh Refinement

For the Eulerian grid, Adaptive Mesh Refinement (AMR) can be used to refine the grids selectively around features requiring greater resolution or a smaller timestep in order to avoid the expense of a global refinement. The two primary approaches for AMR on structured, logically rectangular grids are either Quad/Octree or Block-Structured AMR. Unstructured mesh AMR also exists, though they are typically used in the context of body-fitted methods.

The Quadtree/Octree method [60, 31] subdivides individual grid cells into 4/8 blocks and stores them hierarchically in a tree-structure (available codes include Basilisk(Gerris) [76] and RAMSES [103]). This method of refinement is advantageous in terms of locality though not amenable to parallelization. Block-structured AMR, on the other hand, constructs patches in which a block of grid cells is refined and data stored in overlapping layers. This structure lends itself well to parallelization (at the cost of increased grid storage). Commonly used block-structured codes include AMReX (formerly BoxLib) [119], Chombo [21], and SAMRAI [114]. ForestClaw [15] is an alternate approach that uses non-overlapping grids in an octree manner, using a block-structured approach for parallelization.

The code used for the proposed method is implemented within the AMReX framework, the evaluation of the grid parameters parallelized on the CPU using MPI and OpenMP. AMReX also features particle support through its 'particle container' routines which handles communication between particles and grids. Particles inhabit the finest grid they overlap with neighboring (particle copies from other grids at same level) and ghost (copies at coarse/fine interfaces) particle information available in buffer regions for each grid.

1.2 Scope

In this dissertation, we present a novel method for reconstructing the slope in the coupled level set moment of fluid. This method augments the interface with independently advected Lagrangian particles (each uniquely assigned to a corresponding material) to be used as an alternative to the

MOF reconstruction which involves a nonlinear optimization procedure. This method is novel both for the alternative MOF slope reconstruction and for the use of PLS in the context of N-phase simulations involving more than two phases.

We explore whether the introduction of particles to MOF is advantageous over the original level set MOF and alternative CMOF methods. That is, if it can avoid spurious oscillations due to the checkerboard instability of the MOF interface construction (demonstrated in [116]) while maintaining volume preservation. Results for this particle level set moment of fluid (PLSMOF) method are directly compared to both MOF and CMOF on benchmark passive advection tests and multimaterial phase change tests in terms of accuracy and performance. This new method is demonstrated to work on multiphase problems undergoing phase change and involving triple-points.

CHAPTER 2

MATHEMATICAL MODEL

The multiphase system is comprised of M_{fluid} deforming materials and M_{rigid} non-deforming materials. The fluid materials tessellate the entire computational domain (extended through any rigid bodies), and where a material m_{fluid} intersects with a solid m_{rigid} region, the governing equations for the rigid body is used. In regions not containing a m_{rigid}, the incompressible Navier-Stokes equations of immiscible flows govern the fluid materials.

2.1 Governing Equations

2.1.1 Multimaterial Level Sets

We define the rigid and fluid materials using the following multimaterial level set formulation:

$$\phi_{m_{\text{rigid}}}(\boldsymbol{x},t) = \begin{cases} > 0 & \boldsymbol{x} \in \text{material } m_{\text{rigid}}, \\ \leq 0 & \text{otherwise}, \end{cases} \tag{2.1}$$

$$\phi_{m_{\text{fluid}}}(\boldsymbol{x},t) = \begin{cases} > 0 & \boldsymbol{x} \in \text{material } m_{\text{fluid}} \cup m_{\text{fluid,ghost}}, \\ \leq 0 & \text{otherwise}. \end{cases} \tag{2.2}$$

For position vector $\boldsymbol{x}$ and time t. Material m_{rigid} is defined as the region where $\phi_{m_{\text{rigid}}} > 0$ and the domain of material m_{fluid} as the region where $\phi_{m_{\text{fluid}}} > 0$ and $\phi_{m_{\text{rigid}}} < 0$. m_{ghost} indicates the region in which the fluid level set is extended through the rigid body. This extension is implemented as in [5] in which the fluid interface is extended orthogonally to the rigid body, regardless of contact angle. This configuration can be seen in figure 2.1 below.

The interface level set, $\phi_{m1,m2}$, represents the interface between materials m_1 and m_2

$$\phi_{m_1,m_2}(\boldsymbol{x},t) = \begin{cases} > 0 & \boldsymbol{x} \in \text{material } m_1, \\ < 0 & \boldsymbol{x} \in \text{material } m_2, \\ = 0 & \text{for } \boldsymbol{x} \text{ along } (m_1, m_2) \text{ interface}. \end{cases} \tag{2.3}$$

The interfacial level sets are not stored nor extended through the material level sets as in [116]. Where needed, we instead obtain the interfacial level sets from the involved material level sets as

$$\phi_{m_1,m_2} = \frac{\phi_{m_1} - \phi_{m_2}}{2}. \tag{2.4}$$

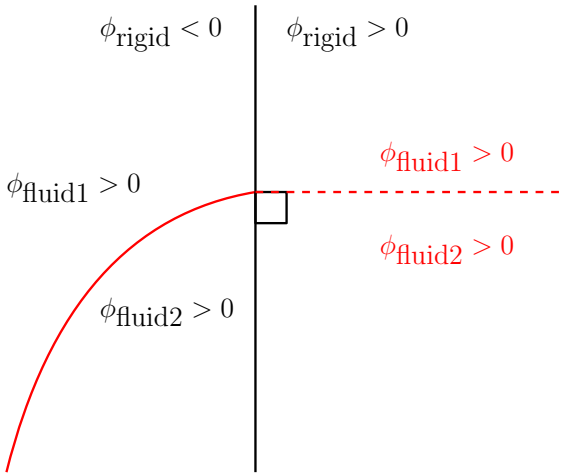


Figure 2.1: fluid level set extension through rigid region

The associated normal $\boldsymbol{n}$ and curvature κ for the level set functions are defined as

$$\boldsymbol{n}_{m_1,m_2} = \frac{\nabla \phi_{m_1,m_2}}{||\nabla \phi_{m_1,m_2}||}, \quad \kappa_{m_1,m_2} = \nabla \cdot \frac{\nabla \phi_{m_1,m_2}}{||\nabla \phi_{m_1,m_2}||}. \tag{2.5}$$

2.1.2 Conservation of mass

Each fluid material, m_{fluid}, is assumed to be incompressible, so that the velocity field $\boldsymbol{u}$ is divergence free within each fluid material:

$$\nabla \cdot \boldsymbol{u} = 0. \tag{2.6}$$

The following conditions on $\nabla \cdot \boldsymbol{u}$ are enforced to account for phase change and any mass sources or sinks:

$$\nabla \cdot \boldsymbol{u} = \sum_{\text{sources}} \frac{\dot{m}_{\text{source}}}{\rho_{\text{source}}} \delta(\phi_{m_{\text{source}}}) - \sum_{\text{sinks}} \frac{\dot{m}_{\text{sink}}}{\rho_{\text{sink}}} \delta(\phi_{m_{\text{sink}}}) \tag{2.7}$$

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

$$\delta(\phi) = H'(\phi) \tag{2.8}$$

and $H(\phi)$ is the Heaviside function,

$$H(\phi) = \begin{cases} 1 & \phi > 0 \\ 0 & \phi \leq 0. \end{cases} \tag{2.9}$$

$\dot{m}$ is the mass flux. For the mass flux of boiling liquid across the liquid/vapor interface, we have

$$\dot{m} = \frac{k_l \nabla T_l \cdot \boldsymbol{n}_{l,v} - k_v \nabla T_v \cdot \boldsymbol{n}_{l,v}}{L}, \tag{2.10}$$

where k_l and k_v are the respective thermal conductivities of the liquid and ambient vapor regions. ρ_l and ρ_v are the respective densities of the liquid and ambient vapor regions. L is the latent heat of vaporization. Vector $\boldsymbol{n}_{l,v}$ is the interface normal that points from the vapor region into the liquid,

$$\boldsymbol{n}_{l,v} = \frac{\nabla \phi_{l,v}}{||\nabla \phi_{l,v}||}. \tag{2.11}$$

2.1.3 Conservation of momentum:

For each material in its domain $\phi_m(\boldsymbol{x}, t) > 0$ we have the following conservation of momentum:

$$(\boldsymbol{u}\rho_m)_t + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}\rho_m + p_m \mathbb{I}) = \nabla \cdot (2\mu_m \mathbb{D}) + \rho_m \boldsymbol{g}(1 - \alpha_m(T_m - T_{0m})). \tag{2.12}$$

with density ρ_m, pressure p_m, temperature T_m, coefficients of thermal expansion α_m, and viscosities μ_m for material m. $\boldsymbol{g}$ is the acceleration due to gravity and $\mathbb{D} = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)$ is the rate of deformation tensor.

2.1.4 Conservation of energy:

For each material in its domain $\phi_m(\boldsymbol{x}, t) > 0$ we have the conservation of energy equation:

$$(\rho_m C_{p,m} T_m)_t + \nabla \cdot (\boldsymbol{u}\rho_m C_{p,m} T_m) = \nabla \cdot (k_m \nabla T_m). \tag{2.13}$$

$C_{p,m}$ is the heat capacity, k_m is the thermal conductivity, and T_m is the temperature corresponding to material m.

2.1.5 Interfacial jump conditions:

We represent a deforming interface undergoing phase change with the following general equations.

$$\phi_{m_s,m_d} + \boldsymbol{u}_{m_s} \cdot \nabla \phi_{m_s,m_d} = -\frac{\dot{m}}{\rho_s} ||\nabla \phi_{m_s,m_d}|| \tag{2.14}$$

$$\phi_{m_d,m_s} + \boldsymbol{u}_{m_s} \cdot \nabla \phi_{m_d,m_s} = \frac{\dot{m}}{\rho_s} ||\nabla \phi_{m_d,m_s}|| \tag{2.15}$$

Or, equivalently:

$$\phi_{m_s,m_d} + \boldsymbol{u}_{m_d} \cdot \nabla \phi_{m_s,m_d} = -\frac{\dot{m}}{\rho_d} ||\nabla \phi_{m_s,m_d}|| \tag{2.16}$$

$$\phi_{m_d,m_s} + \boldsymbol{u}_{m_d} \cdot \nabla \phi_{m_d,m_s} = \frac{\dot{m}}{\rho_d} ||\nabla \phi_{m_d,m_s}|| \tag{2.17}$$

For the level set ϕ_{m_s,m_d} indicating the interface separating a material m_s region from a material m_d region. m_s is defined as the material id associated with a 'source' material and m_d is the corresponding 'destination' material (e.g. for boiling, m_s is the liquid material and m_d the vapor region and for freezing, m_s would indicate the liquid material and m_d the icing material).

For the velocity, pressure, and temperature interface jump conditions between materials m_s and m_d, we apply the following definitions:

$$\boldsymbol{u}_{m_s} \cdot \boldsymbol{n}_{m_s,m_d} - \boldsymbol{u}_{m_d} \cdot \boldsymbol{n}_{m_s,m_d} = \dot{m} \left(\frac{1}{\rho_{m_d}} - \frac{1}{\rho_{m_s}} \right), \tag{2.18}$$

$$\begin{aligned} (p_{m_s}\mathbb{I} - p_{m_d}\mathbb{I}) \cdot \boldsymbol{n}_{m_s,m_d} &= -\sigma_{m_s,m_d}\kappa_{m_s,m_d}\boldsymbol{n}_{m_s,m_d} \\ &\quad + (2\mu_{m_s}\mathbb{D}_{m_s} - 2\mu_{m_d}\mathbb{D}_{m_d}) \cdot \boldsymbol{n}_{m_s,m_d}, \end{aligned} \tag{2.19}$$

$$T_{m_s} = T_{m_d}. \tag{2.20}$$

Where σ_{m_s,m_d} is the prescribed surface tension and κ_{m_s,m_d} is the interface curvature, defined as

$$\kappa_{m_s,m_d} = \nabla \cdot \frac{\nabla \phi_{m1,m2}}{||\nabla \phi_{m1,m2}||}. \tag{2.21}$$

2.1.6 Triple point junction

At a triple point, a three-phase equilibrium (Neumann's triangle [24]) determines the contact angles for a steady state, dependent on the surface tension of the phases in contact. (see Figure 3.3(a)):

$$\frac{\sin(\theta_1)}{\sigma_{23}} = \frac{\sin(\theta_2)}{\sigma_{13}} = \frac{\sin(\theta_3)}{\sigma_{12}}. \tag{2.22}$$

In the figures below, the multimaterial level set constructions for some of the numerical experiments in Chapter 4 are shown. The fluid level sets $\phi_{m_{\text{fluid}}}$ are tessellated such that they occupy the entirety of the computational domain, extended through the rigid bodies.

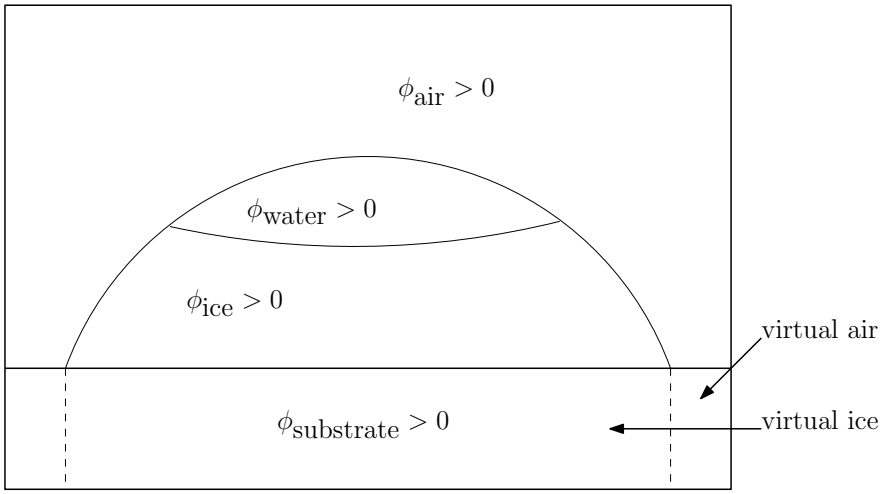


Figure 2.2: Initial level set construction for the problem of a water droplet freezing ontop of a cold substrate. $\phi_{\text{substrate}} > 0$ in the cold substrate, $\phi_{\text{ice}} > 0$ in the ice, $\phi_{\text{water}} > 0$ in the water, and $\phi_{\text{air}} > 0$ in the rest of the domain. In this problem, only the substrate is considered a region material and is prescribed a temperature T_w. The fluid/deforming materials (ice, water, and air) tessellate the full computational domain. The orthogonal extension of the air and ice materials through the rigid substrate is indicated by the dashed lines. The ice material is handled as a deforming material rather than rigid since phase change occurs at the ice/water interface. As in the freezing model presented in [58], the ice and water are considered to be the same material during the surface tension force calculation at the triple-point where the ice, water, and air meet. That is, the interfacial curvature (both at and away from the triple point) is found using the VOF height function method [23, 97].

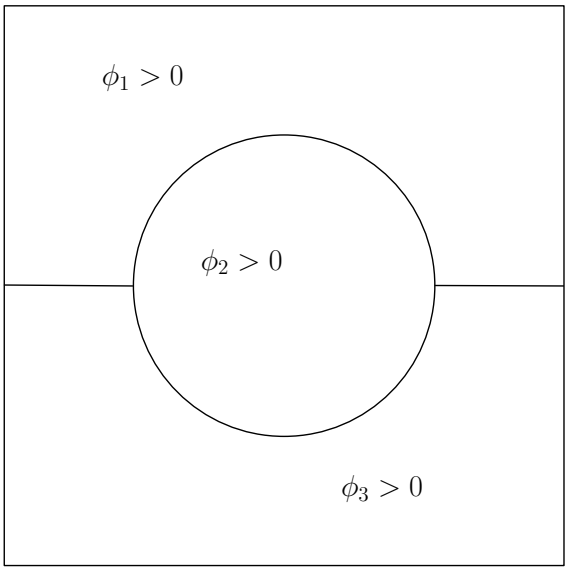


Figure 2.3: Initial level set construction for the simulation of a stretching 'liquid lens' due to surface tension. $\phi_2 > 0$ represents the lens material. ϕ_1, ϕ_2 and ϕ_3 are all deforming fluid materials. The surface tension model described in [54], for 'Stencil contains third fluid' is used to calculate the surface tension forces at the triple points. This surface tension force, $\sum_{m=1}^{M} \gamma_m \kappa_m \nabla H(\phi_m)$, is treated using a modified ghost-fluid approach [30, 35, 101, 55] in which the curvature κ_m at the triple-point is approximated using finite differences of the level set functions ϕ. Away from triple points, the interfacial curvature is obtained by the VOF height function method [97].

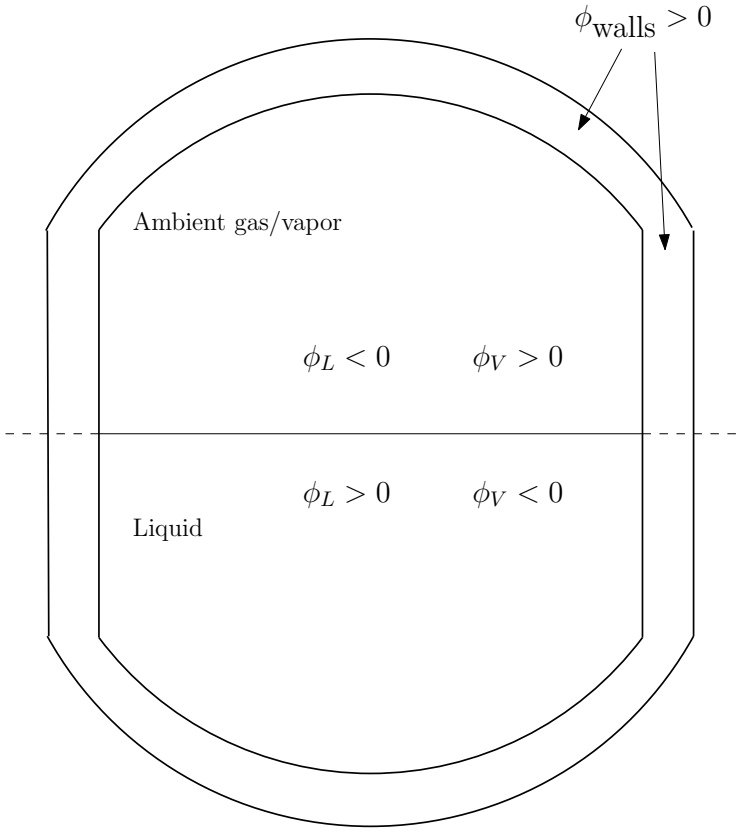


Figure 2.4: Initial level set functions for simulating evaporation within a cryogenic fuel tank. $\phi_{\text{walls}} > 0$ represents the rigid body of the tank which is embedded in the fluid materials. The upper vapor region $\phi_V > 0$ and the lower liquid region $\phi_L > 0$ are both extended through the tank walls to cover the full domain. The extended interface is indicated by the dashed lines. The surface tension model from [54] for 'Stencil contains rigid boundary' is used. The curvature at the (fluid, fluid, solid) triple-point is approximated with $\kappa = \nabla \cdot \boldsymbol{n}$ in which $\boldsymbol{n}$ is a strategically assigned 'ghost normal' in the solid region, consistent with the contact angle condition. Away from the triple points, the interfacial curvature is computed with the VOF height function [97].

CHAPTER 3

NUMERICAL METHOD

The computational domain is discretized onto a uniform rectangular MAC grid. Level sets, volume fractions, centroids, temperature, and pressure are stored at cell centers, while velocities are stored at the cell faces corresponding to their respective direction (i.e. in 2d, horizontal velocities u are stored at midpoint of cell faces with constant x value, and likewise for velocity components v, w in their respective directions) [36]. The computational cell $\Omega_{i,j,k}$ is defined

$$\Omega_{i,j,k} = \left\{ \boldsymbol{x} : x \in \left[x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2} \right], \;\; y \in \left[y_j - \frac{\Delta y}{2}, y_j + \frac{\Delta y}{2} \right], \;\; z \in \left[z_k - \frac{\Delta z}{2}, z_k + \frac{\Delta z}{2} \right] \right\} \tag{3.1}$$

with corresponding cell center $\boldsymbol{x}_{i,j,k} = \{x_i, y_j, z_k\}$.

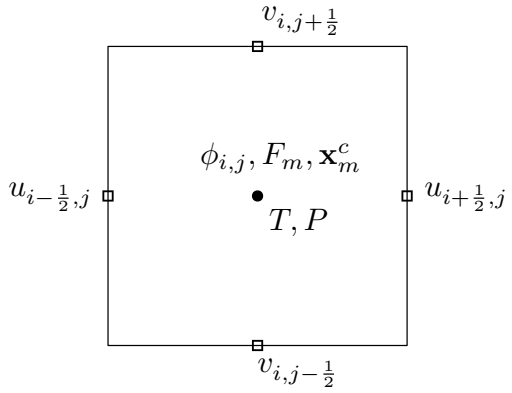


Figure 3.1: 2d cell $\Omega_{i,j}$

Below, the implementation of the Moment of Fluid reconstruction is outlined, followed by the particle reconstruction procedure. In the proposed method, particles are used for the slope reconstruction procedure wherever available- if no particles are available then the MOF routine is used. When particles are used, the centroid information is not needed for the reconstruction (though they are still involved during phase change), and can be considered a hybrid particle CLSVOF procedure in a sense, though the MOF method takes over in many situations. Typical

situations where the method may fall back to the MOF reconstruction include when breakup occurs (if the region is under-resolved) or at coarse/fine interface boundaries on the AMR grid. Particles are located on the finest grid corresponding to their position. Consequently, in regions on coarser grids that are overlapped by finer levels will also not have any particles available, therefore MOF reconstruction is used instead. Due to the particle seeding procedure, particles may not be seeded in regions of breakup if the level set does not change sign between cells. The interface in this under-resolved region may still be captured by the centroids and the MOF reconstruction used. For the AMR grid boundaries between coarse and fine, MOF is used- this is motivated out of convenience rather than necessity as it removes the complexity that arises from communicating between levels.

3.1 Moment-of-Fluid Reconstruction

The zeroth and first order moments (corresponding to the volume fraction F and centroid position $\boldsymbol{x}^c_m$) of material m are defined

$$F^n_{m,(i,j,k)} = \frac{\int_{\Omega^n_{m,(i,j,k)}} \mathrm{d}\Omega}{V_{i,j,k}} \tag{3.2}$$

$$\boldsymbol{x}^{c,n}_{m,(i,j,k)} = \frac{\int_{\Omega^n_{m,(i,j,k)}} \boldsymbol{x}\ \mathrm{d}\Omega}{V^n_{m,(i,j,k)}}, \tag{3.3}$$

where the domain of material m in a cell at time t^n is denoted by $\Omega^n_{m,(i,j,k)}$. The computational cell volume $V_{i,j,k} = \int_{\Omega_{i,j,k}} \mathrm{d}\Omega$, and the volume of the material portion m is $V^n_{m,(i,j,k)} = \int_{\Omega^n_{m,(i,j,k)}} \mathrm{d}\Omega$. Figure 3.3 demonstrates a discretization about a triple point.

3.1.1 MOF Interface Reconstruction

It is important to note that the MOF reconstruction is a local procedure, using only the reference volume fraction F_{ref} and reference centroid $\boldsymbol{x}^c_{\text{ref}}$ to obtain the linear/planar interface reconstruction. The interface update uses the advected volume fraction and centroid as the initial reference for the reconstruction (section 3.3 outlines the advection procedure)

$$F_{\text{ref}} \equiv F^n_{m,(ijk)}, \qquad \boldsymbol{x}^c_{\text{ref}} \equiv \boldsymbol{x}^n_{m,(ijk)} \tag{3.4}$$

We consider the linear interface reconstruction of the form $\boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_{i,j}) + b = 0$ within the evaluated cell (see Figure 3.4). The reconstruction's volume fraction $F_{\text{act}}(\boldsymbol{n}, b)$ is exactly equal to

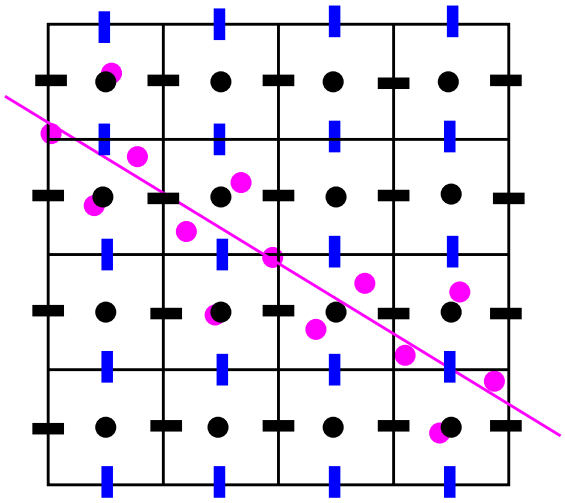


Figure 3.2: Computational grid: Level sets, volume fractions, centroids, and state variables stored at cell centers (black circles). The pink line represents the interface reconstruction and the centroids for the cut cell volume fractions is indicated by the pink circles. The dashes at the cell faces are the locations the velocity is discretized: horizontal velocities u indicated in black, and vertical velocities v indicated in blue.

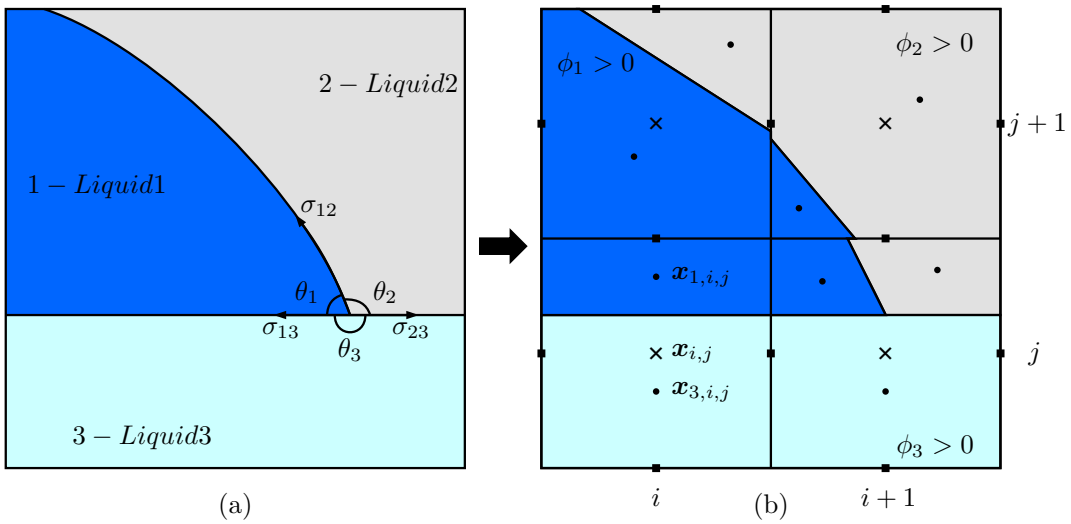


Figure 3.3: (a) Physical Domain, shown here for a triple point. The contact angles θ are determined at equilibrium by the surface tension forces σ.
(b) Discretized Domain. $\times$: cell centers, $\bullet$: cell centroids, $\blacksquare$: MAC velocities

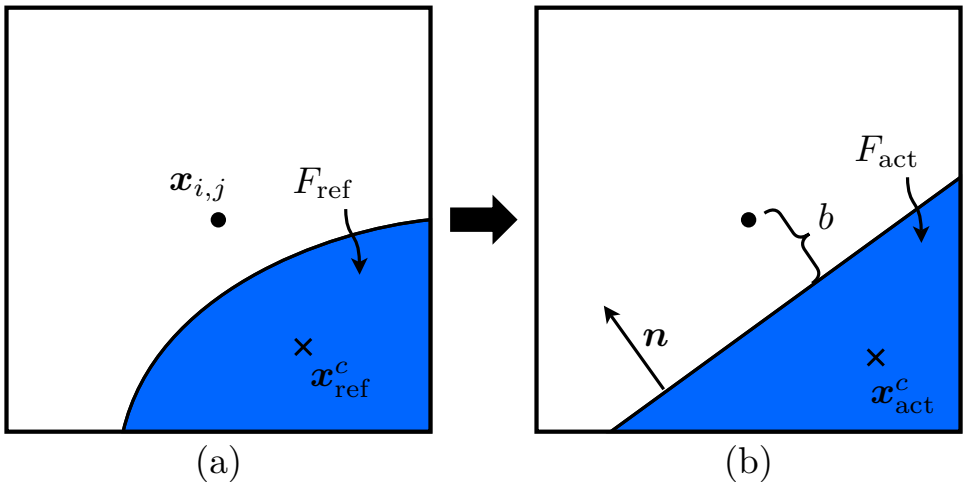


Figure 3.4: (a) Material domain in a cell $\Omega_{i,j}$, a single phase shown in blue corresponding with reference volume F_{ref} and centroid $\boldsymbol{x}^c_{\text{ref}}$.
(b) The piecewise linear MOF reconstruction. The line segment for the reconstructed volume can be represented as $\Omega_{i,j} \cap \{\boldsymbol{x} | \boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_{i,j}) + b = 0\}$.

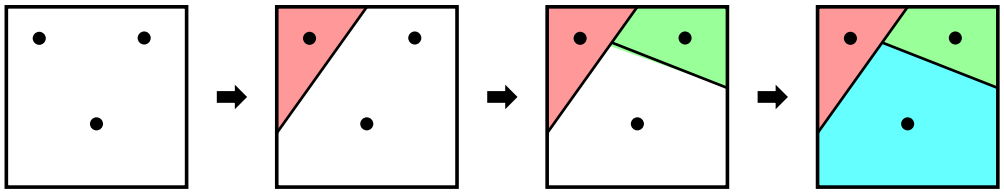


Figure 3.5: MOF reconstruction, volume-tessellation (nested dissection) procedure. Points indicate centroids, white space is the unoccupied region, and zones of color indicate reconstructed materials.

that of the reference, and the location for the reconstructed centroid $\boldsymbol{x}^c_{\text{act}}(\boldsymbol{n}, b)$ is found such that its error with respect to $\boldsymbol{x}^c_{\text{ref}}$ is minimized using the cost function

$$E_{\text{MOF}} = \|\boldsymbol{x}^c_{\text{ref}} - \boldsymbol{x}^c_{\text{act}}(\boldsymbol{n}, b)\|_2 \tag{3.5}$$

with the associated constraint

$$F_{\text{act}}(\boldsymbol{n}, b) = F_{\text{ref}}. \tag{3.6}$$

The method used to obtain the minimizing normal $\boldsymbol{n}$ and intercept b in this nonlinear optimization procedure is outlined in [45], in which E_{MOF} is parameterized in terms of the angles of $\boldsymbol{n}$ and solved using the Gauss-Newton algorithm. Once the slope has been found that minimizes equation 3.5, a hybrid bisection/Newton method is then used to find the associated intercept b that satisfies equation 3.6. Alternatively, the intercept can be obtained analytically from the slope using Lemoine et al's method [53, 62] which speeds up the reconstruction process. This procedure iterates until the optimal slope and intercept are obtained.

$$\boldsymbol{n} = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \text{ in 2d, and } \boldsymbol{n} = \begin{pmatrix} \sin\phi\cos\theta \\ \sin\phi\sin\theta \\ \cos\phi \end{pmatrix} \text{ in 3d.} \tag{3.7}$$

This procedure begins with a whole cell, but each subsequent material reconstructed reduces the unoccupied space in the cell. Only the unoccupied region may be considered when reconstructing remaining materials. The detailed procedure is described in [54] and the reconstruction process visualized in Figure 3.5. The resulting tessellation produces a volume preserving reconstruction for both two-phase and triple-point cells.

3.1.2 Redistancing

From the reconstructed interface, the method outlined below is used to redistance the material (ϕ_m) level set functions with the exact signed distance to the reconstruction within a narrow band of the interface.

- ▶ Initialize the distance functions to a large negative value.
- ▶ Iterate over the entire computational domain.
  - ▷ If a material k is at the cell center, update the sign of the associated level set ϕ_k.
  - ▷ If $F_m > VOLFRAC$, then increment the local material count m.

- ▷ If $F_m < VOLFRAC$, iterate in a $3 \times 3(\times 3)$ neighborhood about the current cell.
  - If $F_m > 1 - VOLFRAC$ within at least one cell in this neighborhood, then increment the local material count m.
- ▷ If more than one material is found within this cell, it is tagged as a 'support' cell.
- ▷ If a support cell is identified, iterate in a $9 \times 9(\times 9)$ neighborhood for which level set values will be redistanced. For each update cell in this region:
  - Find distances from the update cell center to the support cell center, corner, and faces (Figure 3.6(a)).
  - Find the normal distances between the update cell center and each interface within the support cell (Figure 3.6(b)). This distance is rejected if the intersection point is outside of the queried support cell.
  - Find the distances between the update cell center and each triple point within the support cell (Figure 3.6(c)).
  - Find the distances between the update cell center and the interface intersection with the cell faces (Figure 3.6(d)).
  - (3d) Find the distances between the update cell center and each interface plane and line where the plane intersects the cell walls (Figure 3.7).

In short, the distances are evaluated between the center of an update cell to tentative interface points within support cells, located in a neighborhood about the cell evaluated. If the point queried is valid, ϕ_m is updated to reflect the measurement.

As the above query distances are found, probe points about the intersection are used to identify the appropriate materials involved in order to update the respective interface level sets. The probe points are projected inside the different materials using the information of the involved interface normals. The level set values and grid normal vector are iteratively updated as they are found to correspond to the smallest distance measured (for both material and interface functions).
Note: The lower threshold for the volume fraction $VOLFRAC$ within a cell is used to catch support cells that may otherwise not be registered due to the interface coinciding with the grid. If the volume fraction is below the threshold, a second check is performed for neighboring cells to determine if there is a neighbor that has a different material with a $F_m > 1 - VOLFRAC$. If so, it triggers an update. In our experiments we let the minimum volume fraction $VOLFRAC = 10^{-3}$.

Note: This level set redistancing procedure produces an exact distance function to the MOF/CMOF/PLSMOF reconstructed interfaces. However, the interface captured by the zero level set will not directly match that of the reconstructed interface (e.g. Figures 4.28 and 4.29). The redistancing procedure does not affect the mass conservation of the method as the volume fractions are maintained (and advected by volume-preserving CISL(section 3.3.1)). The underlying level-set can however affect the centroid positions of PLSMOF as the level set is used during the PLSMOF reconstruction procedure (section 3.2). Both the level set interface and the centroid positions are used during the phase change procedures. Additionally note: the level set interface does not pass through flotsam (under-resolved volume fractions that migrate away from the interface)- though flotsam is involved during the redistancing step, the sign is determined by the cell center and therefore will not impact the level set interface as it will not change sign for under-resolved flotsam.

Remark: The level set redistancing scheme from the MOF reconstructed interface as outlined above is quite expensive due to the required width of the narrow band that needs to be reinitialized. Rather than updating the entire $9 \times 9(\times 9)$ proposed above, we instead implement a stencil reduction to limit the cells processed about each critical point containing an interface.

3.1.3 Update Stencil Reduction

If there is a level set function that is reasonably defined in a interface-containing cell, then we find a low-order approximate slope from that cell. A tube is drawn in the direction of the slope normal. Cells within the stencil not contained by this tube are removed.

This reduction is motivated by Mauch's algorithm [59] for computing the closest point to a piecewise linear level set function in which the distance function is reconstructed first in regions extending normal to the interface segment before regions extending from the corners are evaluated. In our case, we update only what extends normal to the linear segment. The corner regions near the interface are only updated if near enough to the interface to be covered by the tube thickness. Figures 3.8 and 3.10 demonstrate this procedure.

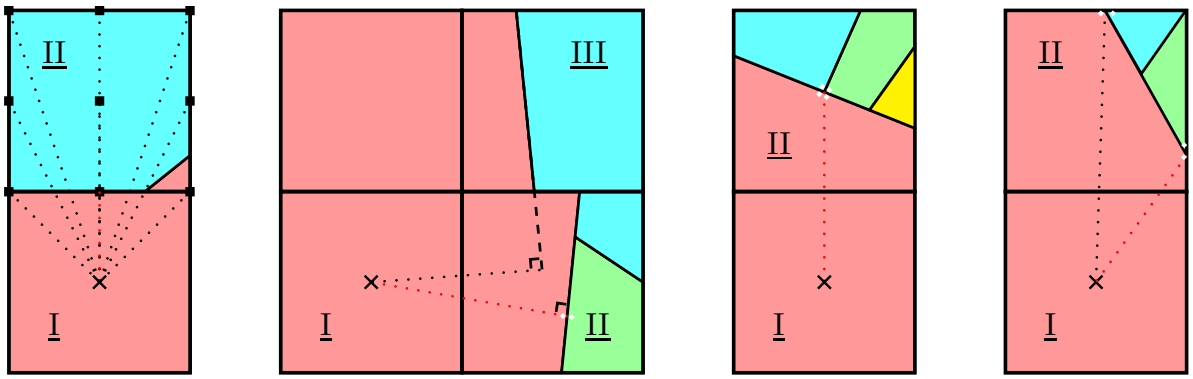


Figure 3.6: MOF redistancing procedure (shown for candidate cell-center $\times$ in update cell I).

(a) Candidate distances (black dotted lines) are measured from the center of update cell (I) to the center, corners, and faces of the support cell (II). In this configuration, the nearest sample point coinciding with a different material is indicated by the red dotted line. If the material interface coincides with the cell boundary, this gives the exact distance.

(b) The distance between update cell (I) using the reconstructed normal for the interface in support cell (II) is shown by the red dotted line. Since the intersection point is contained within the support cell, the two material ids are obtained using test points (white dots) on both sides. In this example, the signed distance and corresponding normal vector for the red material level set, green material level set, and red-green interface level sets are updated. Note: The candidate normal distance with support cell (III) is shown by the black dotted line. Since the intersection point is not contained within the support cell, this distance is not accepted.

(c) The distance between update cell (I) and the material intersections in support cell (II) is measured (red dotted line). Test points (white dots)around the triple point are used to find the material ids. In this example, the red, blue, and green material level sets are updated and the red-blue, red-green, and blue-green interface level sets are updated. Similarly for the red-green-yellow triple point.

(d) Distances between update cell (I) and the cell face intersections in support cell(II) is shown. Test points and level set updates follow as in (c). In this example, red and blue material level sets and blue-red interface level sets are updated (black dotted line) and similarly for red and green.

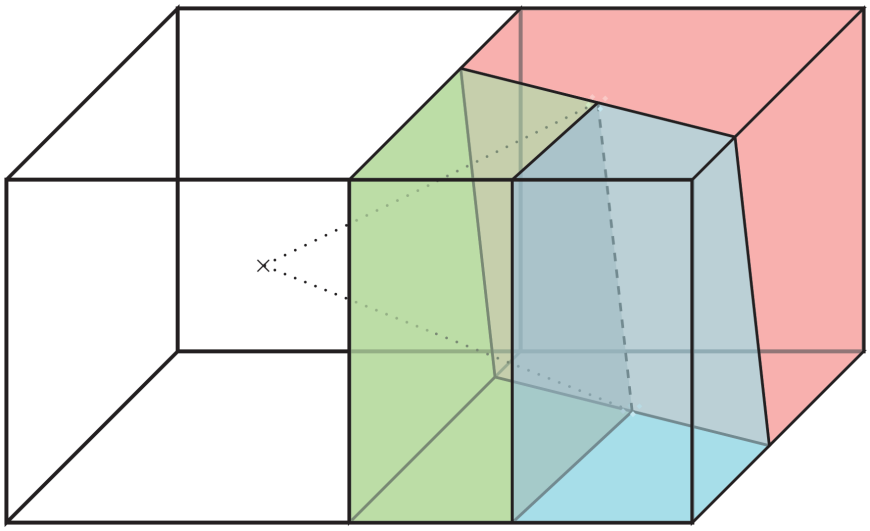


Figure 3.7: 3D MOF redistancing. Update cell (left), querying normal distances to potential nearest interface plane intersection (black dashed line) within support cell (right) as well as distances to the line formed by the intersection of the interface plane with the cell boundaries.

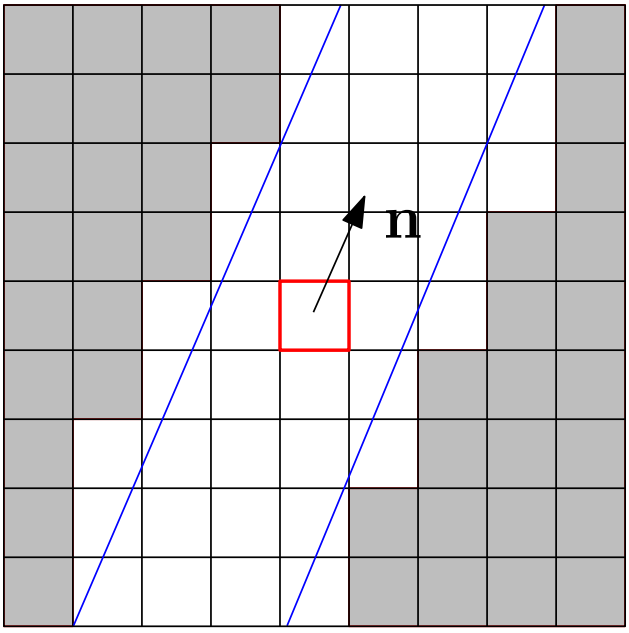


Figure 3.8: The reduced neighborhood stencil. The red cell indicates a support cell about which we construct a $9 \times 9(\times 9)$ update neighborhood. The blue lines indicate the reduced band (3cells thick), aligned in the direction of the interface normal. The grey regions indicate the parts of the $9 \times 9(\times 9)$ stencil that lie outside the narrow band reduction that are not evaluated. If the normal is oriented with the grid, the neighborhood stencil is reduced to a 3x9(x3) region.

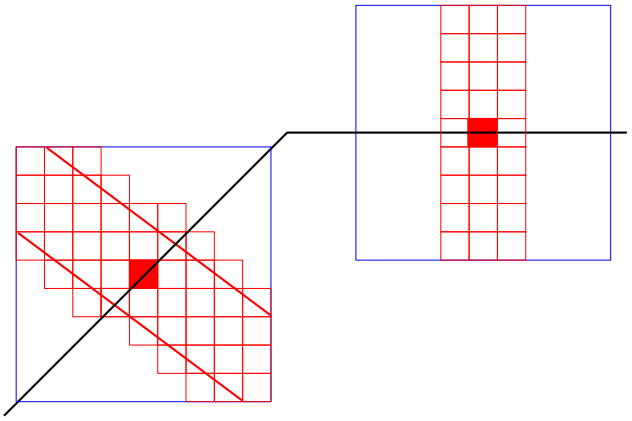


Figure 3.9: Interface stencil regions. The reduced stencil in Fig. 3.8 is aligned in the direction of the interface normal. Shown here in red for two cells on a piecewise linear interface (black lines). By this construction, the cells directly normal to the piecewise linear interface are guaranteed to be updated within the full width of the narrow band about the interface.
It is important to note however, as a consequence of this stencil reduction, there may be cells within the narrow band that are not updated (if they do not extend directly off the interface and the stencils do not overlap). This is demonstrated in the figure below.

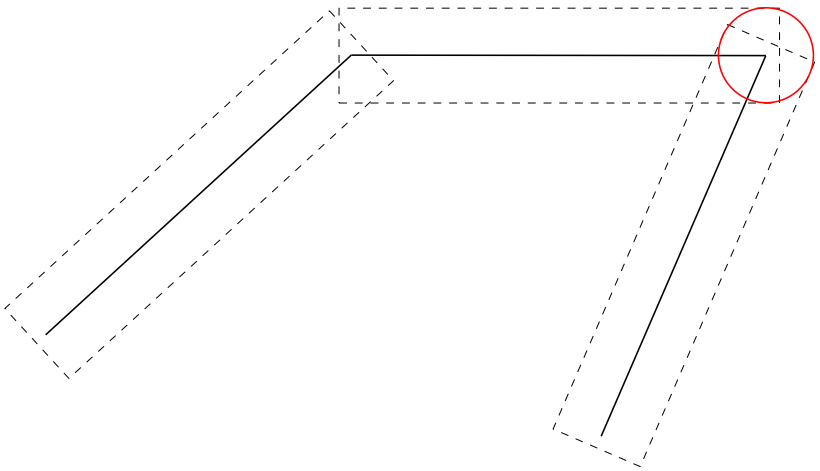


Figure 3.10: The maximum extent of the stencil-covered region of each piecewise linear interface segment is indicated by the dashed lines. At the end of the linear segments, the update region does not extend the full thickness of the narrow band due to the orientation of the stencils. This may produce regions for which cells are within the narrow band but not updated during the redistancing step. This can seen at the junctions of the linear segments (a small gap off the left junction and a larger one on the right, the red circle is centered at a junction with the radius indicating the width of the narrow band). Directly near the end of the linear segment, cells are still guaranteed to be updated due to the stencil thickness, but further away, regions may form in which cells do not get redistanced. Since the cells in the gap region that lie farther from the interface are not updated, the error from these may pollute the level set function and therefore eventually the slopes and volume fractions. However, in practice the error introduced is found to be negligible and the speedup from the stencil reduction significant. In addition to remaining sufficiently accurate, this reduction does not affect the linearity preserving property of the reconstruction (the method is exact for piecewise linear interfaces).

3.2 The Particles

We use marker particles both on the interface and in a narrow band about the interface. These particles are advected independently and are used in lieu of the MOF reconstruction procedure where they are available. The level sets are then redistanced using this particle interface reconstruction in the same manner outlined in the previous section.

3.2.1 Particle Interface Reconstruction

The particles are used during the slope reconstruction step during redistancing as follows:
The linear reconstructed distance function is given by the form

$$\hat{\phi} = \boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_0) + \hat{b} \tag{3.8}$$

Where $\boldsymbol{n}$ and $\hat{b}$ are found such that they minimize the following cost function:

$$C(\boldsymbol{n}, \hat{b}) = \sum_{p \in \text{stencil}} w_p (\hat{\phi}(\boldsymbol{n}, \hat{b}) - \phi_p)^2 + \sum_{i,j,k \in \text{stencil}} w_{ijk} (\hat{\phi}(\boldsymbol{n}, \hat{b}) - \phi_{i,j,k})^2 \tag{3.9}$$

Where 'stencil' is defined as a $3 \times 3(\times 3)$ stencil about the evaluated cell Ω_0. The weights, $w_{i,j,k}$ and w_p are inversely proportional to the distance from the center point, defined here as

$$w_{ijk} = wx_{ijk} * wy_{ijk} * wz_{ijk} * \delta_{\text{smooth}}(\phi_{ijk}) \tag{3.10}$$

$$w_p = w^{\text{dim}} * \delta_{\text{smooth}}(\phi_p) \tag{3.11}$$

where

$$\begin{cases} wx_{ijk} = \begin{cases} 12, & i = 0 \\ 1, & i = 1 \text{ or } -1 \end{cases} \\ wy_{ijk} = \begin{cases} 12, & j = 0 \\ 1, & j = 1 \text{ or } -1 \end{cases} \\ wz_{ijk} = \begin{cases} 12, & k = 0 \\ 1, & k = 1 \text{ or } -1 \end{cases} \end{cases} \tag{3.12}$$

$$w = \begin{cases} 12, & d < \Delta x \\ 1, & d \geq \Delta x \end{cases} \tag{3.13}$$

$d = ||x_p - x_0||$ is the distance between the particle and the stencil center. The weights were constructed as quadrature weights over the neighborhood stencil. ϕ_{ijk} corresponds to the cell-centered level set value and ϕ_p is the biquadratic interpolated level set value at the particle's position.

1	12	1
12	12^2	12
1	12	1

Figure 3.11: $w_{i,j}$ stencil in 2d, from equation 3.10

We define the smooth delta function δ_{smooth} as

$$\delta_{\text{smooth}} = H'_{\text{smooth}} = \begin{cases} 0, & \phi \geq \varepsilon \text{ or } \phi \leq -\varepsilon \\ \frac{1}{2\varepsilon}\left(1 + \cos\left(\frac{\pi\phi}{\varepsilon}\right)\right), & \text{otherwise} \end{cases} \tag{3.14}$$

corresponding to the smooth Heaviside function defined as

$$H_{\text{smooth}} = \begin{cases} 1, & \phi \geq \varepsilon \\ 0, & \phi \leq -\varepsilon \\ \frac{\phi}{2\varepsilon} + \frac{\sin(\frac{\pi\phi}{\varepsilon})}{2\pi} + \frac{1}{2}, & \text{otherwise} \end{cases} \tag{3.15}$$

using $\varepsilon = \sqrt{\dim}\,\Delta x$ (the transition thickness corresponds to the diagonal of the cell).

$C(\boldsymbol{n}, \hat{b})$ is minimized using a least squares approach, finding the the optimal slope $\boldsymbol{n}$ and corresponding intercept $\hat{b}$ for the involved particles. After the optimal parameters have been found, we replace $\hat{b}$ by b in order to agree with the existing volume reconstruction. This b is found by a hybrid bisection/Newton method, such that the volume fraction is satisfied for the new slope. The second term in Eq. 3.9 is a regularization term using the underlying grid data.

Remark: If there are no particles or the resulting matrix is singular, then the MOF reconstruction is used.

3.2.2 Particle initialization and deletion procedures

The procedure to add or delete particles is as follows:

For each computational cell $\Omega_{i,j,k}$,

- ▶ If $|\phi_m| > \Delta x$ for materials $1 \leq m \leq M$ then do not subdivide the cell, i.e. $\Delta x_{\text{sub}} = \Delta x$. Otherwise, the cell is subdivided into N^{dim} pieces, where $\Delta x_{\text{sub}} = \frac{\Delta x}{N}$.
- ▶ For each subdivision (or otherwise non-divided cell):
  - ▷ If the subdivision contains no particles:
    - Interpolate ϕ_m and $\boldsymbol{x}_m^{\text{cp}}$ to $\boldsymbol{x}_{\text{sub}}$.
    - Let $m^* = \text{argmax}\, \phi_m$.
    - If $\phi_{m^*} > \Delta x$ then the particle is rejected.
    - If $\phi_{m^*} \leq \Delta x$, and $\boldsymbol{x}_{m^*}^{\text{cp}} \in \Omega_{\text{sub}}$,
      - ∘ then add particle with position $\boldsymbol{x}_{m^*}^{\text{cp}}$, material id m^*, initialized with value $\phi_{m^*,p} = 0$ (on-interface particle).
    - If $\phi_{m^*} \leq \Delta x$, and $\boldsymbol{x}_{m^*}^{\text{cp}} \notin \Omega_{\text{sub}}$,
      - ∘ then add particle with position $\boldsymbol{x}_{\text{sub}}$, material id m^*, initialized with value $\phi_{m^*,p} = \phi_{m^*}(\boldsymbol{x}_{\text{sub}})$ (near-interface particle).
  - ▷ If the subdivision contains particles, then for each particle in the subdivision:
    - If $\phi_p = 0$ and $|\phi_{mp}| > \Delta x_{\text{sub}}$, then delete the particle.
    - If $\phi_p > 0$ and $\phi_{mp} < 0$ then delete the particle.
    - If $|\phi_{mp}| > \Delta x$ then delete the particle.
    - If $\phi_p > 0$ and $0 \leq \phi_{m,p} \leq \Delta x_{\text{sub}}$ then $\phi_p = \phi_{m,p}$.
    - If the number of particles within a subcell n_{ptc} exceeds specified parameter n^{delete}, then delete the extra particles, beginning with the closest neighbors.

Where m_p is the particle material id, ϕ_p is the level set value of particle p, and $\phi_{m,p}(x_p)$ is interpolated level set value to the particle's position. x^{cp} is the closest point on the interface to x_{sub} ($x^{\text{cp}} = x_{\text{sub}} - \phi_{m,p}(x_p)\boldsymbol{n}_{m,p}$).

To summarize, cells containing the interface are subdivided uniformly. If the subdivision doesn't already contain a particle, we look to seed a new one- either directly on the interface if the subcell contains it, or at the subdivision center with appropriate level set value. If the subdivision contains a particle, we check if a deletion is necessary- if a particle is trailing (can occur when characteristics

merge), or moved into a different material it is culled, otherwise the particle's corresponding level set is updated to correspond to the reinitialized function.

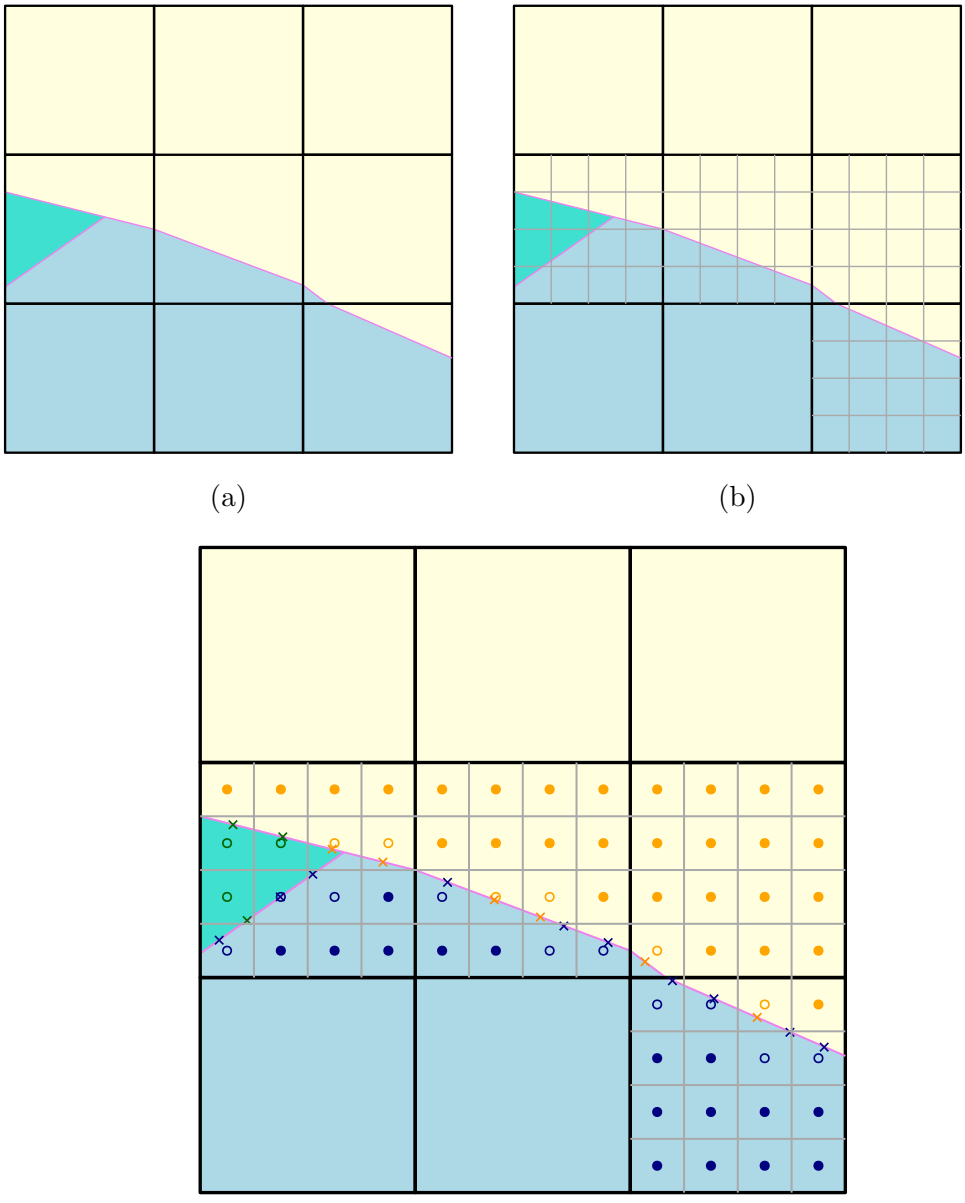


Figure 3.12: (a) initial cell, grid lines: Δx
(b) subdividing cell, $N = 4$, grid lines: Δx_{sub}
(c) Particle Seeding. $\bullet$: near-interface particles, $\times$: on-interface particles, $\bigcirc$: cell-centers for particle sent to interface, color indicates associated material

3.3 Staggered grid projection

Outline of the operator-split staggered grid projection method:

1. Find the time step $\Delta t = t^{n+1} - t^n$:

$$\begin{cases} \Delta t_1 &= \min\limits_{d=1,\ldots,\dim} \dfrac{\min \Delta x_d}{2(dim) \max ||u_d^n||} \\ \Delta t_2 &= \min\limits_{d=1,\ldots,\dim} \dfrac{\min \Delta x_d}{(dim) \max ||u_d^{\text{phase change},n-1}||} \\ \Delta t_3 &= \min\limits_{d=1,\ldots,\dim} \min\limits_{m_1,m_2=1,\ldots,M} \Delta x_d^{3/2} \sqrt{\dfrac{\rho_{m_1} + \rho_{m_2}}{2\pi\sigma_{m_1,m_2}}} \end{cases} \tag{3.16}$$

$$\Delta t = \text{CFL} \min(\Delta t_1, \Delta t_2, \Delta t_3) \tag{3.17}$$

where 'dim' is the dimension of the problem (2 or 3). We let the coefficient 'CFL' $= \frac{1}{2}$. The time step is restricted by the CFL conditions on the MAC velocity, the rate of phase change, and surface tension.

3.3.1 Cell Integrated Semi-Lagrangian

The following Cell Integrated Semi-Lagrangian (CISL) procedures are implemented and terms discretized as in [72, 54, 105]. This method follows an operator-splitting approach. The steps listed below are solved by the conservative Weymouth and Yue [113], or alternatively, the Euler Implicit-Lagrange Explicit (EI-LE) [83, 7] directionally-split methods.

The idea behind the CISL integration is to trace the characteristics backwards of a cell being updated using in an unconditionally stable Semi-Lagrangian manner [95, 109]. From this a 'departure' region is constructed and mapped back to the cell and the containing volume fraction and centroid reconstructed for the updated time. For the mapping function and full CISL algorithm, the reader is referred to [54]. The CISL-MOF reconstruction is visualized in Fig. 3.13 for a single direction. The discretizations used for the CISL algorithm are provided below.

2. Directionally split CISL-MOF, advection of the volume fractions and centroids
For materials $m = 1, \ldots, M$,

$$F_{m,t} + \nabla \cdot (\boldsymbol{u} F_m) = 0 \tag{3.18}$$

$$\boldsymbol{x}_{m,t}^c = \boldsymbol{u}(\boldsymbol{x}_m^c) \tag{3.19}$$

$$F^{n+1}_{m,(i,j,k)} = \frac{\int_{\Omega^{Depart}_{i,j,k}} \chi^n_m(\boldsymbol{x})d\boldsymbol{x}}{||\Omega^{\text{Depart}}_{i,j,k}||} \tag{3.20}$$

$$\boldsymbol{x}^{c,n+1}_{m,(i,j,k)} = \frac{\int_{\mathcal{T}_{\text{CISL}}(\Omega^{\text{Depart}}_{m,(i,j,k)})} \boldsymbol{x}\chi^n_m(\mathcal{T}^{-1}_{\text{CISL}}(\boldsymbol{x}))d\boldsymbol{x}}{||\mathcal{T}_{\text{CISL}}(\Omega^{Depart}_{m,(i,j,k)})||} \tag{3.21}$$

Where χ_m is defined as a characteristic function such that $\chi_m(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in \Omega_{m,(i,j,k)}$, 0 otherwise. $\mathcal{T}_{\text{CISL}}$ represents the mapping function. Refer to Figure 3.13 for integration over the region of the backwards-traced characteristics.

3. CISL level set advection

$$\phi_{m,t} + \boldsymbol{u} \cdot \nabla\phi_m = 0, \quad m = 1, \ldots, M \tag{3.22}$$

$$\phi^{n+1}_{m,(i,j,k)} = \frac{\int_{\Omega^{\text{Depart}}_{m,(i,j,k)}} \phi^n_m(\boldsymbol{x})d\boldsymbol{x}}{||\Omega^{\text{Depart}}_{i,j,k}||} \tag{3.23}$$

4. CISL advection of the MAC velocity

$$(\rho\boldsymbol{u})_t + \nabla \cdot (\rho\boldsymbol{u}\boldsymbol{u}) = 0, \tag{3.24}$$

$$u^{n+1}_{i-1/2,j,k} = \frac{\int_{\Omega^{\text{Depart}}_{i-1/2,j,k}} \sum_{m=1}^{M} \rho_m\chi^n_m(\boldsymbol{x})u^n(\boldsymbol{x})d\boldsymbol{x}}{\int_{\Omega^{\text{Depart}}_{i-1/2,j,k}} \sum_{m=1}^{M} \rho_m\chi^n_m(\boldsymbol{x})d\boldsymbol{x}} \tag{3.25}$$

$$v^{n+1}_{i,j-1/2,k} = \frac{\int_{\Omega^{\text{Depart}}_{i,j-1/2,k}} \sum_{m=1}^{M} \rho_m\chi^n_m(\boldsymbol{x})v^n(\boldsymbol{x})d\boldsymbol{x}}{\int_{\Omega^{\text{Depart}}_{i,j-1/2,k}} \sum_{m=1}^{M} \rho_m\chi^n_m(\boldsymbol{x})d\boldsymbol{x}} \tag{3.26}$$

$$w^{n+1}_{i,j,k-1/2} = \frac{\int_{\Omega^{\text{Depart}}_{i,j,k-1/2}} \sum_{m=1}^{M} \rho_m\chi^n_m(\boldsymbol{x})w^n(\boldsymbol{x})d\boldsymbol{x}}{\int_{\Omega^{\text{Depart}}_{i,j,k-1/2}} \sum_{m=1}^{M} \rho_m\chi^n_m(\boldsymbol{x})d\boldsymbol{x}} \tag{3.27}$$

5. CISL temperature advection (liquid and gas materials)

$$(\rho_m C_{p,m} T_m)_t + \nabla \cdot (\rho_m C_{p,m}\boldsymbol{u}T_m) = 0, \quad m = 1, \ldots, M \tag{3.28}$$

6. Directionally split particle advection (2nd order Runge-Kutta)

$$\boldsymbol{x}_{p,t} = \boldsymbol{u}(\boldsymbol{x}_p) \tag{3.29}$$

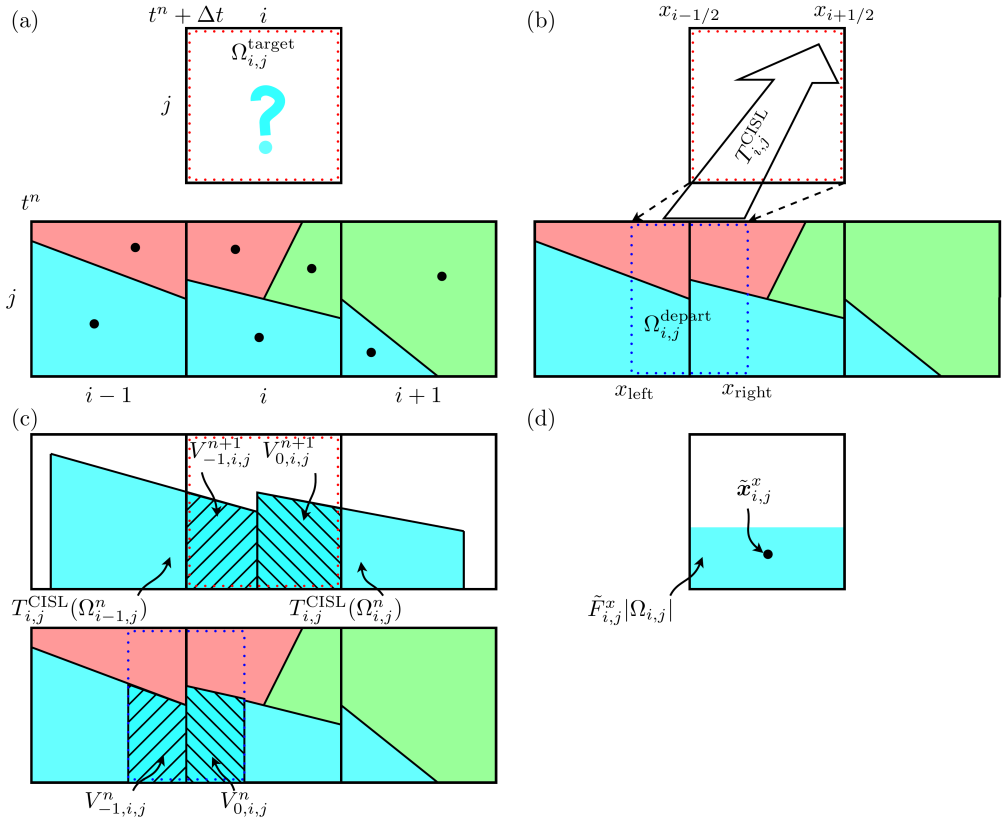


Figure 3.13: CISL-MOF method [figure from [72]], shown here for x-direction for the blue material.
(a) Reconstructed MOF interface and centroids at departure time t^n for slab of cells [{i-1, i, i+1},j] and target cell $\Omega^{\text{target}}_{i,j}$ at arrival time $t^n + \Delta t$.
(b) Characteristics are traced backwards and CISL mapping function T^{CISL} calculated.
(c) The CISL mapping function sends the material from the departure time to the update time and its intersection with the target cell is found.
(d) The contained region is combined to obtain the volume fraction and centroid

3.3.2 Phase Change

To ensure mass conservation, it is necessary that the motion of the interface remain consistent with the mass flux across the interface. The relevant steps are outlined below, implemented from [105].

1. Redistribution of mass source (the mass flux $\dot{m}$ is redistributed form source material m_s to destination material m_d): Eq. 2.10. This is implemented as follows

- ▶ Loop through the grid, for every cell that has a valid closest point map, find if it changes phase at the corresponding closest point.
- ▶ If it does, then $\dot{m}$ is found:
- ▶ $\dot{m} = \left[\frac{k\nabla T \cdot \boldsymbol{n}}{L}\right]$

2. Phase change velocity (from source material m_s to destination material m_d).

$$\boldsymbol{u}^{\text{phase change}} = -\frac{\dot{m}}{\rho_{m_d}}\boldsymbol{n}_{m_d}, \left(\text{ or } \boldsymbol{u}^{\text{phase change}} = \frac{\dot{m}}{\rho_{m_s}}\boldsymbol{n}_{m_s} \right) \tag{3.30}$$

3. Phase change, level set unsplit advection

$$\phi_{m,t} + \boldsymbol{u}^{\text{phase change}} \cdot \nabla\phi_m = 0, \qquad m = m_s \text{ or } m_d. \tag{3.31}$$

4. Phase change: unsplit CISL advection

$$F_{m,t} + \nabla \cdot (\boldsymbol{u}^{\text{phase change}} F_m) = 0, \qquad m = m_s \text{ or } m_d. \tag{3.32}$$

$$\boldsymbol{x}^c_{m,t} = \boldsymbol{u}^{\text{phase change}}(\boldsymbol{x}^c_m) \tag{3.33}$$

5. Unsplit particle advection for phase change (1st order Forward Euler)

$$\frac{d\boldsymbol{x}_p}{dt} = \boldsymbol{u}^{\text{phase change}}(\boldsymbol{x}_p) \tag{3.34}$$

6. Thermal diffusion

$$\frac{(\rho C_{p,m})^{\text{mix},n+1}}{\Delta t^{\text{swept}}}(T_m^{n+1} - T_{\text{sat},m}) = \nabla \cdot (k_m \nabla T_m^{n+1}) \tag{3.35}$$

This large sparse matrix system is solved using the multigrid preconditioned conjugate gradient (MGPCG) method. (See figure 3.14 for the temperature gradient). Note: The temperature is stored at corresponding material centroids; as the interface shifts during phase change, a crossing time approach is used to interpolate the temperature from the old time to new centroids.[56, 105]

7. Viscosity

$$\frac{\rho^{\text{mix},n+1}}{\Delta t}(\boldsymbol{u}^* - \boldsymbol{u}^{\text{advection}}) = \nabla \cdot (2\mu\mathbb{D}^*) - \rho^{n+1}(\alpha^{n+1}(T^{n+1} - T_0))\boldsymbol{g} \tag{3.36}$$

An implicit solver is used for the viscous term $\nabla \cdot (2\mu\mathbb{D}^*)$. In this case, BiCGSTAB using a Jacobi preconditioner and red-black Gauss-Seidel smoother.

8. Pressure projection The pressure projection step is formulated using Chorin's method [20]. From the momentum equation, we have

$$\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^{*}}{\Delta t}=-\frac{\nabla p^{n+1}}{\rho^{\mathrm{MAC,mix},n+1}}+\boldsymbol{g}-\frac{\sum_{m=1}^{M}\gamma_m\kappa_m\nabla H(\phi_m)}{\rho^{\mathrm{MAC,mix},n+1}} \tag{3.37}$$

On the right hand side are the viscous force, gravitational force, and surface tension force terms, respectively. Note: the surface tension force term is handled in a ghost fluid manner [30, 101]. Remark: This equation exhibits a very strong discontinuity at the triple point (in which both ∇H and κ_m behave as delta functions, proportional to $1/\Delta x$, this is further exacerbated once the divergence is taken (Eq. 3.38) and the whole term gains a large error proportional to $1/\Delta x^3$. However, similarly as in [14] (for an error proportional to $1/\Delta x$), it is found that since this error is localized just at the triple point, it hasn't been observed to affect the behavior away from the triple point. In the numerical experiments, the κ_m discontinuity converges under grid refinement (future convergence studies are warranted).

Taking the divergence of both sides,

$$\frac{1}{\Delta t}\left(\nabla\cdot\boldsymbol{u}^{n+1}-\nabla\cdot\boldsymbol{u}^{*}\right)=\nabla\cdot\left[-\frac{\nabla p^{n+1}}{\rho^{\mathrm{MAC,mix},n+1}}+\boldsymbol{g}-\frac{\sum_{m=1}^{M}\gamma_m\kappa_m\nabla H(\phi_m)}{\rho^{\mathrm{MAC,mix},n+1}}\right] \tag{3.38}$$

From equation 2.7, we enforce the expansion term

$$\nabla\cdot\boldsymbol{u}^{n+1}=\sum_{\text{sources}}\frac{\dot{m}_{\text{source}}}{\rho_{\text{source}}}\delta(\phi_{m_{\text{source}}})-\sum_{\text{sinks}}\frac{\dot{m}_{\text{sink}}}{\rho_{\text{sink}}}\delta(\phi_{m_{\text{sink}}}) \tag{3.39}$$

and use the following γ_m:

$$\gamma_{m_1}=\gamma_{m_2}=\frac{1}{2}\sigma_{m_1,m_2} \tag{3.40}$$

for two materials $m1$, $m2$ in a $3\times3(\times3)$ local stencil. If three materials $m1$, $m2$, $m3$, are present in the stencil, then the following terms are used instead (satisfying Eq. 2.22):

$$\gamma_{m_1}=\frac{1}{2}(\sigma_{m_1,m_2}+\sigma_{m_1,m_3}-\sigma_{m_2,m_3}) \tag{3.41}$$

$$\gamma_{m_2}=\frac{1}{2}(\sigma_{m_1,m_2}+\sigma_{m_2,m_3}-\sigma_{m_1,m_3}) \tag{3.42}$$

$$\gamma_{m_3}=\frac{1}{2}(\sigma_{m_1,m_3}+\sigma_{m_2,m_3}-\sigma_{m_1,m_2}) \tag{3.43}$$

Equation 3.38 can then be solved for the pressure term. BiCGSTAB using a multigrid preconditioner and red-black Gauss-Seidel smoother is used.
The velocity for the next timestep is then

$$\boldsymbol{u}^{n+1}=\boldsymbol{u}^{*}-\Delta t\frac{\nabla p^{n+1}}{\rho^{\mathrm{MAC,mix},n+1}} \tag{3.44}$$

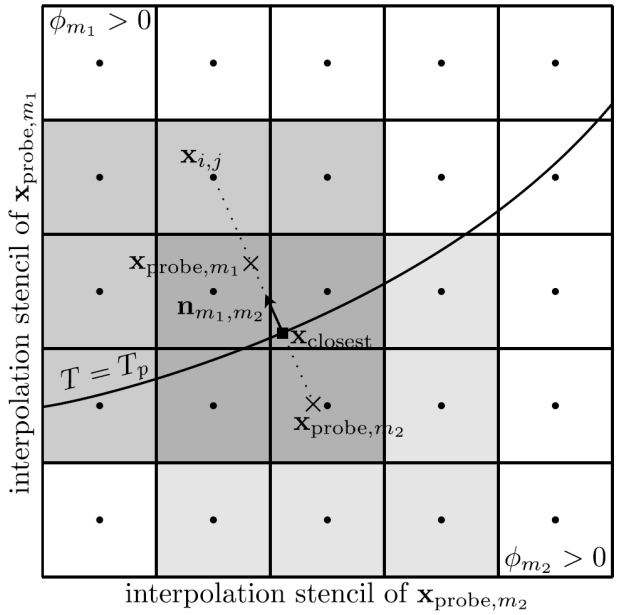


Figure 3.14: [figure from [105]]
The closest point is found on the interface by $x_{\text{closest}} = x_{i,j} - \phi_{m_1,m_2}\boldsymbol{n}_{m_1,m_2}$,
two probe points are extended in either normal direction and $T(x_{\text{probe}})$ is found by interpolation.
$\nabla T_{m_1} = (T_{\text{probe},m_1} - T_{\text{sat},m_1,m_2})\boldsymbol{n}_{m_1,m_2}/h$ and $\nabla T_{m_2} = -(T_{\text{probe},m_2} - T_{\text{sat},m_1,m_2})\boldsymbol{n}_{m_1,m_2}/h$, where T_{sat} is the phase change saturation temperature.

3.4 Algorithm

The involved steps for the PLSMOF method described prior are implemented thus:

1. Initialize (for each material) a level set function, VOF function, centroids, velocity (on the cell center grid), density, and temperature at cell centers.

   The level set function is replaced with the exact signed distance to the multimaterial MOF reconstructed interface as described in section 3.1.

   Seed initial particles using the level set interface given by initial MOF reconstruction. And apply a redistancing step for the level set using the particles. (This has a smoothing effect, removing any noise from the MOF inital reconstruction).

   A divergence free MAC velocity is then derived from the cell centered velocity:
   ($t = 0$ only), we solve

   $$\boldsymbol{u}^{\text{MAC}} = \boldsymbol{u}^{\text{cell\_to\_MAC}} - \frac{\nabla p}{\rho},$$

   $$\nabla \cdot \boldsymbol{u}^{\text{MAC}} = 0$$

2. For each timestep (as described in section 3.3),

   (a) Directionally split advection of volume fractions, centroids, level set functions, MAC velocity, cell velocity, temperature, density, particles. Note: the slope reconstruction (either MOF or particle-augmented) is performed before every CISL sweep.

   (b) Redistance the level set, add and delete particles.

   (c) Determine $\dot{m}$, T_{sat} for the rate of phase change (refer to Fig. 3.14).

   (d) Update the interface using unsplit advection (volume fractions, centroids, level set functions, particles). Slope reconstruction.

   (e) Redistance the level set, add and delete particles.

   (f) Determine the source terms for $\nabla \cdot \boldsymbol{u}$ equation.

   (g) Determine temperature diffusion (thermal conductivity) and viscosity (velocity diffusion).

   (h) Projection step.

   (i) AMR grid synchronization and cycle.

CHAPTER 4

NUMERICAL EXPERIMENTS

In the experiments below, PLSMOF is tested on rigid body benchmarks as well as multiphase flow problems involving triple points and phase change. It is compared against both the level set MOF and CMOF methods. Unless otherwise specified, PLSMOF parameters are taken to be $N = 4$ (number of subdivisions) and $n^{\text{delete}} = 2$ (limit of particles per subcell). It is important to note that for the reconstructed interface, the 'flotsam' does not necessarily coincide with the zero level set- a flag in the algorithm determines whether flotsam is kept or truncated. For the experiments below, we do not truncate the volume fractions and the flotsam is kept. Additionally, wherever the reconstructed interface is visualized, the visual for the linear/planar reconstructing segments may extend past the what is actually used (e.g. at triple points, the segments of the tessellated region cutting through each other) - this is a visualization artifact only.

4.1 Zalesak's Disk

In this section, the PLSMOF method is tested on the 2d rigid body rotation test of 'Zalesak's disk' [118]. We then compare results with MOF and CMOF. The problem domain is defined on $[0, 100] \times [0, 100]$. The initial rigid body is a circular disk is centered at (50,75), with a radius of 15. A slot is cut out from the bottom of the disk with width 5 and height 25. And the prescribed velocity field is given by:

$$\begin{cases} u &= -(\pi/314)(y - 50) \\ v &= (\pi/314)(x - 50) \end{cases}$$

Table 4.1 shows a comparison of symmetric difference error

$$E_{\text{sym}} = |\Omega^{\text{approx}} \cap \Omega^{\text{exact,complement}}| + |\Omega^{\text{approx,complement}} \cap \Omega^{\text{exact}}|, \tag{4.1}$$

of the MOF, CMOF, and PLSMOF methods after one full rotation ($t = 628.0$) of the disk.

Table 4.1: Comparison of symmetric difference error (E_{sym}) at $t = 628$.
'GN' indicates the Gauss-Newton optimization slope reconstruction
'N' indicates the number of subdivisions used in the particle procedure.
Figures 4.1 and 4.2 show a comparison between PLSMOF and MOF/CMOF after one full rotation of the disk.

Δx	Δt	MOF GN	CMOF GN	PLSMOF $N = 4$
$\frac{100}{96}$	$\frac{628.0}{1155}$	6.8	19.0	16.8
$\frac{100}{192}$	$\frac{628.0}{2066}$	2.5	7.3	6.48

Table 4.2: Cost comparison of combined centroid error minimization and intercept calculation
Reconstruction cost: (average number of iterations (GN), reconstruction time).
Note: the PLSMOF reconstruction only involves the intercept calculation.

Δx	Δt	MOF-GN	CMOF-GN	PLSMOF
100/96	628.0/1155	3.6, 0.010	7.3, 0.05	0, 0.009
100/192	628.0/2066	3.7, 0.024	6.7, 0.09	0, 0.019

Table 4.3: Grid refinement study, PLSMOF, $t = 628$.
'Coarse' and 'fine' correspond the two grid resolutions used in the previous tables in which the AMR adapts about the interface (interface is fully contained at the appropriate finest level). 'Curvature' indicates a curvature-based criterion in which the AMR adapts around regions of high curvature instead. The finest level of the 'curvature' case is the same as the 'fine' case. Figure 4.3 shows the mesh adaptation and final interface after one full rotation of the 'curvature' case.

refinement:	coarse	fine	curvature
E_{sym}	16.8	6.48	10.34
# particles	5,431	11,295	8,119
# grid cells	3,840	10,240	5,600

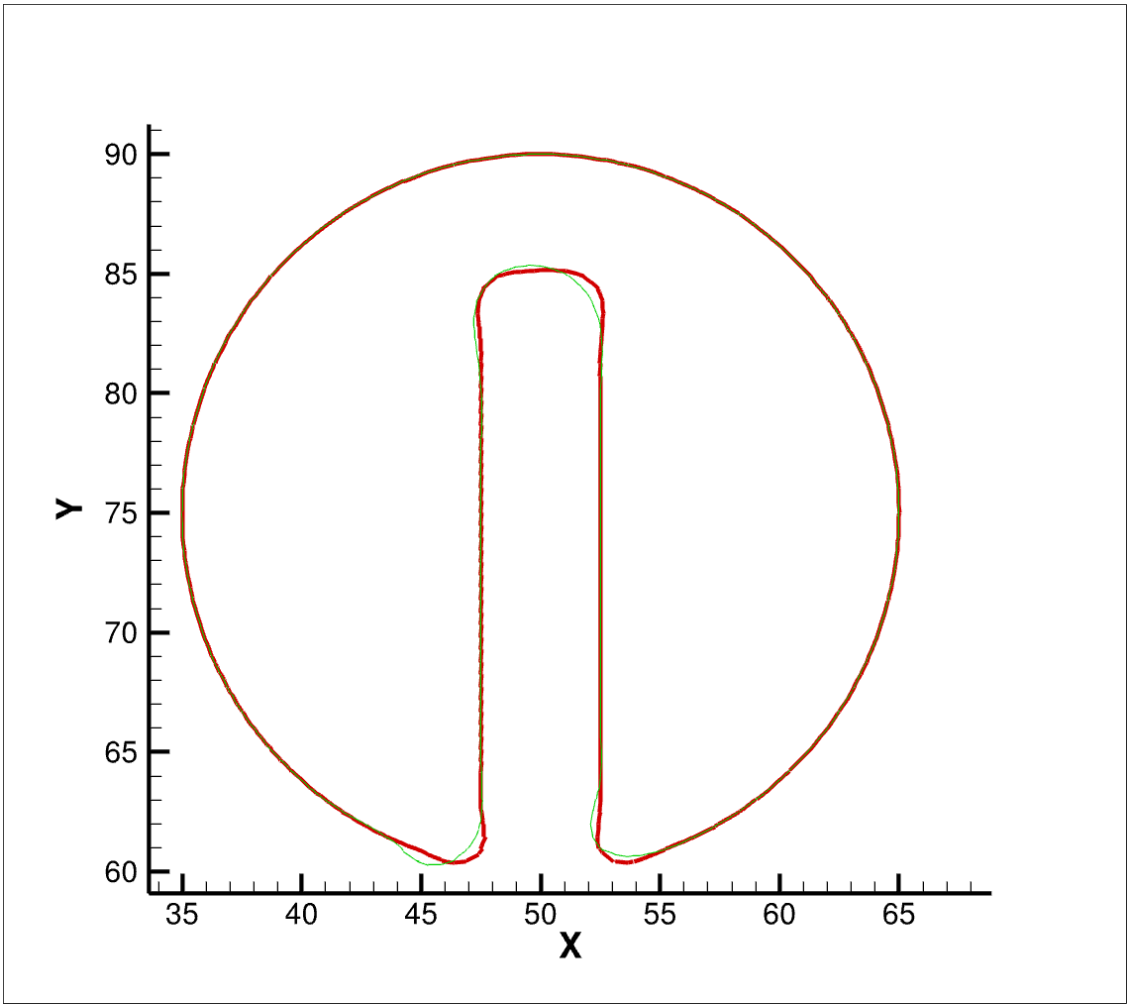


Figure 4.1: MOF compared with PLSMOF, Zalesak's problem, $t = 628.0$ on a 192^2 grid. red: MOF, green: PLSMOF

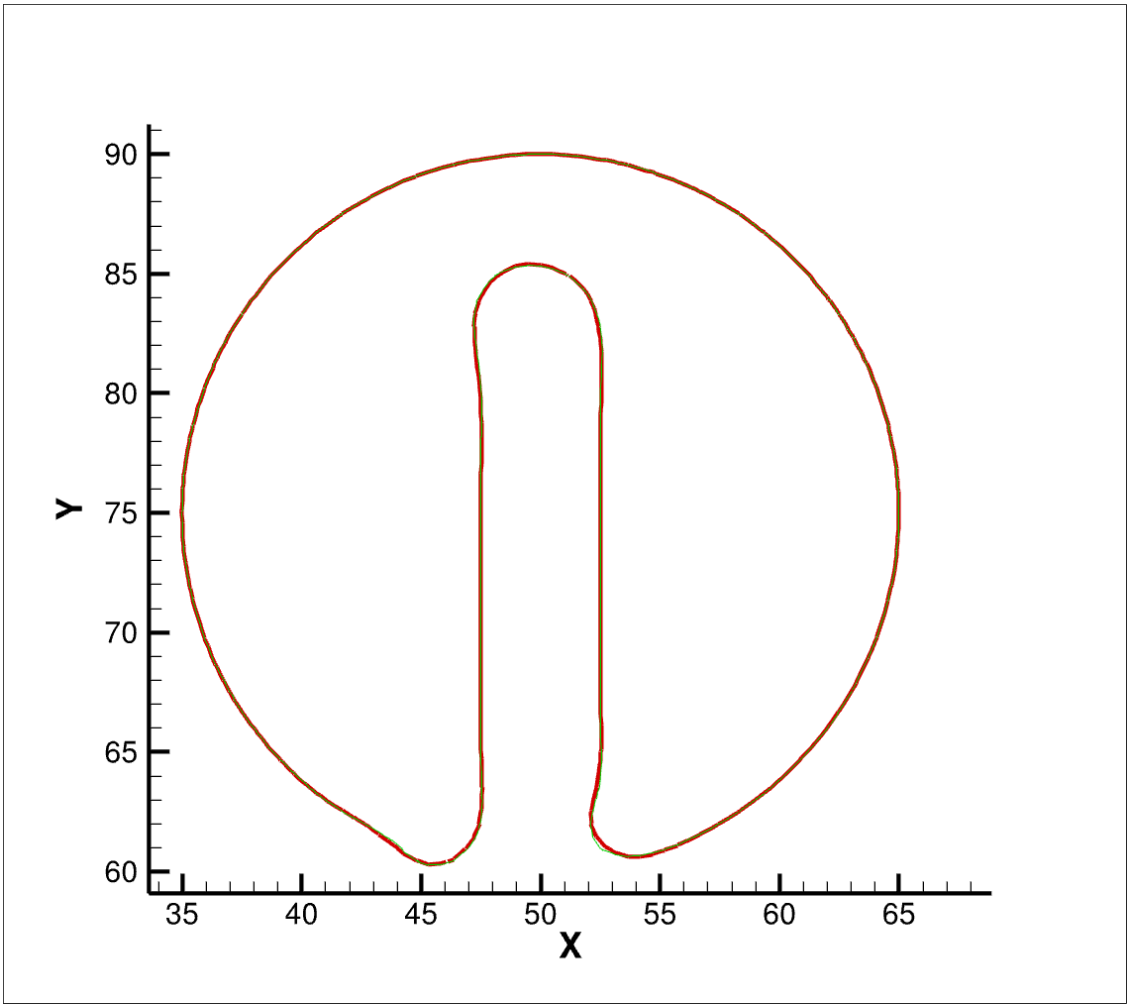


Figure 4.2: CMOF compared with PLSMOF, Zalesak's problem, $t = 628.0$ on a 192^2 grid. red: CMOF, green: PLSMOF

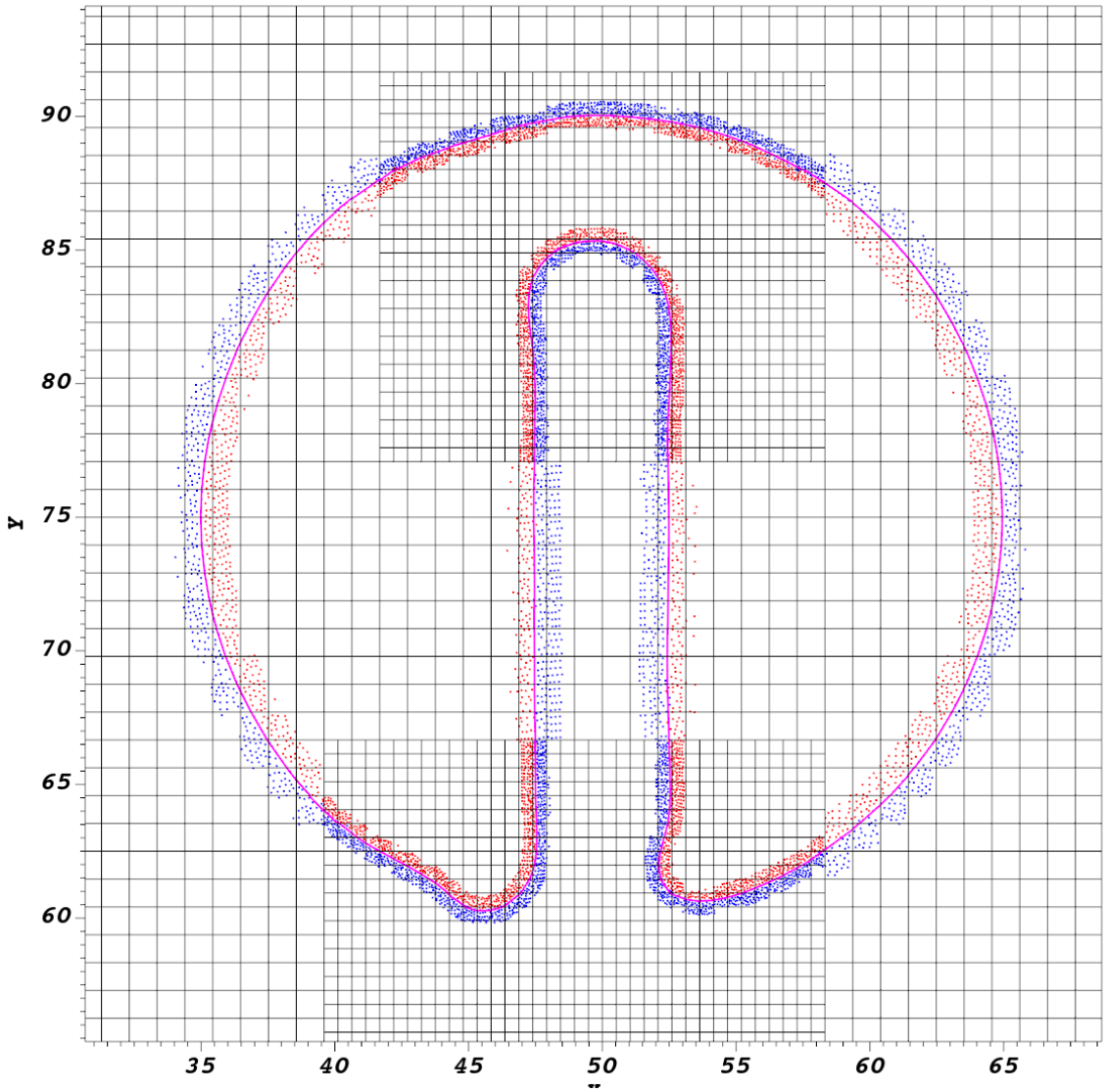


Figure 4.3: PLSMOF, curvature-based grid refinement, $t = 628.0$, finest grid: 192^2 .
red: interior particles, blue: exterior particles, pink: reconstructed interface
note: particle density is directly tied to the grid resolution.
More figures for this test problem are found in appendix B.1.
Remark: particles are contained only on the finest available mesh, in underlying regions and at coarse/fine boundaries, the MOF reconstruction is used instead of the particle-assisted reconstruction.

4.2 Rotating Disk

In this test, a disk (same as in the Zalesak test above, but without the notch cut out) comprising of two separate halves is rotated. It is set up as a multimaterial problem such that the left half of the disk is is material 1, the right half is material 2, and outside is material 3. The domain, velocity field, disk radius and position is the same as in the problem above.

In table 4.4, it can be seen that as with the Zalesak test (table 4.1). The MOF produces a much better result in terms of the symmetric difference error for rigid body motion. CMOF and PLSMOF produce similar results to one another on the rigid body tests. In table 4.5 we see a direct performance comparison between the three methods. The computational time for the PLSMOF is on par with that of the MOF method, both performing significantly faster than the reconstruction stage of CMOF.

While the MOF yields better results on the rigid body tests, it will be demonstrated in section 4.4 that MOF is susceptible to noise which it is unable to remove. However, both CMOF and PLSMOF are able to damp out these perturbations. It is important to note from the cost comparison (table 4.5) and results across the different test problems that PLSMOF behaves similarly to CMOF, but at a computational cost on par with that of MOF.

Table 4.4: Comparison of the symmetric difference error E_{sym} for the three materials after 5 revolutions ($t = 3140$)

	material 1	material 2	material 3
MOF	2.9	2.7	3.9
CMOF	19.3	14.0	18.9
PLSMOF	19.5	12.1	17.6

Table 4.5: Cost comparison for the three materials after 5 revolutions ($t = 3140$)
performance profiling times shown in seconds

MOF	reconstruction:	340
	Level set redistancing (MOF):	351
CMOF	reconstruction:	1837
	Level set redistancing (CMOF):	327
PLS-MOF	PLSMOF reconstruction:	330
	(intercept + slopes + particle advection procedures)	
	Level set redistancing (PLSMOF):	329

Table 4.6: Reconstruction cost (per cell)
three materials disk rotation (Step = 6280, $t = 3140$)
Note: the PLSMOF reconstruction only involves the intercept calculation.

MOF	iterations to converge:	6.9
	CPU time:	0.018
CMOF	iterations to converge:	13.6
	CPU time:	0.13
PLS-MOF	iterations to converge:	0
	CPU time:	0.009

Remark: CMOF is slower than MOF as it requires twice as many Gauss-Newton iterations to converge (see table 4.2 for the Zalesak test and table 4.6 for the disk). CMOF not only computes the centroid relative to the supercell, but also needs the volume fraction within the local cell as well. Additionally, the MOF implementation uses the 'Notus' library functions developed by Lemoine et al. [53, 62]. This library gives substantial speedup over the original MOF method as the cost function can be found without having to obtain the intercept, only once the slope calculation has converged is the intercept calculated. CMOF however is unable to take advantage of these routines. Note that in table 4.6, the reconstruction cost shown for PLSMOF is only the time to find one intercept per cell whereas the CMOF reconstruction finds 1 intercept per iteration (so if we multiply the PLSMOF time by the average CMOF iterations to converge, $13.6 \times 0.009 = 0.12$, we see that the intercept calculation time per intercept is similar).

To understand why CMOF is slower than MOF, we can examine its construction:
A supercell $\Omega^s_{i,j}$ is defined about center cell $\boldsymbol{x}_{i,j} = \{x_i, y_j\}$

$$\Omega^s_{i,j} = \left\{ \boldsymbol{x} : x \in \left[x_i - \frac{3\Delta x}{2}, x_i + \frac{3\Delta x}{2} \right], \quad y \in \left[y_i - \frac{3\Delta y}{2}, y_i + \frac{3\Delta y}{2} \right] \right\} \tag{4.2}$$

with volume fraction and centroid

$$F^{s,n}_{m,i,j} = \frac{\sum_{i'=-1}^{1} \sum_{j'=-1}^{1} F^n_{m,i+i',j+j'} V_{i+i',j+j'}}{\sum_{i'=-1}^{1} \sum_{j'=-1}^{1} V_{i+i',j+j'}} \tag{4.3}$$

$$\boldsymbol{x}^{c,s,n}_{m,i,j} = \frac{\sum_{i'=-1}^{1} \sum_{j'=-1}^{1} F^n_{m,i+i',j+j'} \boldsymbol{x}^{c,n}_{m,i+i',j+j'} V_{i+i',j+j'}}{\sum_{i'=-1}^{1} \sum_{j'=-1}^{1} F^n_{m,i+i',j+j'} V_{i+i',j+j'}}. \tag{4.4}$$

where s denotes the supercell, and the other variables are defined as for MOF (equations 3.2 and 3.3). The CMOF method is then to obtain the optimal slope $\boldsymbol{n}$ and intercept b such that the following minimization problem is satisfied:

$$E_{\text{CMOF}} = \| \boldsymbol{x}^{c,s}_{\text{ref}} - \boldsymbol{x}^{c,s}_{\text{act}}(\boldsymbol{n}, b) \|_2 \tag{4.5}$$

with the associated constraint

$$F_{\text{act}}(\boldsymbol{n}, b) = F_{\text{ref}}. \tag{4.6}$$

Where the reference volume fraction corresponds to the center cell MOF volume fraction, $F_{\text{ref}} \equiv F^n_{m,i,j}$, and the reference centroid is that of the supercell, $\boldsymbol{x}^c_{\text{ref}} \equiv x^{c,s,n}_{\text{m,i,j}}$. That is, we find the slope that minimizes the distance between the supercell and CMOF centroid, subject to the constraint that the center cell's volume fraction must match the original volume fraction (see Fig. 4.4). Note: in the situation where the slope/intercept produces a line outside of the center cell, the supercell volume fraction is instead as the reference.

The Gauss-Newton method is used to solve for the slope $\boldsymbol{n}$ that minimizes equation 4.5. Then a hybrid bisection/Newton method is used to find the corresponding intercept b such that 4.6 is satisfied. Due to the slope reconstruction of CMOF depending on the supercell, the speedup from the analytic cell intercept from Lemoine et al. [53, 62] cannot be used, as they are derived for a single cell. In Ye et al. [116], a machine-learning decision tree method is instead used for speedup (CMOF-ML); in which a preprocessing step computes a decision tree for slopes and intercepts corresponding to volume fractions within a cell. However, in these benchmark cases, only the Gauss-Newton CMOF (CMOF-GN) is used.

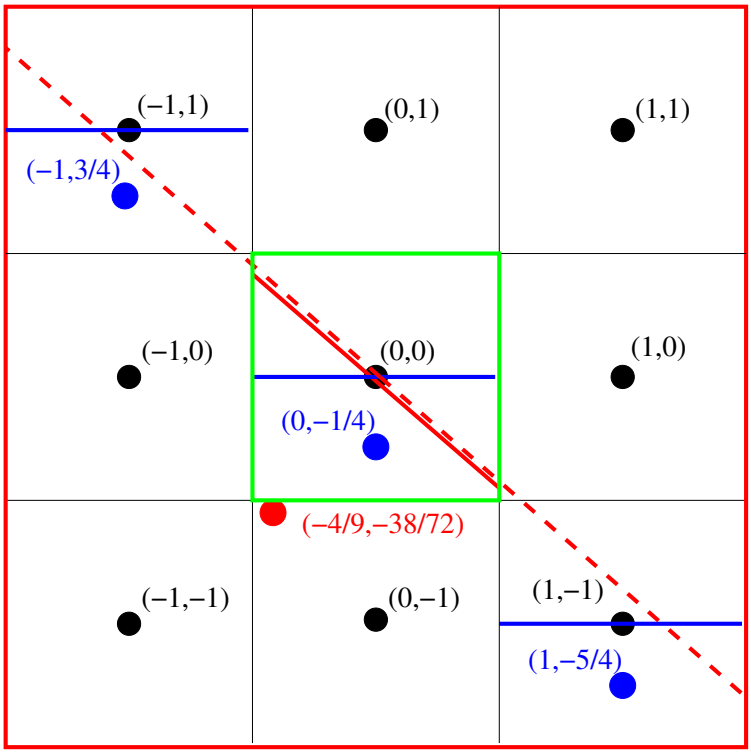


Figure 4.4: CMOF reconstruction. Black dots represent cell centroids, blue dots represent 'material 1' centroids, and the red dot represents the CMOF supercell (red boundary) centroid associated with the center cell (green boundary) for 'material 1'.
The CMOF reconstructed interface (solid red line inside the center cell) is found such that it is the line segment which minimizes the difference between the CMOF reference centroid (red dot) 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 within the center cell) equals the actual CMOF reconstructed volume fraction ('material 1' volume fraction below the solid red line in the center cell).

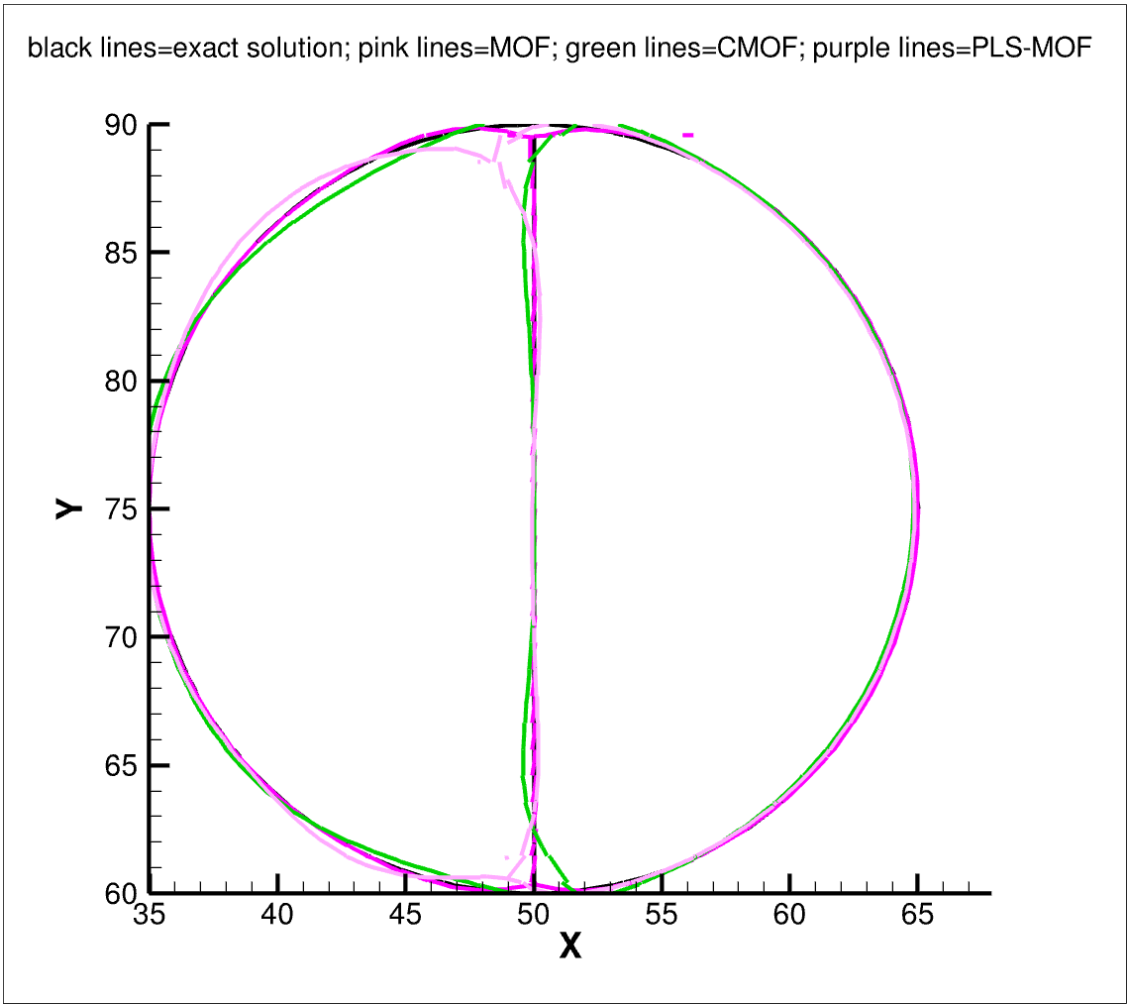


Figure 4.5: Results after 5 revolutions of the half disks at $t = 3140$.
Symmetric error and profiling shown in tables 4.4 and 4.5.

4.3 Bubble Formation

Here we present the results for the 'bubble formation' [67] test. A submerged nozzle, located at the bottom of the domain, slowly injects gas into the fluid-occupied region. The nozzle radius is 0.085cm. The finest grid resolution $\Delta x_{\text{fine}} = 0.010625$cm. The domain and physical parameters used correspond to those in [67]. The simulation is performed in axisymmetric "RZ" coordinates, the top of the domain has an outflow boundary condition and the other walls are slip boundaries (except for the axis of symmetry).

Figures 4.6, 4.7, and 4.7 show the reconstructed interfaces for the MOF, CMOF, and PLSMOF tests, respectively. After the initial transients end, the expected effective bubble diameter is $d_{effective} = 0.499$cm.

Note: the results for MOF (Fig. 4.6) exhibit flotsam and non-physical noise due to instabilities introduced at the nozzle. CMOF and PLSMOF (Figs. 4.7 and 4.8) do not exhibit this behavior (this is further explored in the next section 4.4). Additionally, both CMOF and PLSMOF produce almost identical $d_{effective}$ values.

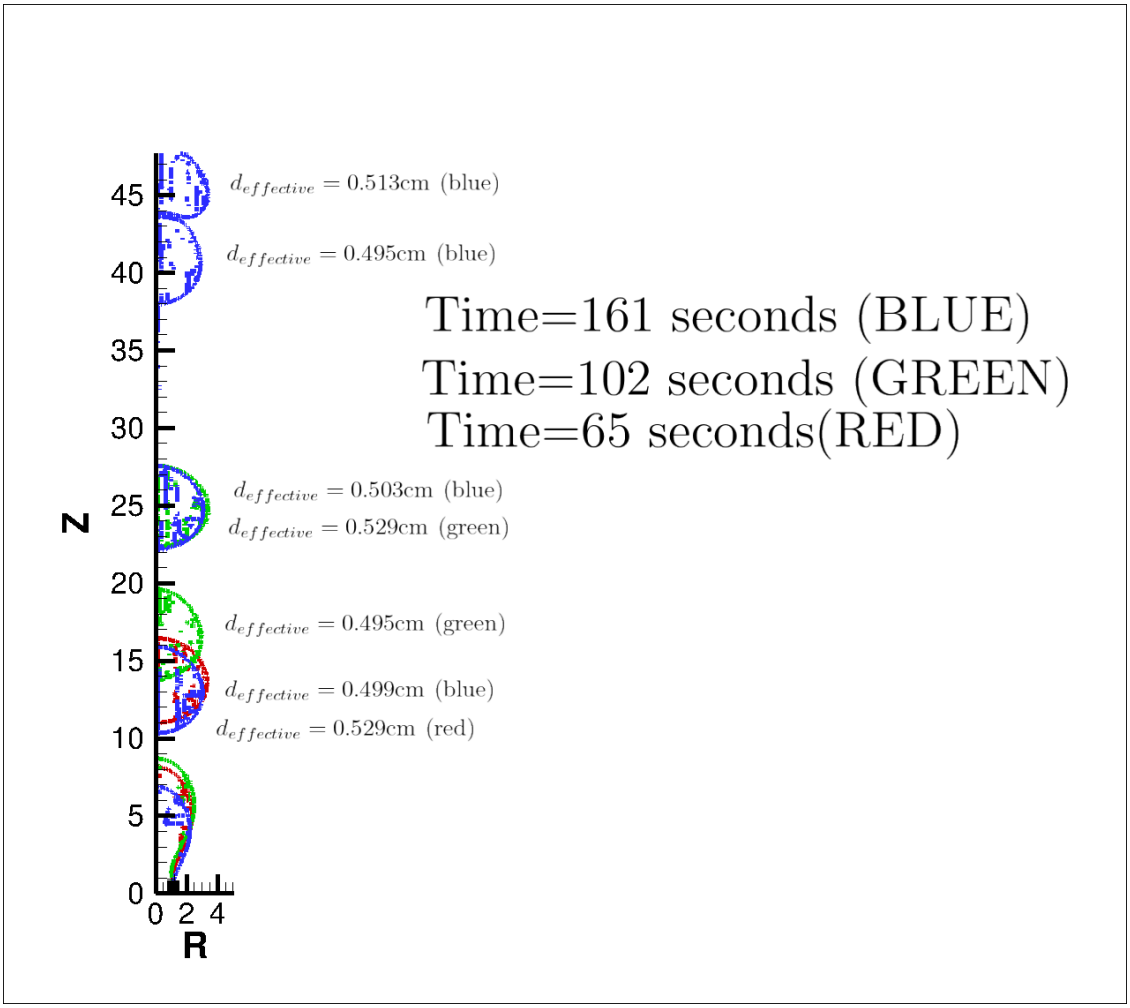


Figure 4.6: **MOF**, bubble formation problem.
Effective fine grid resolution : $\Delta x_{\text{fine}} = 0.010625$cm
Step numbers are 13400 ($t = 65.3$), 21600 ($t = 101.66$), and 35800 ($t = 160.7$).

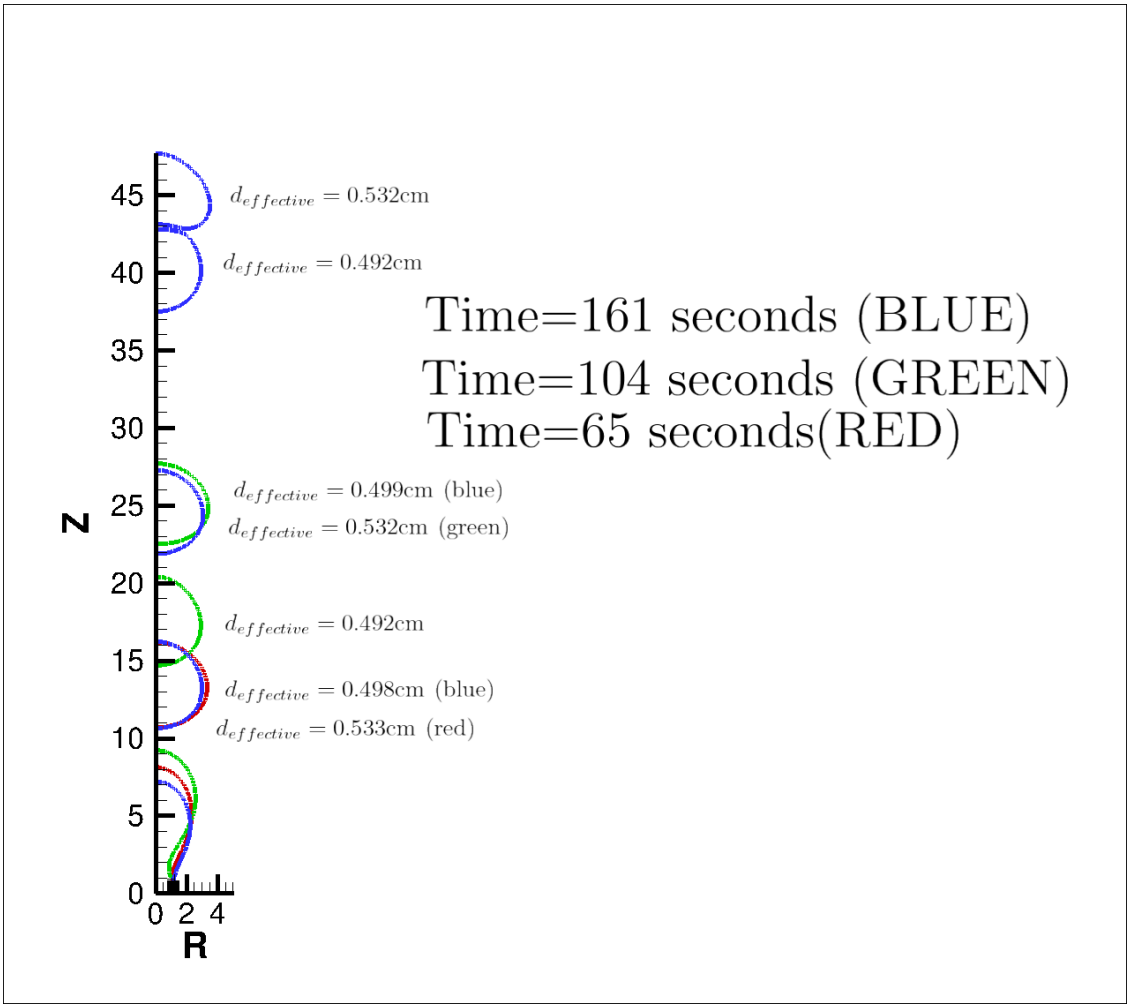


Figure 4.7: **CMOF**, bubble formation problem.
Effective fine grid resolution : $\Delta x_{\text{fine}} = 0.010625$cm
Step numbers are 13400 ($t = 65.3$), 21600 ($t = 103.8$), and 34200 ($t = 161.3$).

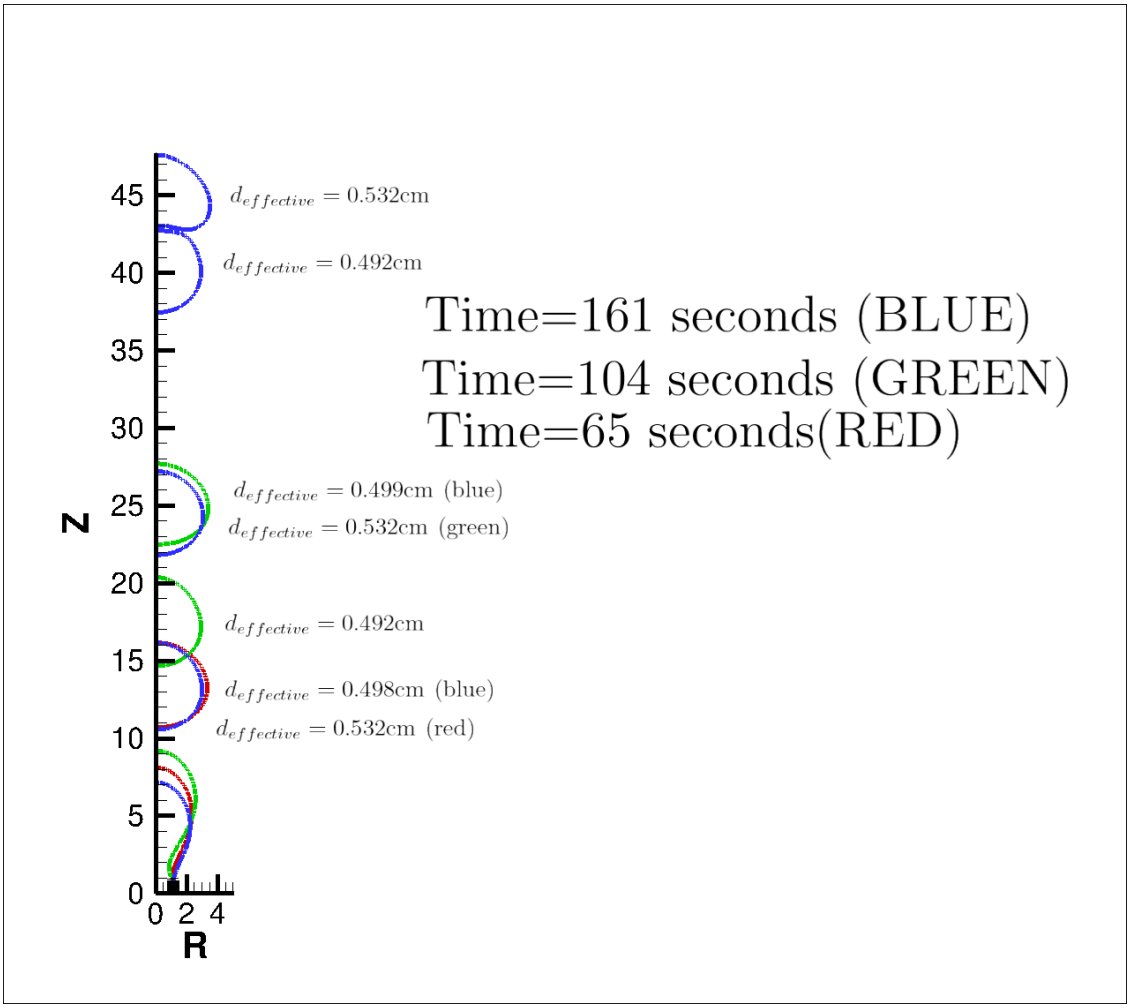


Figure 4.8: **PLSMOF**, bubble formation problem.
Effective fine grid resolution : $\Delta x_{\text{fine}} = 0.010625$cm
Step numbers are 13400 ($t = 65.3$), 21600 ($t = 103.8$), and 34200 ($t = 161.2$).

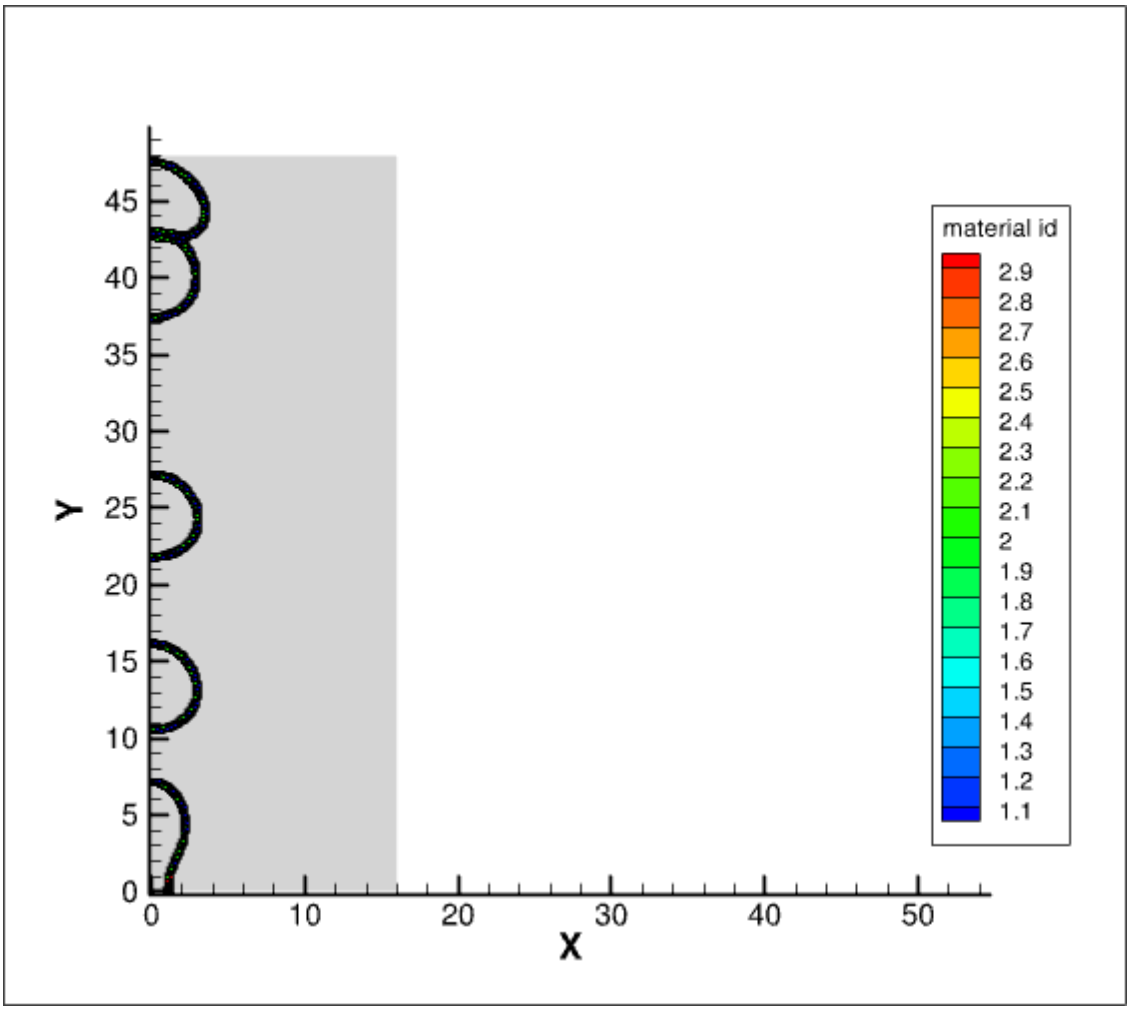


Figure 4.9: **PLSMOF**, bubble formation problem,
shown with particles.

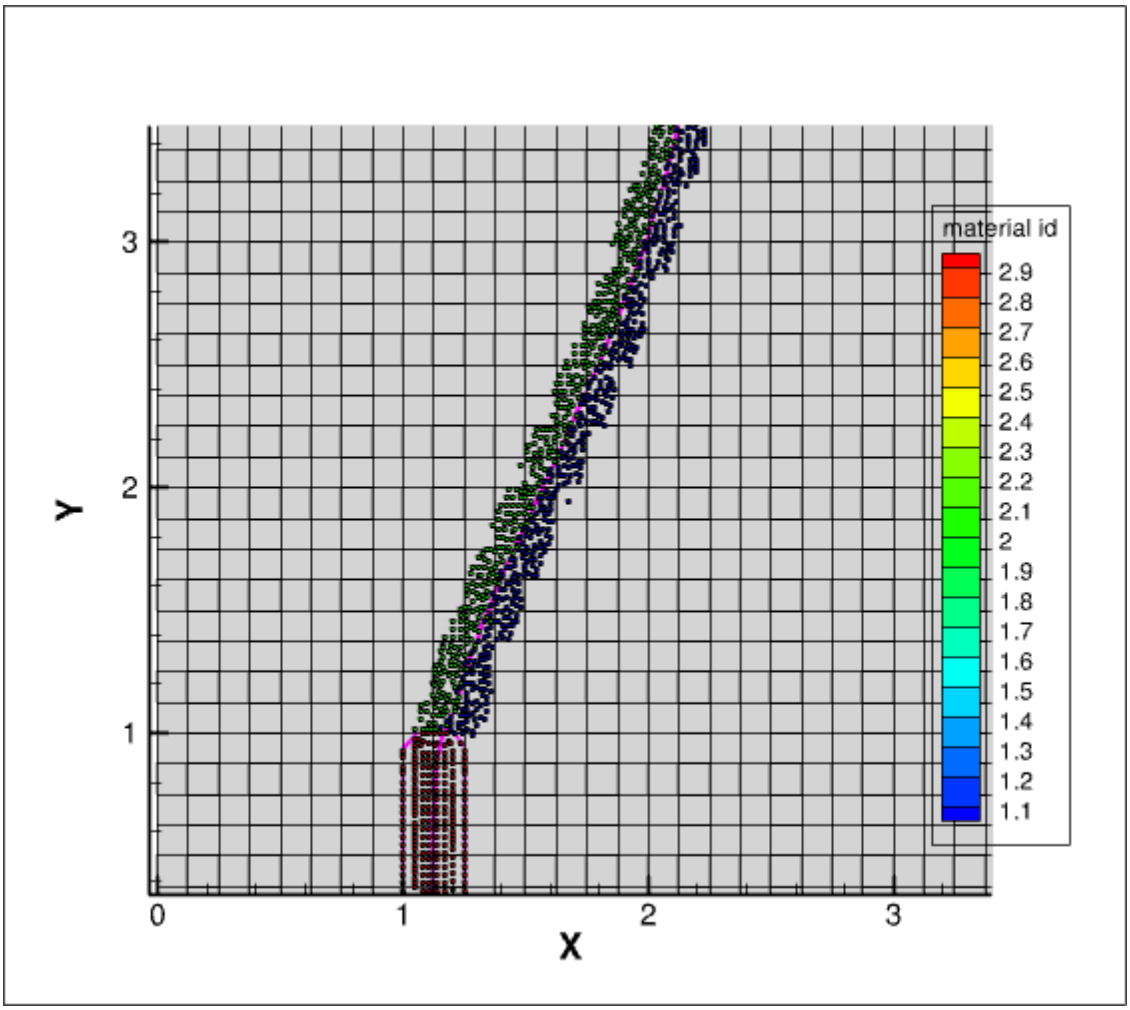


Figure 4.10: **PLSMOF**, bubble formation problem, nozzle closeup.
red: nozzle particles, blue: liquid particles, green: vapor particles, pink: piecewise linear interface

4.4 Stretching Liquid Lens

In this test, we examine the behavior of MOF, CMOF, and PLSMOF in the presence of noise. The test used in this comparison is the 'liquid lens' [49, 54]. This simulation demonstrates the stretching of a 'liquid lens' due to the surface tension forces at the junctions where the three fluids meet.

The liquid lens test is set up as follows: The center of the domain is occupied by a circular fluid material m_2 with diameter 0.3. The remaining space is filled by two different fluid materials, one in the top half m_1 and the other in the bottom m_3 (Figure 2.3). All three of these materials are assigned the same density $\rho = 1$ and viscosity $\mu = 1/60$. The surface tension coefficients are as follows: between the top material and the center material $\sigma_{12} = 2/45$, between the bottom material and the center material $\sigma_{23} = 2/45$, and between the top and bottom materials we have $\sigma_{12} = 5/90$. The computational domain is $[0, 1] \times [0, 1]$.

Additionally in this problem setup, noise is intentionally added to the initial centroids as

$$\boldsymbol{x}^{\text{ref}}_{\text{noise}} = \boldsymbol{x}^{\text{ref}} + (2r - 1)\Delta x, \tag{4.7}$$

where r is a random variable sampled from a uniform distribution ($0 \leq r \leq 1$).

The expected steady-state solution should have material m_2 being stretched into the shape of a biconvex lens with a major axis length $L_0^{\text{exact}} = 0.460$. The corresponding perimeter of the lens at the steady state is $\mathcal{P}_0^{\text{exact}} = 1.055$. Due to presence of both surface tension and viscosity forces, one would expect that physically, any initial noise should be smoothed and damped out. However due to the locality of the MOF construction, this noise persists (see figure 4.11). In contrast, CMOF (figure 4.12) and PLSMOF exhibit an immediate smoothing, rapidly damping out the noise. Note that the "high frequency" signals potentially generated by the MOF reconstruction or damped out by CLSMOF and PLSMOF are not recognized by conventional curvature discretization algorithms.

In table 4.7 below, a comparison is made between the three methods at the steady state for the computed major axis length L_0. Figure 4.14 shows a plot of the measured lens perimeter over time vs the analytic solution.

This test demonstrates how the MOF reconstruction is unable to smooth out the initial noise as the CMOF and PLSMOF methods accomplish due to neighboring information being used during the reconstruction procedure. In this test we obtain similar results between CMOF and PLSMOF,

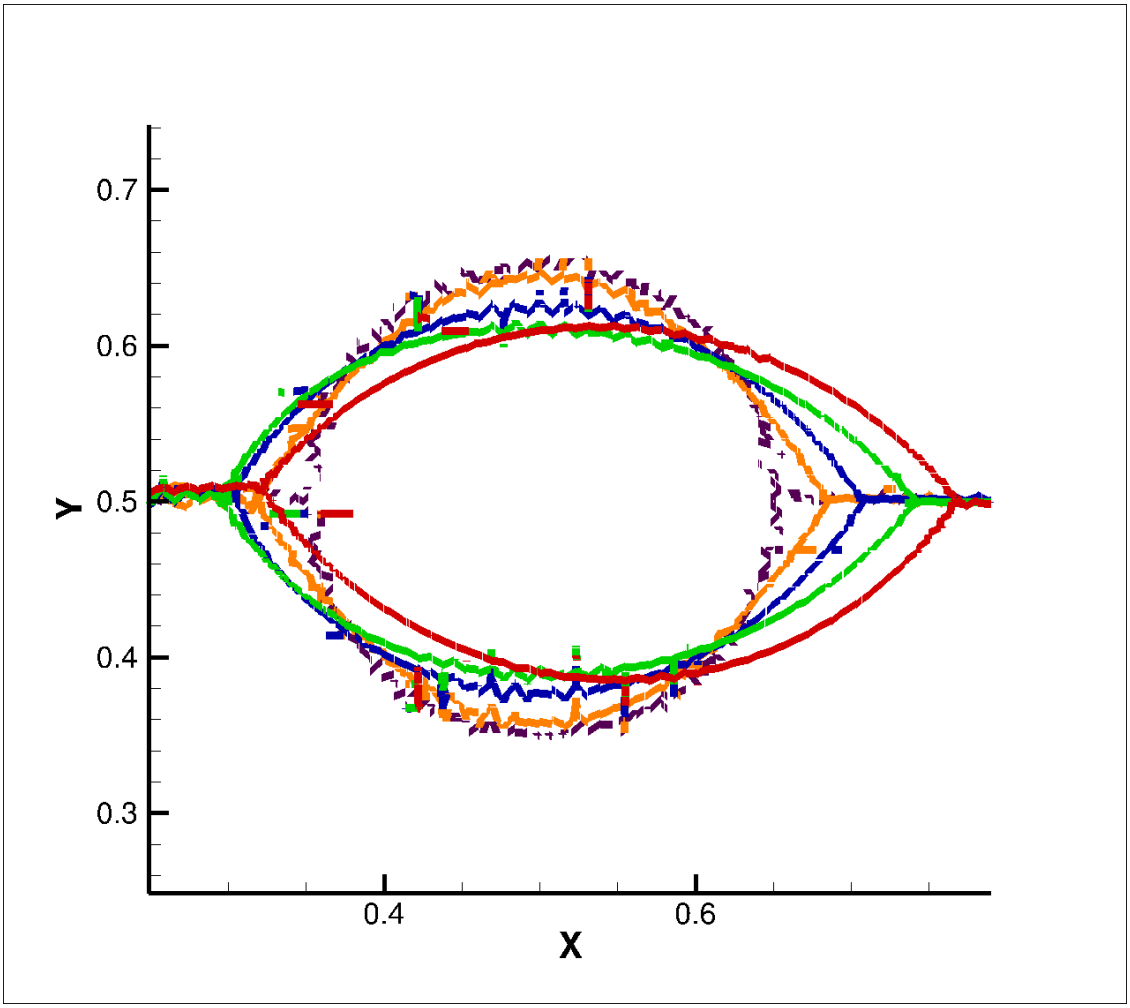


Figure 4.11: **MOF**, stretching of a liquid lens.
Times shown are:
0.0 (brown), 0.187 (orange), 0.375 (blue), 0.749 (green), and 4.0 (red).
Effective fine grid resolution : $\Delta x_{\text{fine}} = 1/128$.
Initial noise persists.

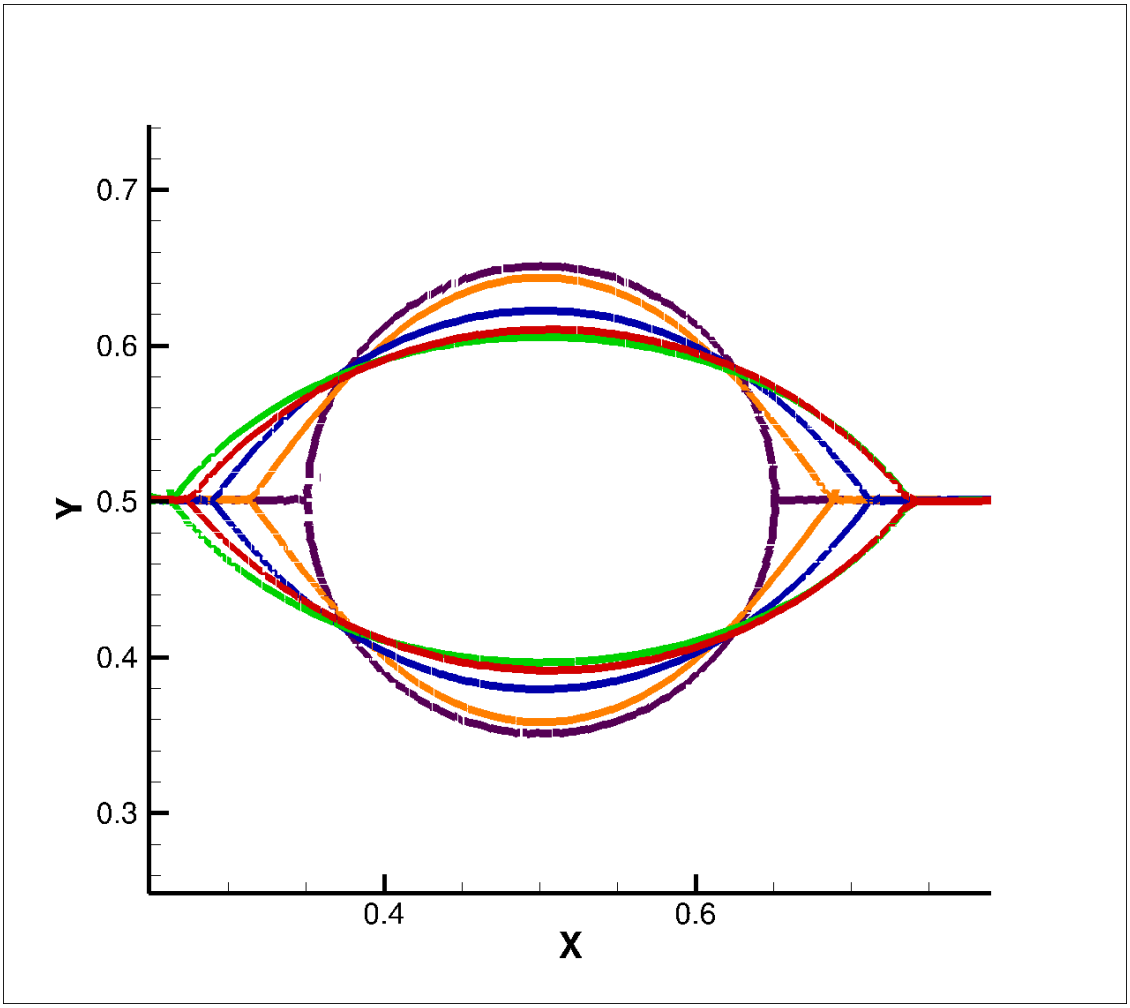


Figure 4.12: **CMOF**, stretching of a liquid lens.
Times shown are:
0.0 (brown), 0.187 (orange), 0.375 (blue), 0.749 (green), and 4.0 (red).
Effective fine grid resolution : $\Delta x_{\text{fine}} = 1/128$.
Initial noise immediately damped out (c.f. Figure 4.11).

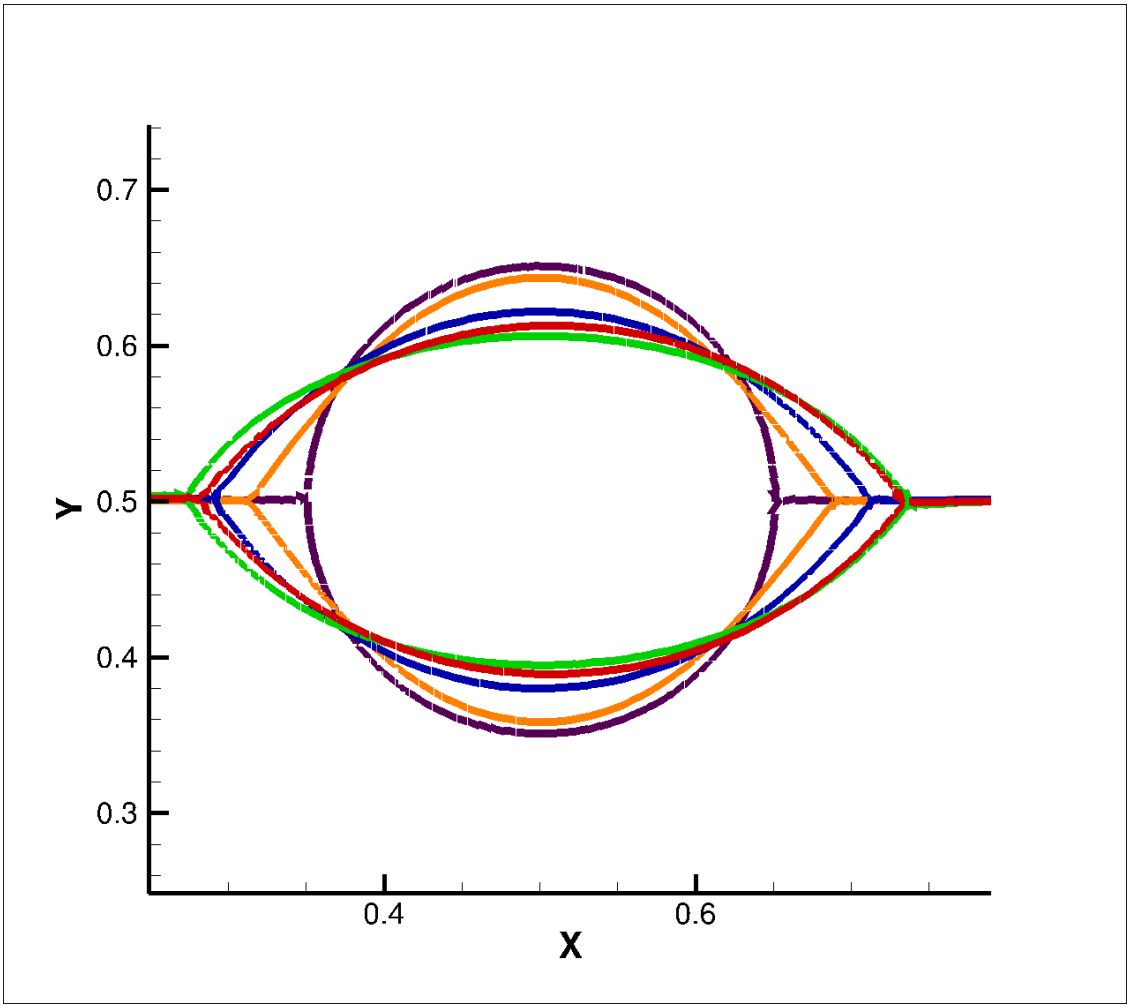


Figure 4.13: **PLSMOF**, stretching of a liquid lens.
Times shown are:
0.0 (brown), 0.187 (orange), 0.375 (blue), 0.749 (green), and 4.0 (red).
Effective fine grid resolution : $\Delta x_{\text{fine}} = 1/128$.
Initial noise immediately damped out (c.f. Figure 4.11).

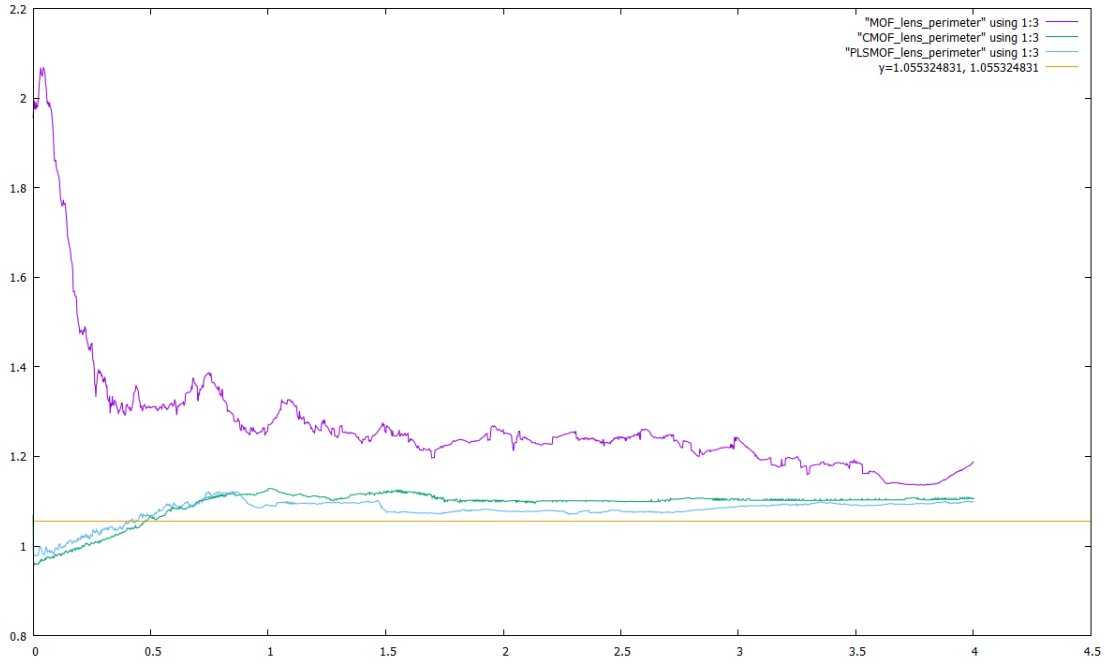


Figure 4.14: Lens perimeter vs time
purple: MOF, green: CMOF, blue: PLSMOF, orange: $\mathcal{P}_0^{\text{exact}}$
Note: at the initial time, the perimeter calculation yields a 2% measurement error from top and bottom material initial MOF perimeter measurements, so we expect $\sim$4% error in perimeter measurement.

Table 4.7: MOF, CMOF, and PLSMOF comparison of steady-state major axis length L_0, measured from the reconstructed interface. L_0^{exact} is the major axis length of the expected steady-state. Note: This distance is measured as the horizontal distance between the reconstructed triple points of the stretching lens. The numerical solutions may deflect diagonally off-axis and therefore not be a sufficiently representative metric. Figure 4.14 instead shows the measured perimeter of the lens compared to the analytic solution.

Method	L_0	L_0^{exact}
MOF	0.441	0.460
CMOF	0.468	0.460
PLSMOF	0.450	0.460

both exhibiting the same damping behavior and producing comparable results (though they vary in terms of the calculated lens perimeter). Unlike CMOF and PLSMOF, the MOF reconstruction doesn't 'see' beyond the local cell and therefore will be unable to smooth the reconstruction between cells.

In large Reynolds number (Re[1]) and large Weber number (We[2]) regimes, the MOF reconstruction has shown good results, such as in the context of turbulent jets with atomization and spray [6]. However, the observed persistence of noise in the MOF scheme shown in this test indicates that the MOF reconstruction may not be as suitable for low Re or low We regimes in which the effects of surface tension may dominate. The reconstructed MOF interface does not respond to the effects of 'smoothing out' behavior of surface tension and therefore would not be expected to directly damp out any surface waves, allowing them to persist. Figure 4.15 below shows another example of how the local reconstruction is unaffected by any neighboring data. This motivates the use of schemes like CMOF and PLSMOF in this context that use alternative slope reconstruction procedures that are able to be influenced by neighboring information.

[1] $Re = \frac{\rho u L}{\mu}$ (ratio of inertial forces to viscous forces)

[2] $We = \frac{\rho u L}{\sigma}$ (ratio of drag force to cohesion force)

ρ: density, u: velocity, L: characteristic length, μ: dynamic viscosity, σ: surface tension

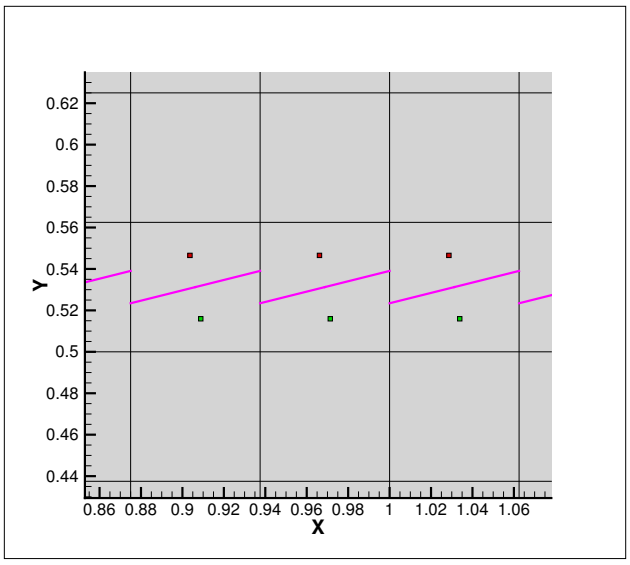


Figure 4.15: Saw-tooth interface. As demonstrated in Ye et al. [116], Repeated MOF reconstructions of this interface do not alter the initial jagged interface. CMOF, and similarly PLSMOF, due to the use of information from neighboring cells will smooth out any such interfaces, removing the jagged reconstruction features seen in MOF.

4.5 Freezing Droplet

For this test, results for the 'freezing of water droplet on a cold substrate' problem [42, 106] are presented and compared. This test involves three deforming liquid materials and fourth solid substrate material (refer to figure 2.2 for the initial material setup). Note: in order to capture the phase change, the ice material is treated as a fluid body material rather than rigid. The substrate upon which the droplet sits is prescribed a constant temperature T_w. The physical parameters used for the experiment is listed in table 4.8 and the interfacial parameters in table 4.9.

Table 4.8: Physical parameters of the air, water, ice, and substrate materials.

variable	substrate	ice	water	air
density (ρ)	rigid	0.917	1.0	0.00129
viscosity (μ)	rigid	rigid	1.7E−2	1.7E−4
modeling viscosity (μ_{model})	rigid	rigid	100.017	100.00017
modeling gravity ($\boldsymbol{g}$)	rigid	rigid	9.8E−4	9.8E−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

The 'modeling' coefficients in tables 4.8 and 4.9 have each been scaled (e.g. by a factor of 10^{-6} for surface tension) in order to give agreement with experiments from [42] (c.f. table 4.10). This is motivated in a similar manner to Denner et al. [25] in which artificial viscosity is added to the problem in order to eliminate unphysical vibrations of the liquid/gas interface.

This has the added effect of greatly relaxing the time step constraint (3.16) due to the surface tension,

$$\Delta t < \Delta x^{3/2} \sqrt{\frac{\rho_L + \rho_G}{2\pi\sigma}}, \tag{4.8}$$

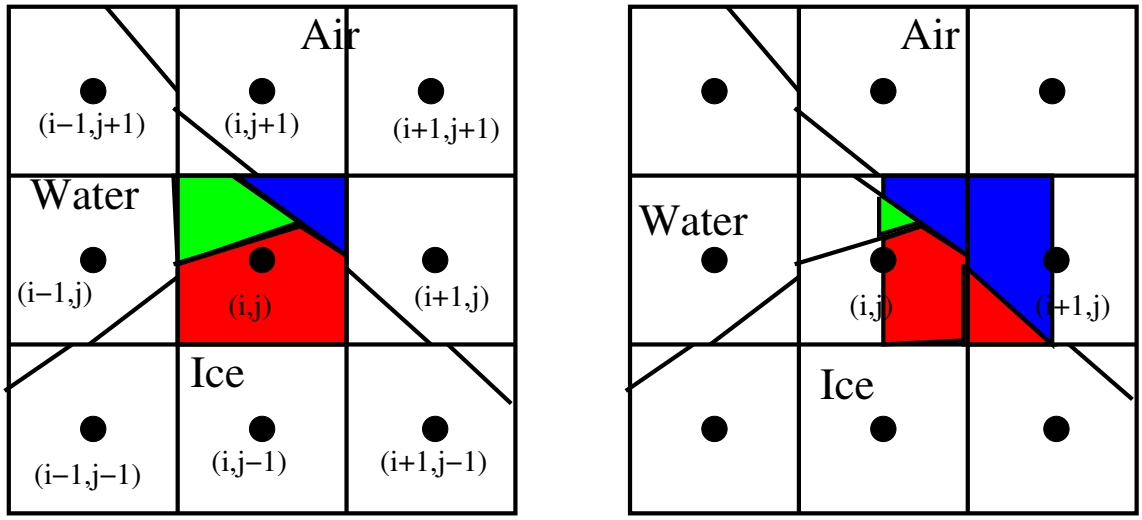
since σ_{m_1,m_2} is scaled down.

Table 4.9: Interfacial Physical properties for comparing with the icing experiments from [42].

variable	value
surface tension (σ)	72.8
modeling surface tension (σ_{model})	7.28E−5
physical latent heat (L)	3.34E+9
modeling latent heat (L_{model})	65.0E+9
saturation temperature (T_{sat})	273.0

The volume weights of cell-centered or MAC quantities (denoted 'mix', e.g. in equations 3.35, 3.36, and 3.37) are determined by the volume fraction of the reconstructed interface within the corresponding control volume (cell-centered or face-centered). Figure 4.16 demonstrates the control volume weighing.

Figures 4.17 and 4.18 compare the freezing test, both in planar R-Z and for the hemispherical droplet to the 1D analytic Stefan model. Both cases show agreement (with some slight discrepancy for the hemispherical case as it is not as directly comparable to the analytic problem as the horizontal surface case).



(a) cell-centered volume weights

(b) MAC grid volume weights

Figure 4.16: Illustration of volume weights for computing the cell-centered (left) and MAC (right) grid mixture parameters.
(a) Volume weights (within control volume (i, j) for the mixture product of density and specific heat ρC_p, and the cell-centered mixture density ρ (as appears in the viscosity equation Eqs. 3.36).
(b) Volume weights (within control volume $(i + 1/2, j)$) for the MAC-grid mixture of the density, $\rho^{\text{MAC,mix}}$, (as appears in the pressure gradient equation Eq. 3.37).

Table 4.10: Time to freeze a water droplet with initial volume 4.95E−5cm^3 as a function of the wall temperature (°C) 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 [42]. Much closer agreement is made with experiments when the latent heat is prescribed as 65.0E+9 as opposed to when he 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

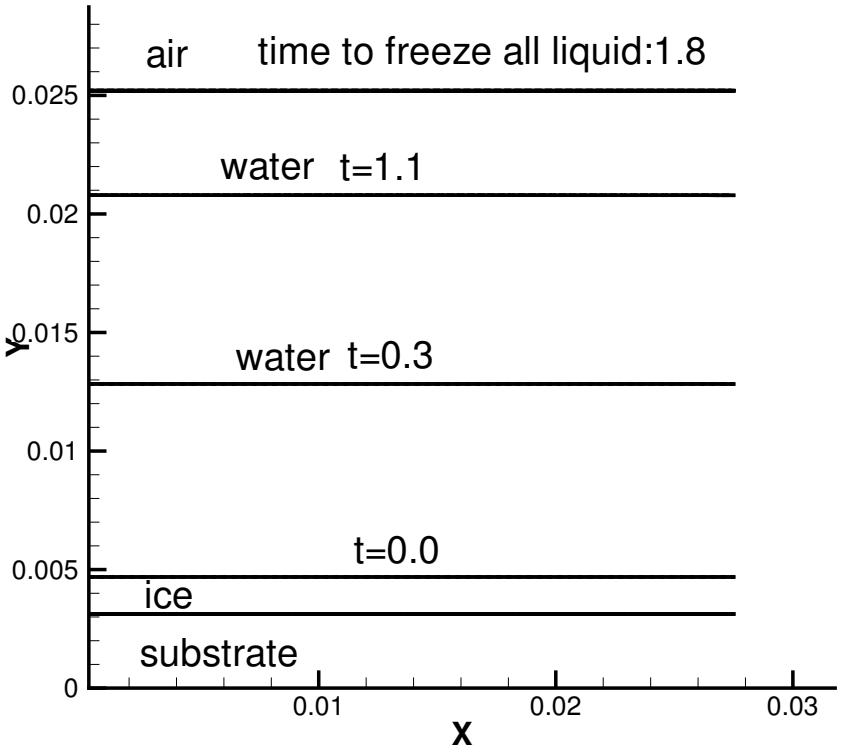


Figure 4.17: Freezing of a planar interface in R-Z coordinates.
The substrate temperature is maintained at a constant $-2°$C.
The latent heat is 3.34E+9 erg/g.
Grid size is 64×64.

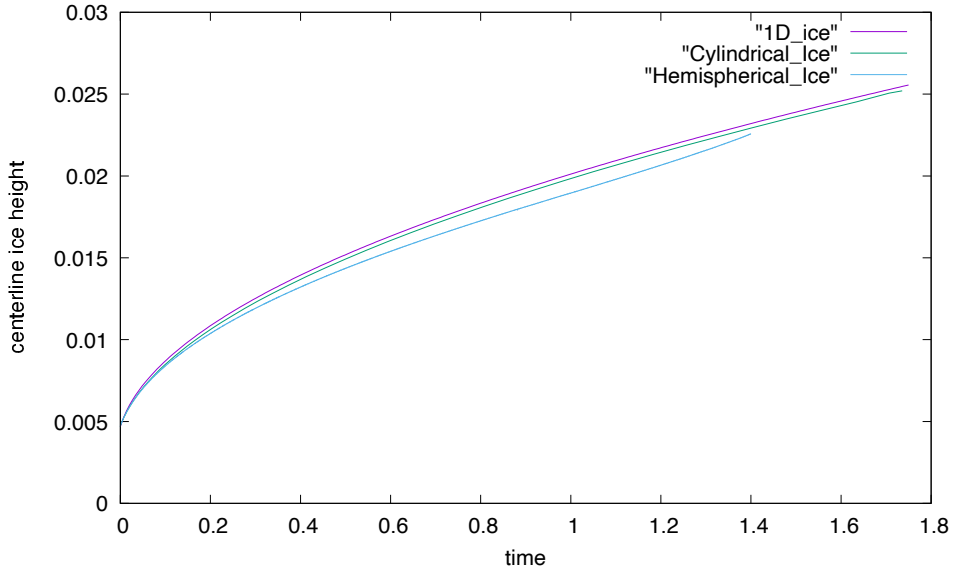


Figure 4.18: **CMOF**: Verification of the Stefan model: 1D model, cylindrical R-Z test (Fig. 4.17), and 3D hemispherical ice test. Position versus time of the freezing liquid-ice interface is plotted. The rate of freezing of the 3D Hemispherical droplet is within 5% of the "comparable" cylindrical freezing case. The Cylindrical R-Z freezing rate agrees within 1.1% 1D analytical model.

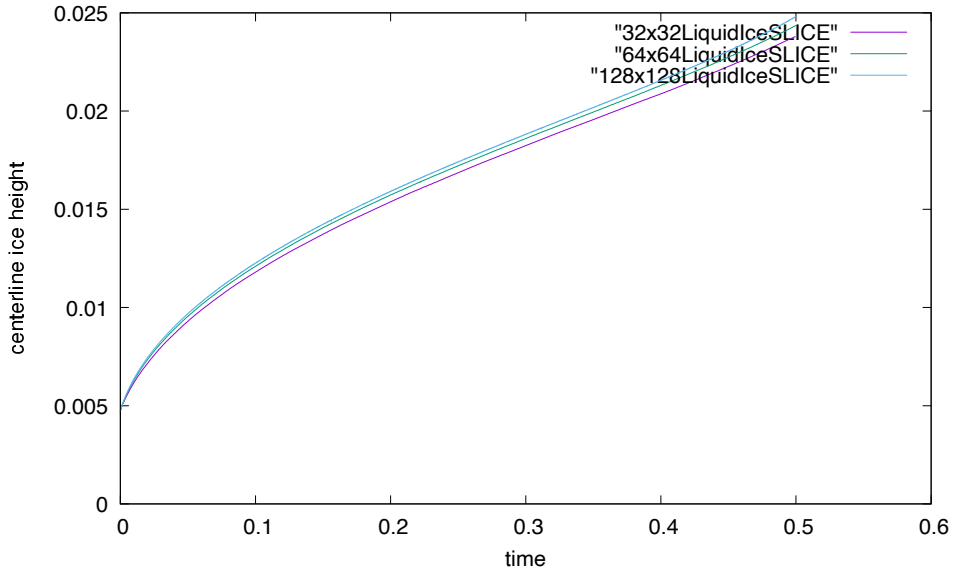


Figure 4.19: **CMOF**: Convergence study: solidification of a liquid droplet on a cold substrate. The position versus time of the liquid-ice interface is plotted for several grid refinement levels. The substrate temperature is maintained at a constant $-2°$C.
The Latent Heat is 3.34E+9 erg/g. Grid sizes are 32×32, 64×64, and 128×128.

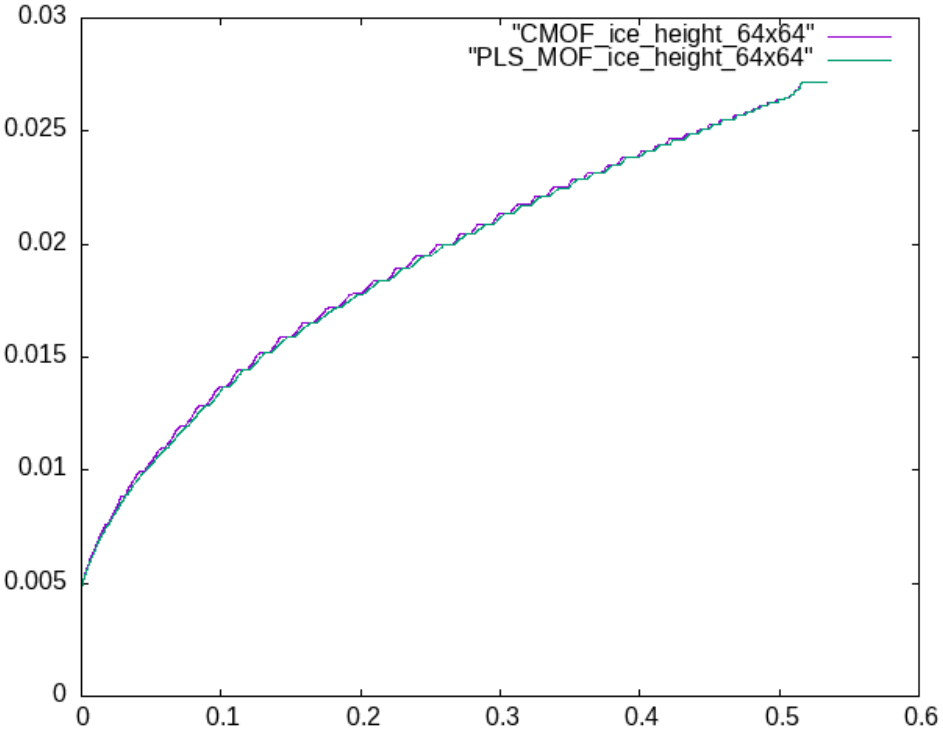
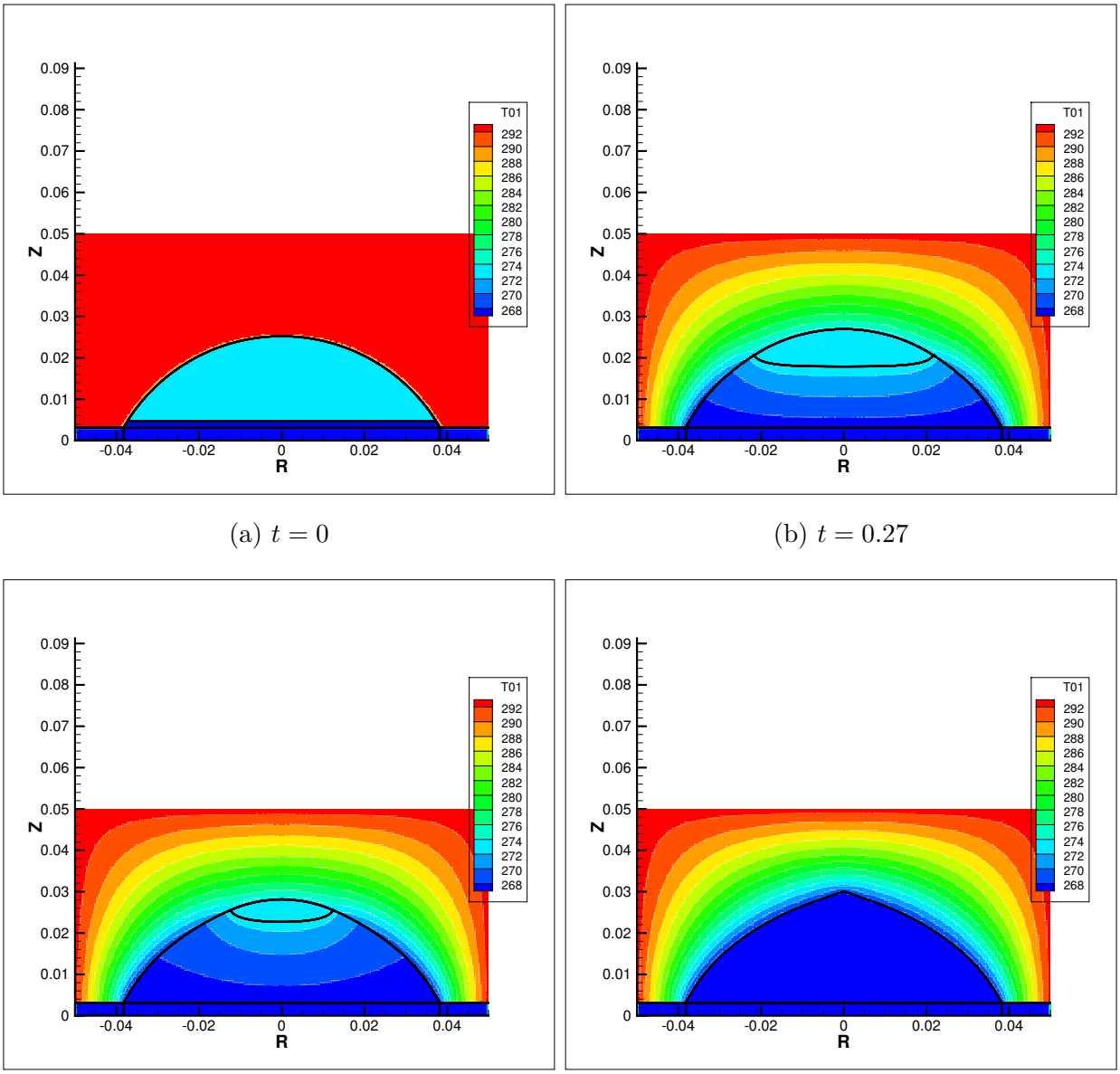


Figure 4.20: Ice height vs time. 64×64 grid
purple: CMOF, green: PLSMOF
PLSMOF and CMOF produce near-identical results on this test, profiling benchmark in table 4.11.

Table 4.11: Performance comparison: CMOF vs PMOF 64×64 freezing test (Fig. 4.20).
3000 steps up to time $t = 0.5085$. Profiling times shown in seconds.
Note: interface reconstruction for CMOF involves the Gauss-Newton slope routines. For PLSMOF it entails the combined cost of the particle addition/deletion, advection, and slope reconstruction routines.

CMOF	interface reconstruction:	1034
	Level set redistancing (CMOF):	168
PLS-MOF	interface reconstruction:	376
	Level set redistancing (PLSMOF):	168

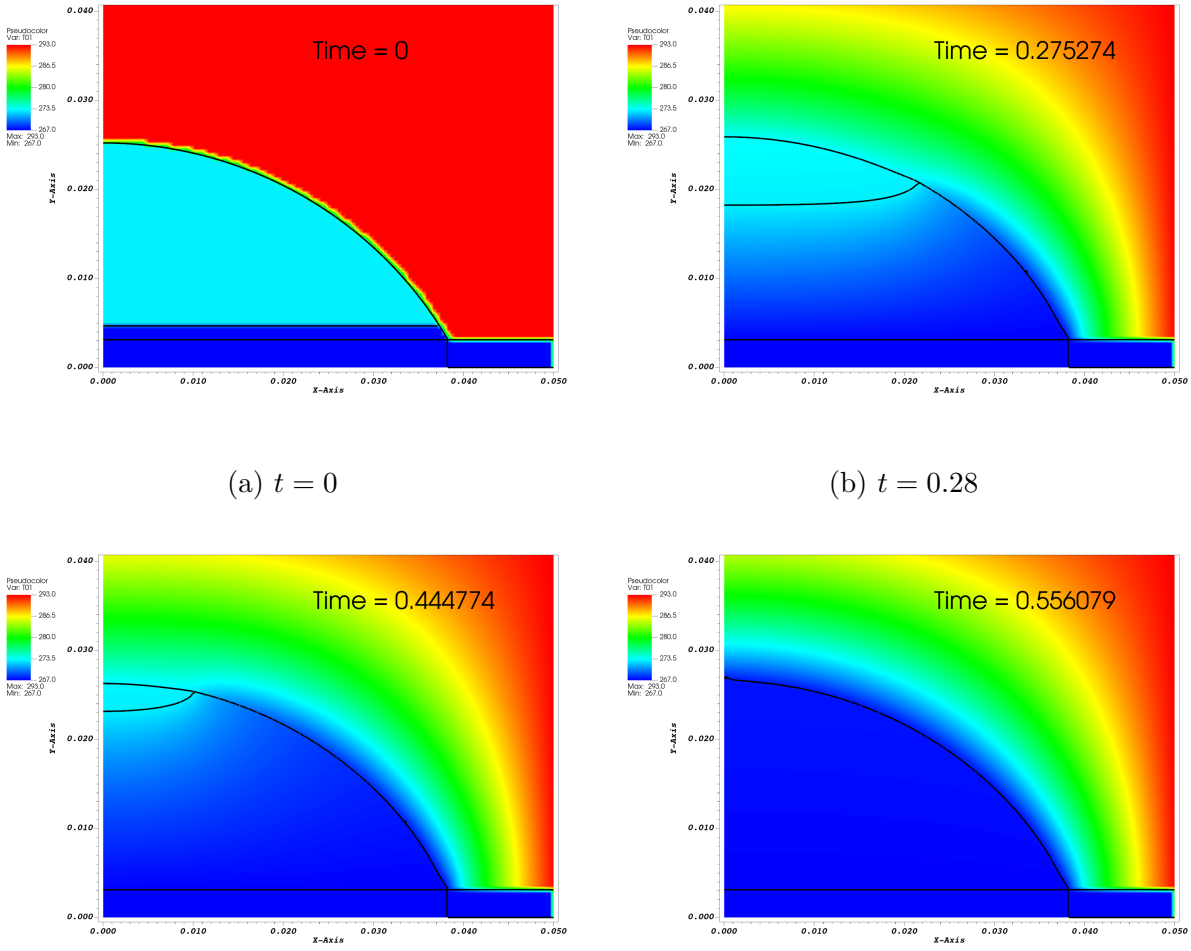
Again, when comparing CMOF with PLSMOF (see test in figure 4.20 and corresponding benchmark in table 4.11), we see that both methods produce almost the same results. However the PLSMOF method is significantly faster CMOF. It must be noted that all the benchmarks between PLSMOF and CMOF have been using the Gauss-Newton procedure for CMOF. Ye et al. [116] proposed an alternative machine-learning procedure (CMOF-ML) in which a training cost is incurred at the start, followed by significant speedup. PLSMOF has not yet been compared against CMOF-ML. However, the CMOF-ML method is not generalizable to unstructured grids, PLSMOF on the other hand is generalizable and does not have that limitation.



(a) $t = 0$ (b) $t = 0.27$

(c) $t = 0.44$ (d) $t = 0.59$

Figure 4.21: **CMOF**: Freezing droplet on cold substrate: liquid, gas, ice interfaces over time.
The substrate temperature is maintained at $-6°$C.
The Latent Heat is 3.34E+9 erg/g.
Grid size is 128×128.



(a) $t = 0$

(b) $t = 0.28$

(c) $t = 0.44$

(d) $t = 0.56$

Figure 4.22: **PLSMOF**: Freezing droplet on cold substrate: liquid, gas, ice interfaces over time.
The substrate temperature is maintained at $-6°$C.
The Latent Heat is 3.34E+9 erg/g.
Grid size is 128×128.

4.6 Deformation of Spherical Ullage Due to Impinging Jet

In this test we report results for the tank pressure control experiment (TPCE)[11, 10], in which a liquid jet impinges on a spherical vapor ullage in a microgravity environment. In the TPCE experimental results [11, 10], the following behaviors were found: If the jet strength exceeded the Weber number threshold $\mathrm{We}_j > 14$, then the jet would penetrate the bubble. If the Weber number was between the values $1.5 \leq \mathrm{We}_j \leq 14$, then bubble may become asymmetric. And if the Weber was below the threshold $\mathrm{We}_j \leq 1.5$, then the vapor bubble would be propelled to the top of the tank without any breakup.

The Weber number We_j is defined as

$$\mathrm{We}_j = \frac{\rho_l V_0^2 R_0^2}{\sigma D_j}. \tag{4.9}$$

ρ_l is the density of the liquid, V_0 is the velocity of the liquid jet at the nozzle, R_0 is the nozzle radius, σ is the coefficient of surface tension for the liquid/vapor interface, and D_j is the diameter of the jet where it impinges the vapor region.

Tests for $\mathrm{We}_j = 4.875$ and $\mathrm{We}_j = 5.25$ are shown below. The $\mathrm{We}_j = 4.875$ case shows the bubble being deformed and pushed to the top of the tank without breakup (Figures 4.23 and 4.24). And the $\mathrm{We}_j = 5.25$ case shows the impinging jet piercing the bubble (Figures 4.25, 4.26 and 4.27).

Figure 4.28 shows a 2D cross-section comparison of the MOF, CMOF and PLSMOF reconstructed interfaces after the bubble breakup for the $\mathrm{We}_j = 5.25$ test (at time $t = 15.5$) and figure 4.29 shows the associated level sets.

Additional figures are shown in appendix B.3 and B.4 for the PLSMOF tests, exhibiting particles, for $\mathrm{We}_j = 4.875$ and $\mathrm{We}_j = 5.25$, respectively.

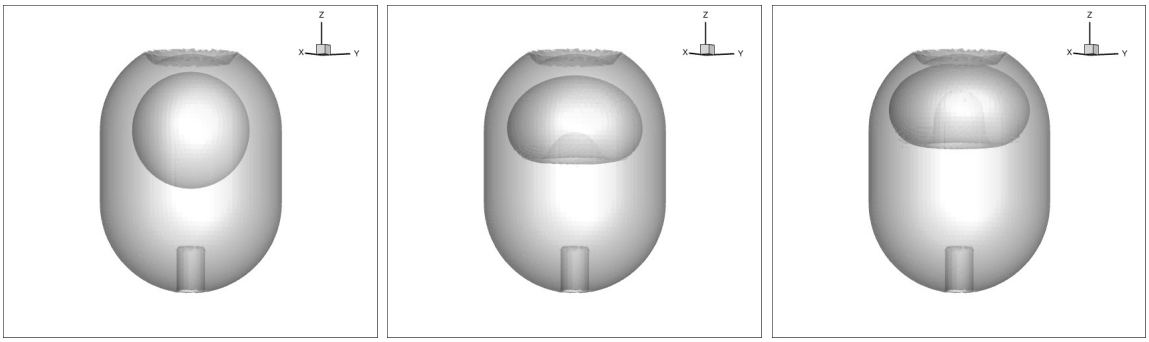


Figure 4.23: **CMOF**, deformation of a spherical ullage due to a liquid jet, $We_j = 4.875$. Times $t = 0.0$, 8.45, and 15.7 shown. The jet does not penetrate the ullage. Gridsize $64 \times 64 \times 64$.

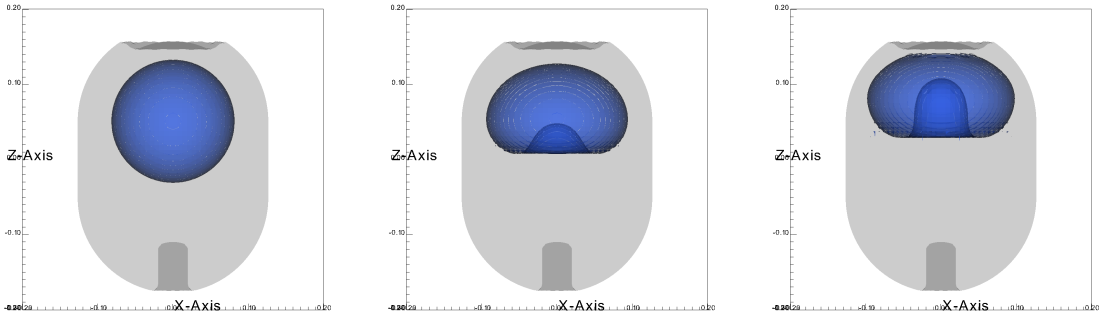


Figure 4.24: **PLSMOF**, deformation of a spherical ullage due to a liquid jet, $We_j = 4.875$. Times $t = 0.0$, 8.45, and 15.68 shown. The jet does not penetrate the ullage. Gridsize $64 \times 64 \times 64$.

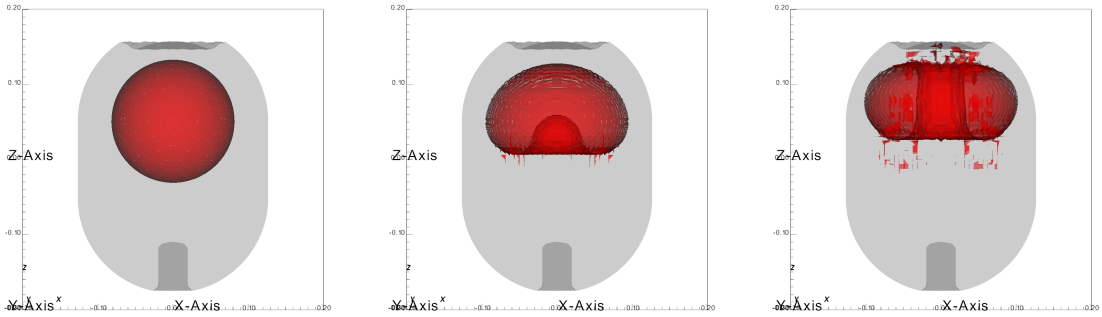


Figure 4.25: **MOF**, deformation of a spherical ullage due to a liquid jet, $We_j = 5.25$. Times $t = 0.0$, 7.84, and 15.51 shown. The jet penetrates the ullage. Grid size $64 \times 64 \times 64$.

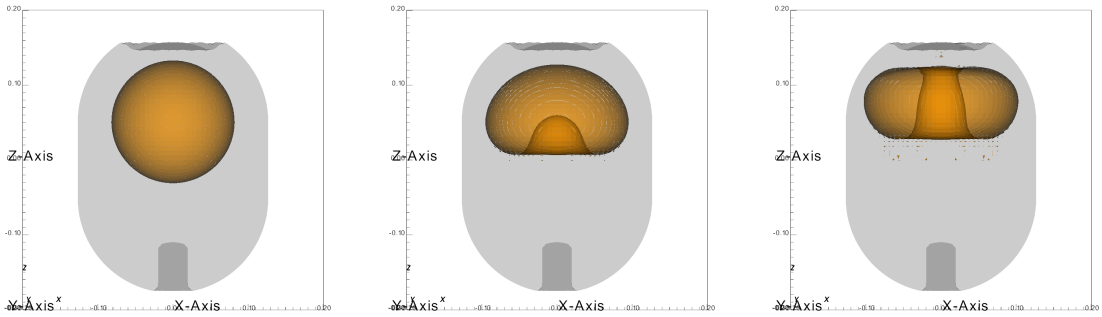


Figure 4.26: **CMOF**, deformation of a spherical ullage due to a liquid jet, $We_j = 5.25$. Times $t = 0.0$, 7.84, and 15.83 shown. The jet penetrates the ullage. Grid size $64 \times 64 \times 64$.

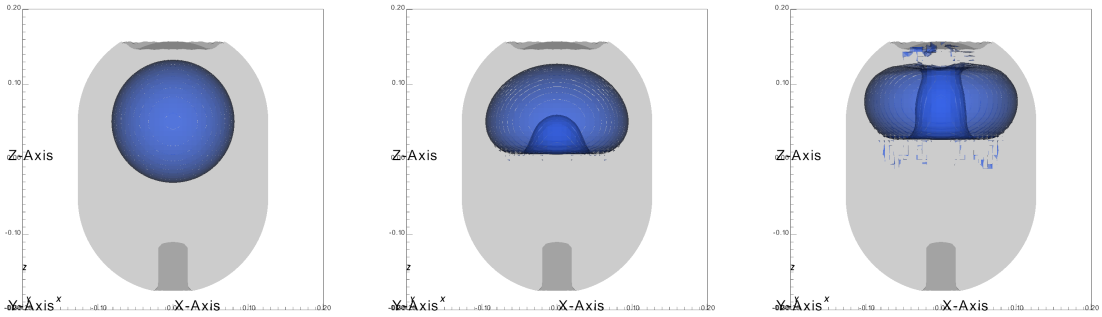


Figure 4.27: **PLSMOF**, deformation of a spherical ullage due to a liquid jet, $We_j = 5.25$. Times $t = 0.0$, 7.84, and 15.60 shown. The jet penetrates the ullage. Grid size $64 \times 64 \times 64$. Note: the reconstructed interface produces a notable amount of flotsam when the jet pierces the bubble, as the vapor interface is rapidly separated and small escaped volume fractions are launched away from the 'core' of the interface.

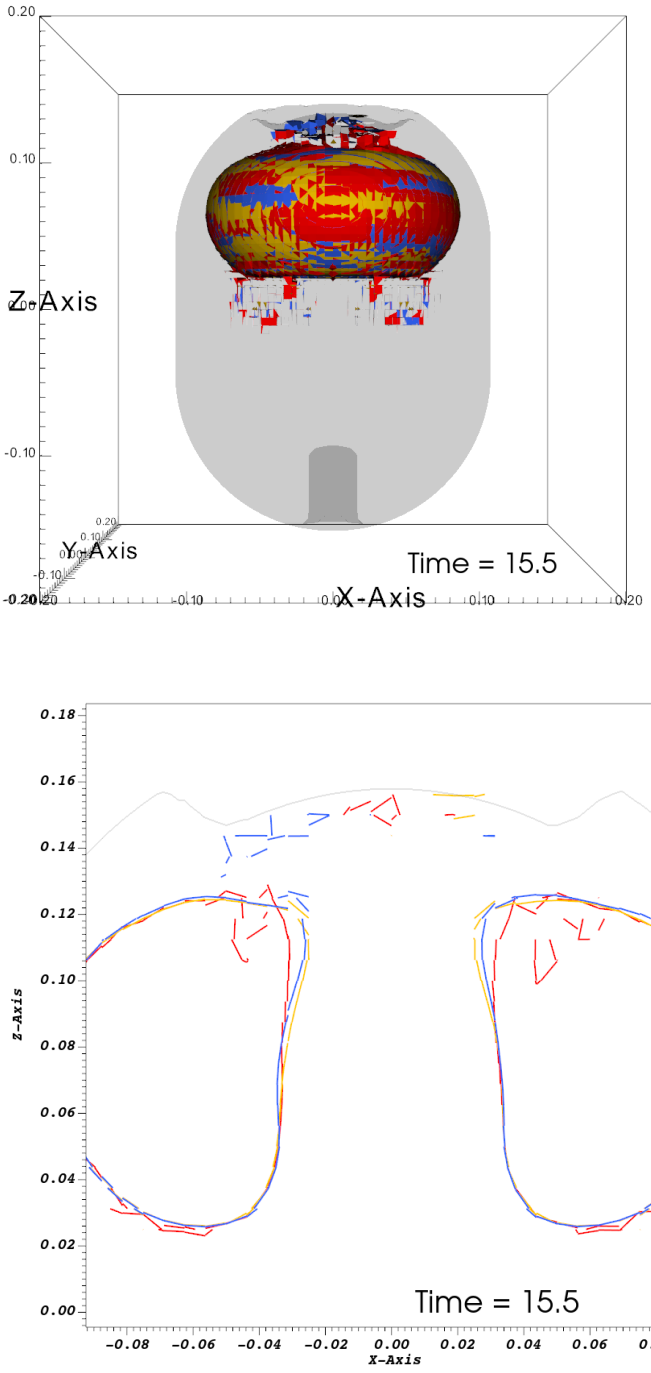


Figure 4.28: Comparison of vapor/liquid *reconstructed interface* at $t = 15.5$ after break-up of the spherical ullage due to an impinging jet ($We_j = 5.25$).
MOF (red), CMOF (orange), PLSMOF (blue). Top: 3D view, Bottom: 2D slice at $y = 0$.
Grid size $64 \times 64 \times 64$.

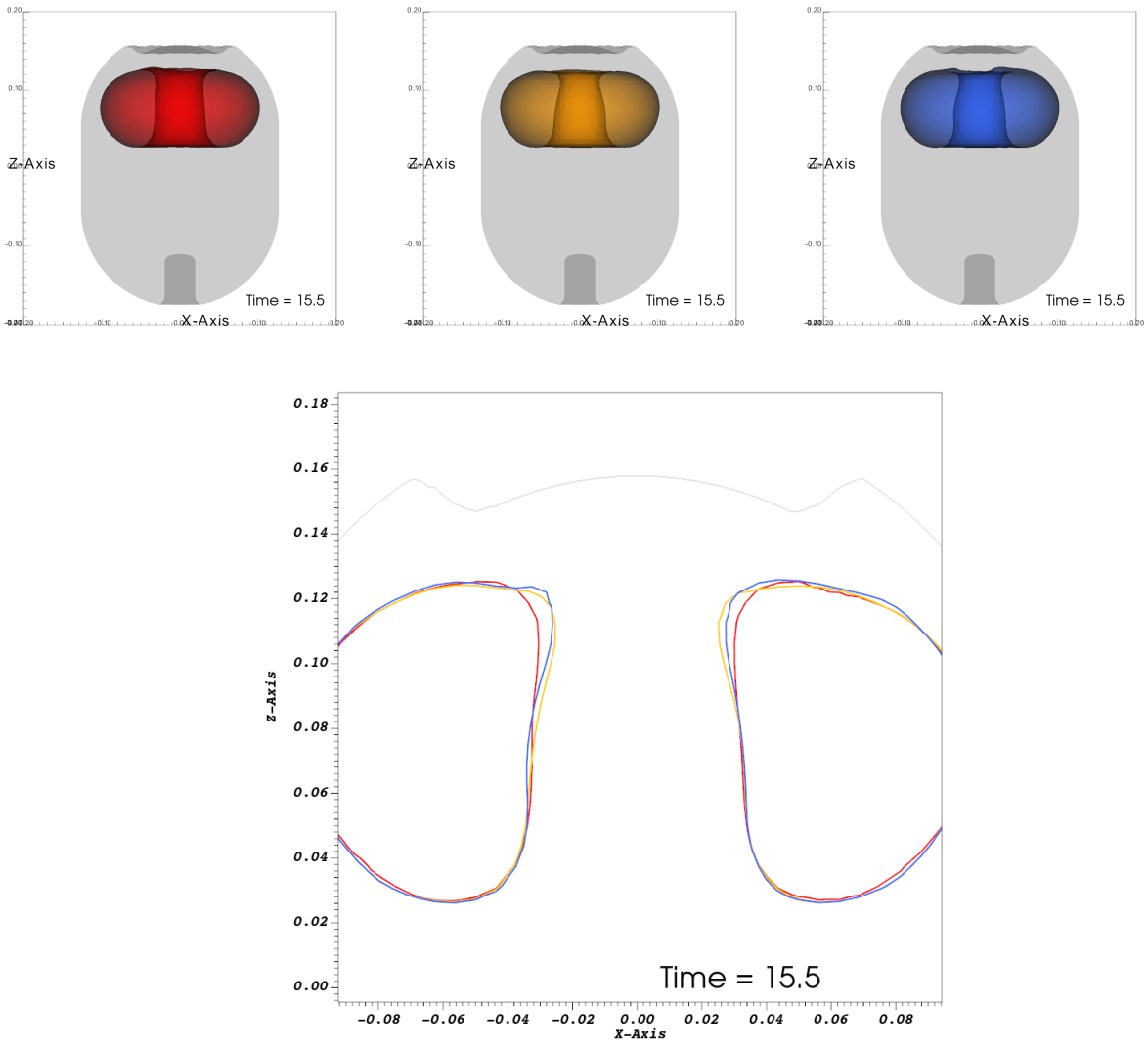


Figure 4.29: Comparison of vapor/liquid *level set interface* at $t = 15.5$, c.f. Figure 4.28. MOF (red), CMOF (orange), PLSMOF (blue). Top: 3D view, bottom: 2D slice at $y = 0$.

CHAPTER 5

CONCLUSIONS

In this dissertation, a new method has been presented for reconstructing the moment-of-fluid slope with the use of particles about the interface. It has been shown to work on N-phase flow problems and is volume preserving. The use of particles in conjunction with the level set has not been previously done in the context of three or more different phases. The novel particle level set moment of fluid (PLSMOF) method has been compared with both the level set moment of fluid (MOF) and the continuous moment of fluid (CMOF) methods on benchmark tests for rigid body motion and N-phase fluid flow involving phase change.

From the numerical experiments it has been shown that while MOF shows better results CMOF and PLSMOF on rigid body motion, on N-material fluid dynamics problems on the other hand, especially in conditions where surface tension effects dominate such as for low Weber numbers, the MOF method may not be as suitable due to the locality of its construction. This locality prevents the reconstruction from being able to 'see' the surface tension effects between neighboring cells, therefore it will not inherently damp out surface waves. In the presence of noise, this is seen by the checkerboard instability that persists. CMOF and PLSMOF remove this locality by the use of a supercell or supporting particles, respectively. Both CMOF and PLSMOF are demonstrated to remove the checkerboard instabilities and account for the 'damping out' behavior of the surface tension forces and may be more suitable in such cases. PLSMOF shows similar results to CMOF in all tests (with the added benefit of being generalizable to unstructured meshes). However in terms of performance, CMOF is more expensive than MOF, while PLSMOF about matches that of MOF. So in replacing the MOF interface calculation with a particle augmented routine, PLSMOF offers an improvement over MOF in that it is able to avoid checkerboard instability during the interface reconstruction without impacting performance.

Future work includes exploring the effects of adjusting PLSMOF particle weights and particle seeding/deletion criterion on diffusive behavior (as the locality can be adjusted) as well as the impact on performance.

APPENDIX A

VOF HEIGHT FUNCTION TECHNIQUE

Below is described the VOF height function technique [38] for approximating the curvature, implemented as in [97]. This method produces a second order accurate approximation to the interfacial curvature between two materials directly from the piecewise linear reconstruction.

The orientation of the free surface is first determined from $\boldsymbol{n} = \nabla\phi/|\nabla\phi|$. A 7×3 stencil is then constructed about cell (i, j), oriented in the direction of the largest component of $\boldsymbol{n}$ (i.e. (i, j) s.t. $|n_i| > |n_j|$). We then obtain three vertical sums, $F_{i'}, i' = i-1, i, i+1$, defined as the exact integrals $F_i = \int_{x_{i-1/2}}^{x_{i+1/2}} h(x) + C(j)$, up to a constant, of height function $h(x)$ for each column of the stencil. The height function $h(x)$ is defined as the distance between the end of the stencil within the material, up to the piecewise linear reconstructed interface (the original VOF height function technique instead constructs h from the vertical sum of the volume fractions). We approximate the first and second derivatives of the height function using a second order central distance discretization:

$$h'(x_i) = \frac{F_{i+1} - F_{i-1}}{\Delta x}, \qquad h''(x_i) = \frac{F_{i+1} - 2F_i + F_{i-1}}{\Delta x^2} \tag{A.1}$$

From this, we then can get the second order approximation to the curvature.

$$\kappa_{\mathrm{i,j}} = \frac{h''(x_i)}{(1 + h'(x_i)^2)^{3/2}}. \tag{A.2}$$

Then, for the curvature of a cell (i, j), we then find the neighboring cell $(i+1, j)$ for which the level set function changes sign. If the interface is aligned vertically, we have

$$\kappa_I = \begin{cases} \kappa_{\mathrm{i,j}}, & \theta < 1/2 \\ \kappa_{\mathrm{i+1,j}}, & \text{otherwise} \end{cases} \tag{A.3}$$

If it is instead aligned horizontally, we have

$$\kappa_I = (1 - \theta)\kappa_{\mathrm{i,j}} + \theta\, \kappa_{\mathrm{i+1,j}} \tag{A.4}$$

Higher order discretizations for the height function can be found in [98]. In the context of PLSMOF, it is advantageous to use the height function technique for calculating the curvature as it is more robust in the presence of flotsam than the curvature obtained from the level set function. However, when at a triple point, the level set is instead used for the curvature calculations.

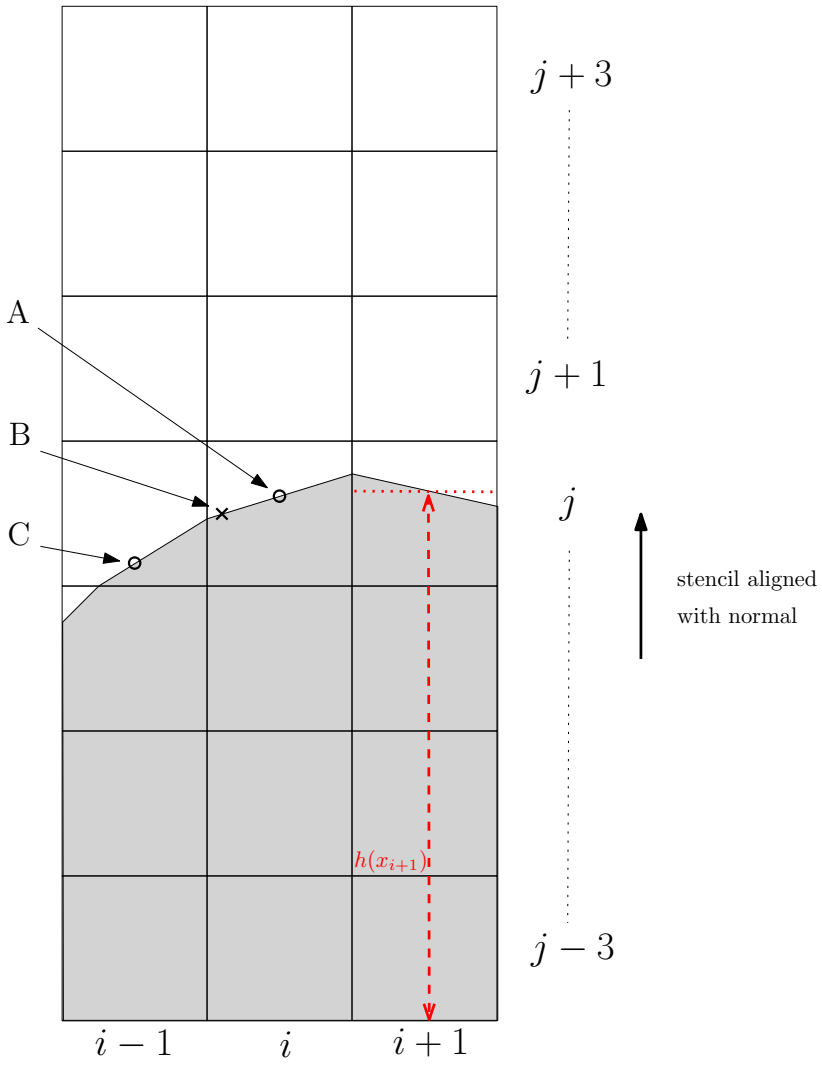


Figure A.1: 7×3 stencil for height function curvature calculation.
The red dashed line shows the height function for a column.
The interfacial curvature at 'B' is obtained by the linear interpolation of curvatures 'A' and 'C'.

APPENDIX B

NUMERICAL EXPERIMENTS, ADDITIONAL FIGURES

B.1 Zalesak's Disk, curvature refinement

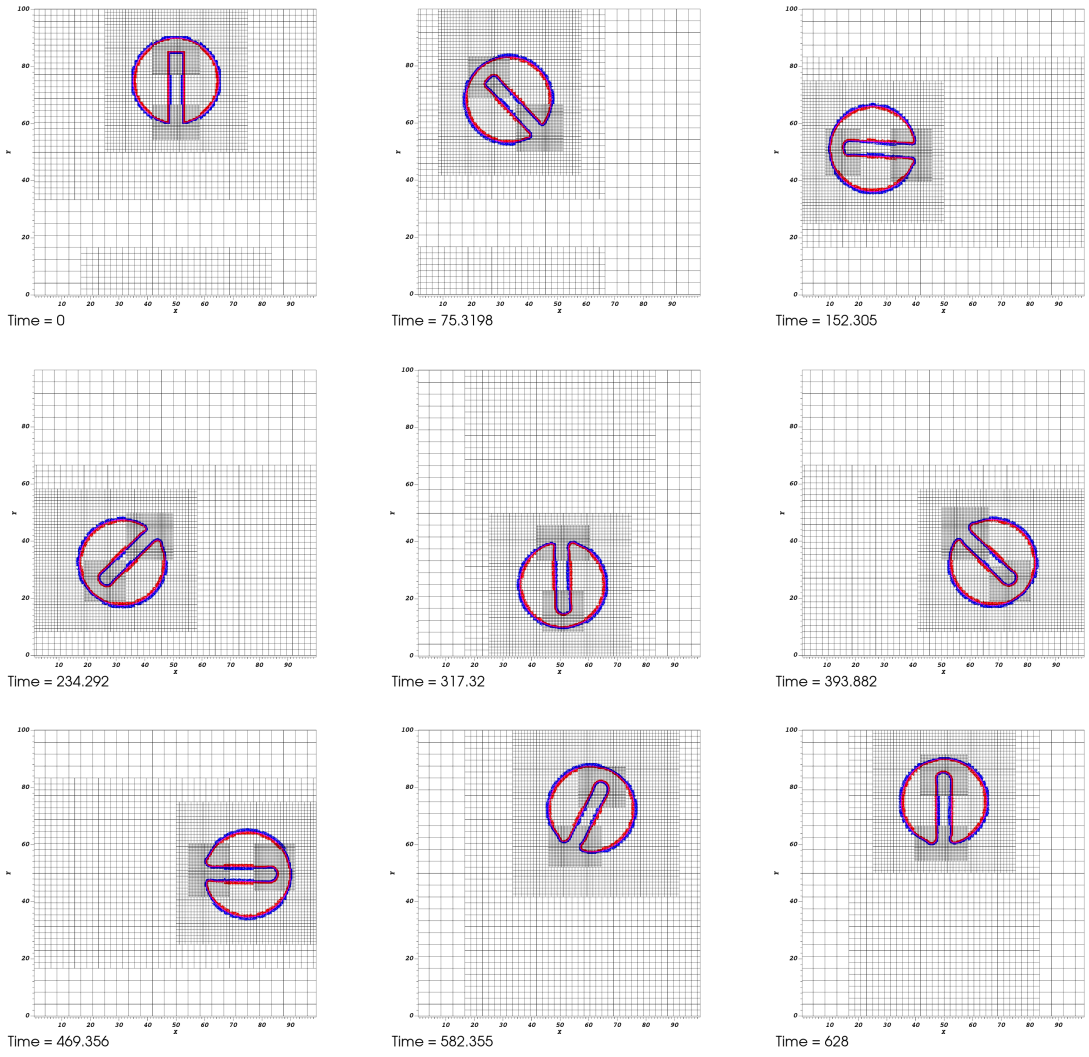


Figure B.1: **PLSMOF**, zalesak's disk, AMR curvature refinement (particles on finest grids) Comparison to tests with interface wholly contained on the finest level in Table 4.3.

B.2 Freezing droplet

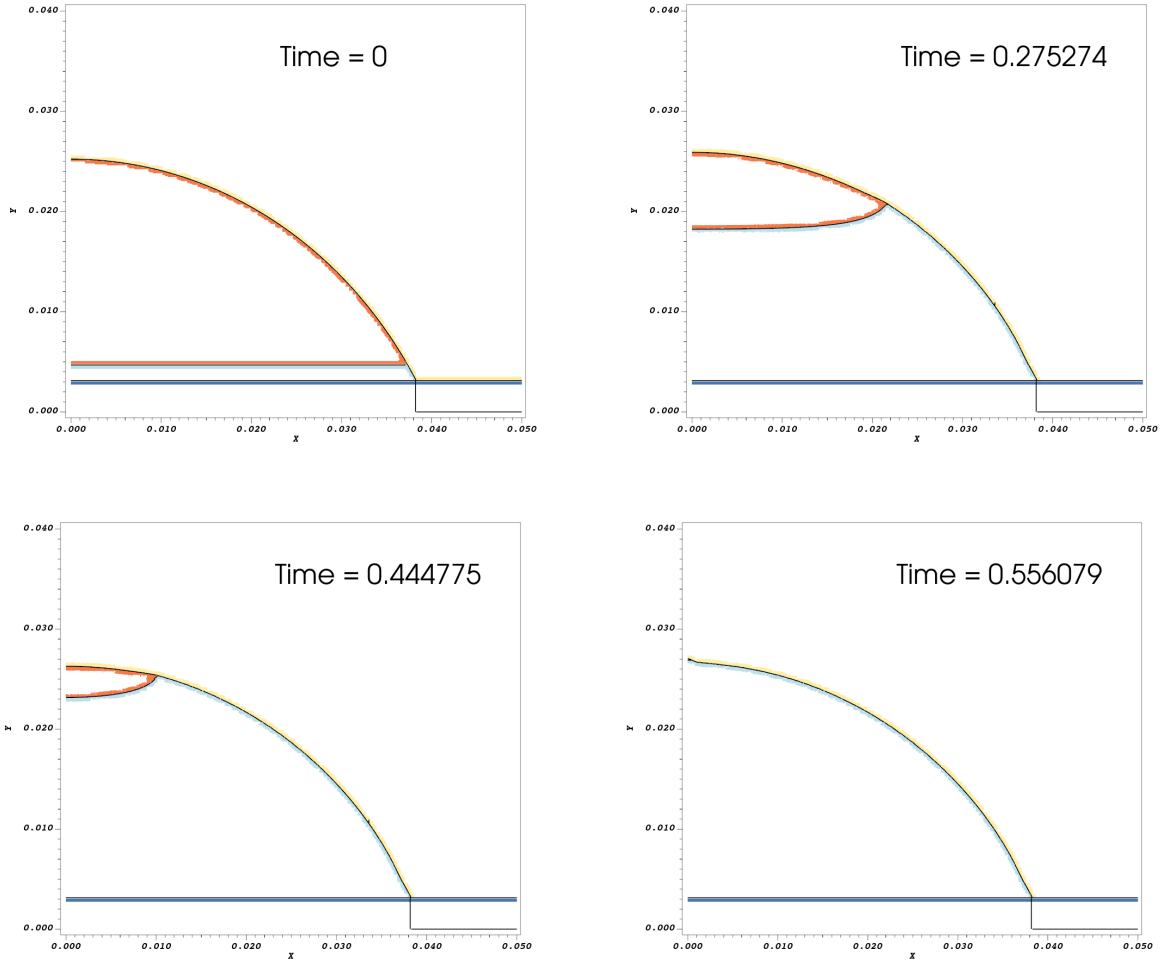


Figure B.2: **PLSMOF**, freezing liquid droplet on cold substrate (corresponding to the test shown in Figure 4.22)
Particles: yellow-air, orange-water, cyan-ice, blue-substrate

B.3 TPCE test, $\mathrm{We}_j = 4.875$

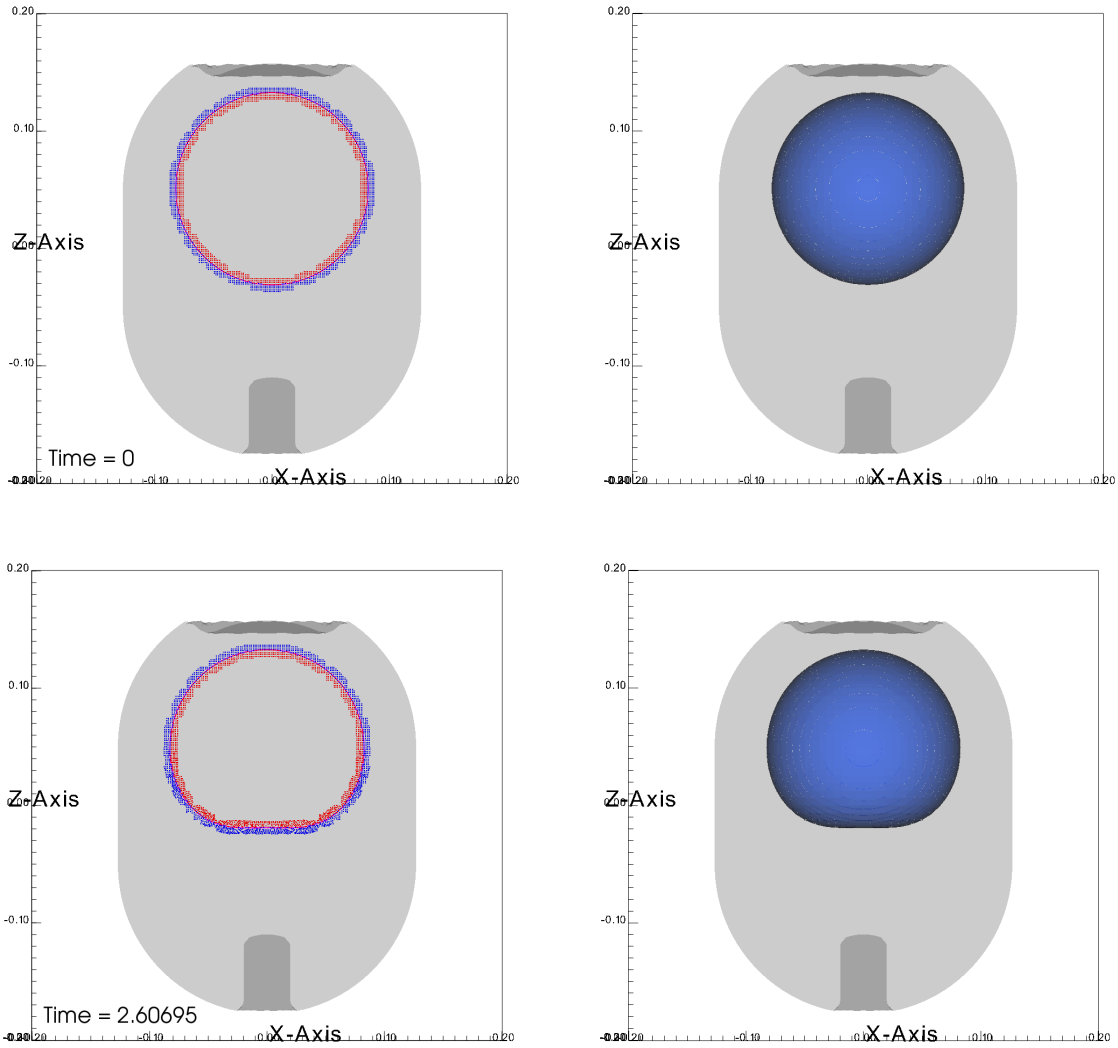


Figure B.3: **PLSMOF**, deformation of spherical ullage due to liquid jet, $\mathrm{We}_j = 4.875$.
LEFT: cross-section of particles shown for $-0.003 < y < 0.003$.
Red: vapor material particles, Blue: liquid material particles (tank particles not shown)
RIGHT: reconstructed vapor/liquid interface

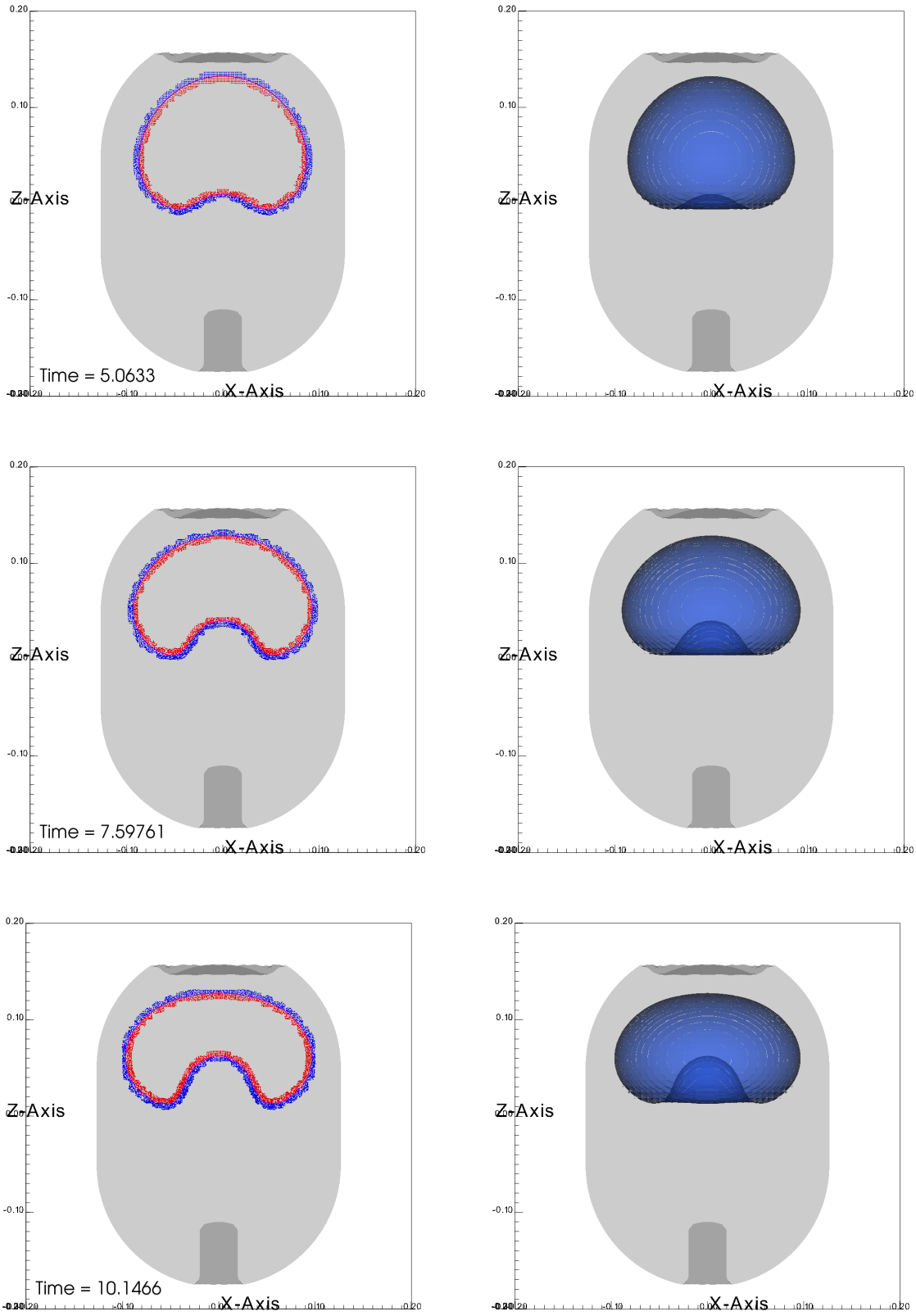


Figure B.4: **PLSMOF**, deformation of spherical ullage due to liquid jet, $We_j = 4.875$ (continued)

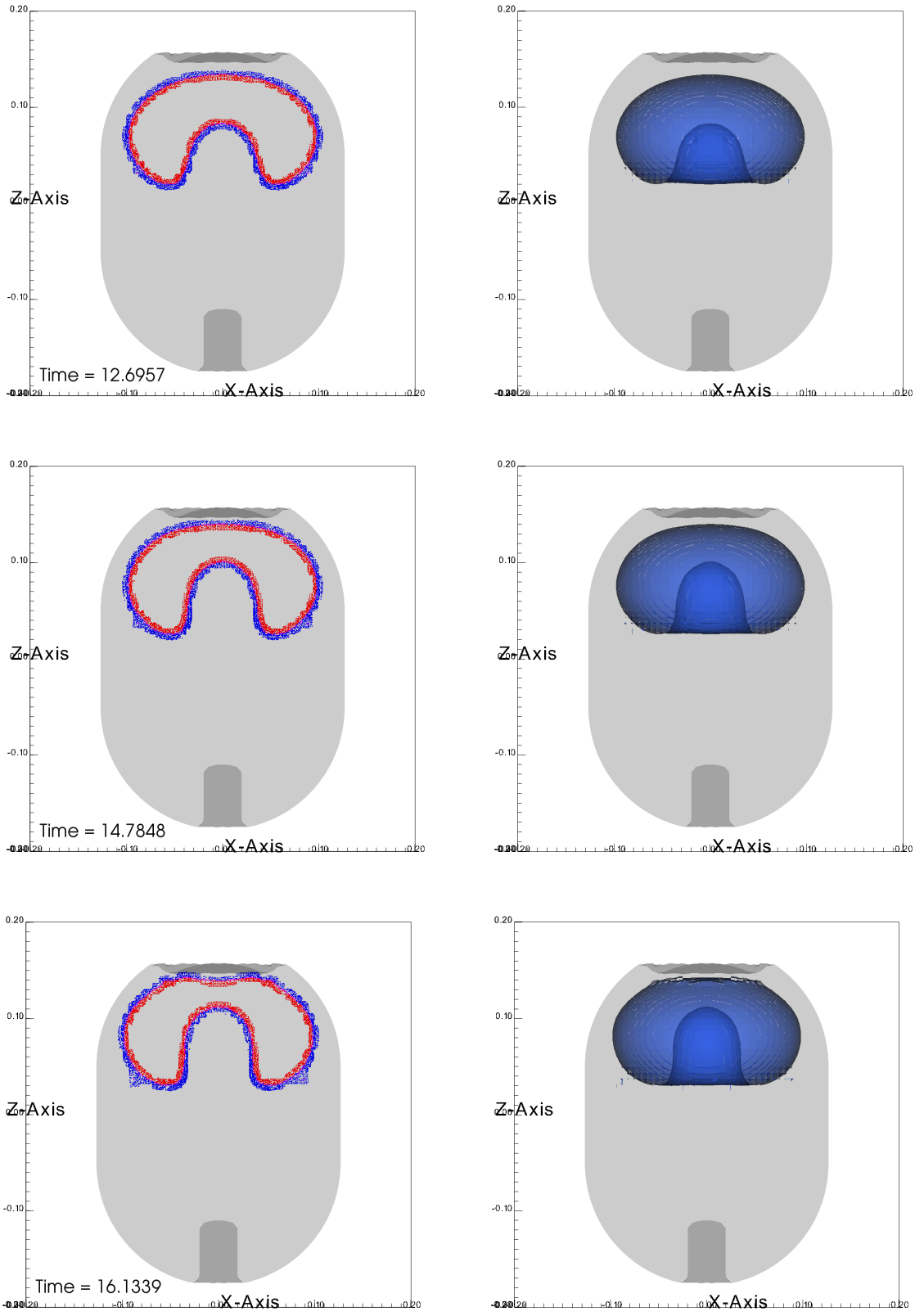


Figure B.5: **PLSMOF**, deformation of spherical ullage due to liquid jet $\mathrm{We}_j = 4.875$ (continued)

B.4 TPCE test, $\mathrm{We}_j = 5.25$

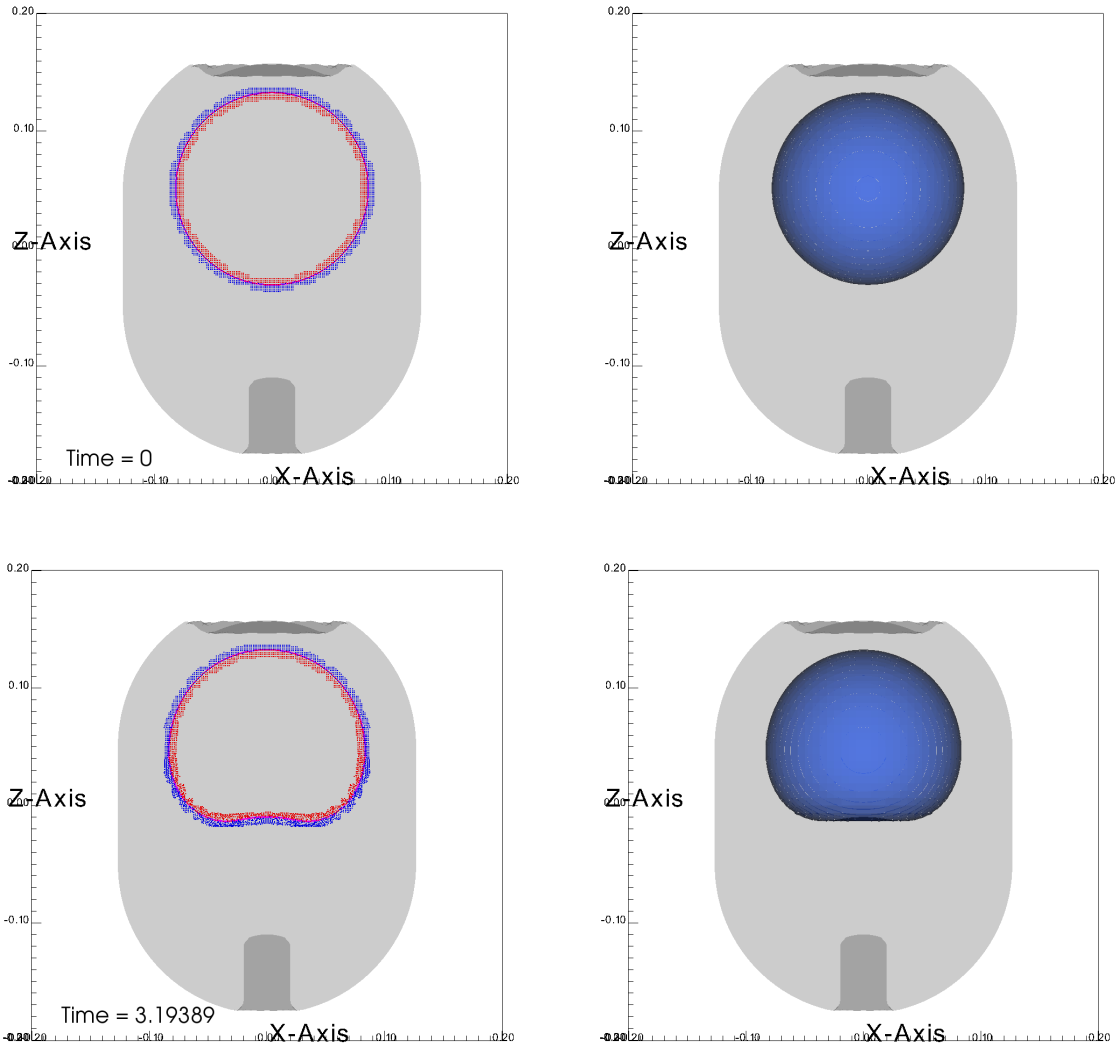


Figure B.6: **PLSMOF**, deformation of spherical ullage due to liquid jet, $\mathrm{We}_j = 5.25$.
LEFT: cross-section of particles shown for $-0.003 < y < 0.003$.
Red: vapor material particles, Blue: liquid material particles (tank particles not shown)
RIGHT: reconstructed vapor/liquid interface

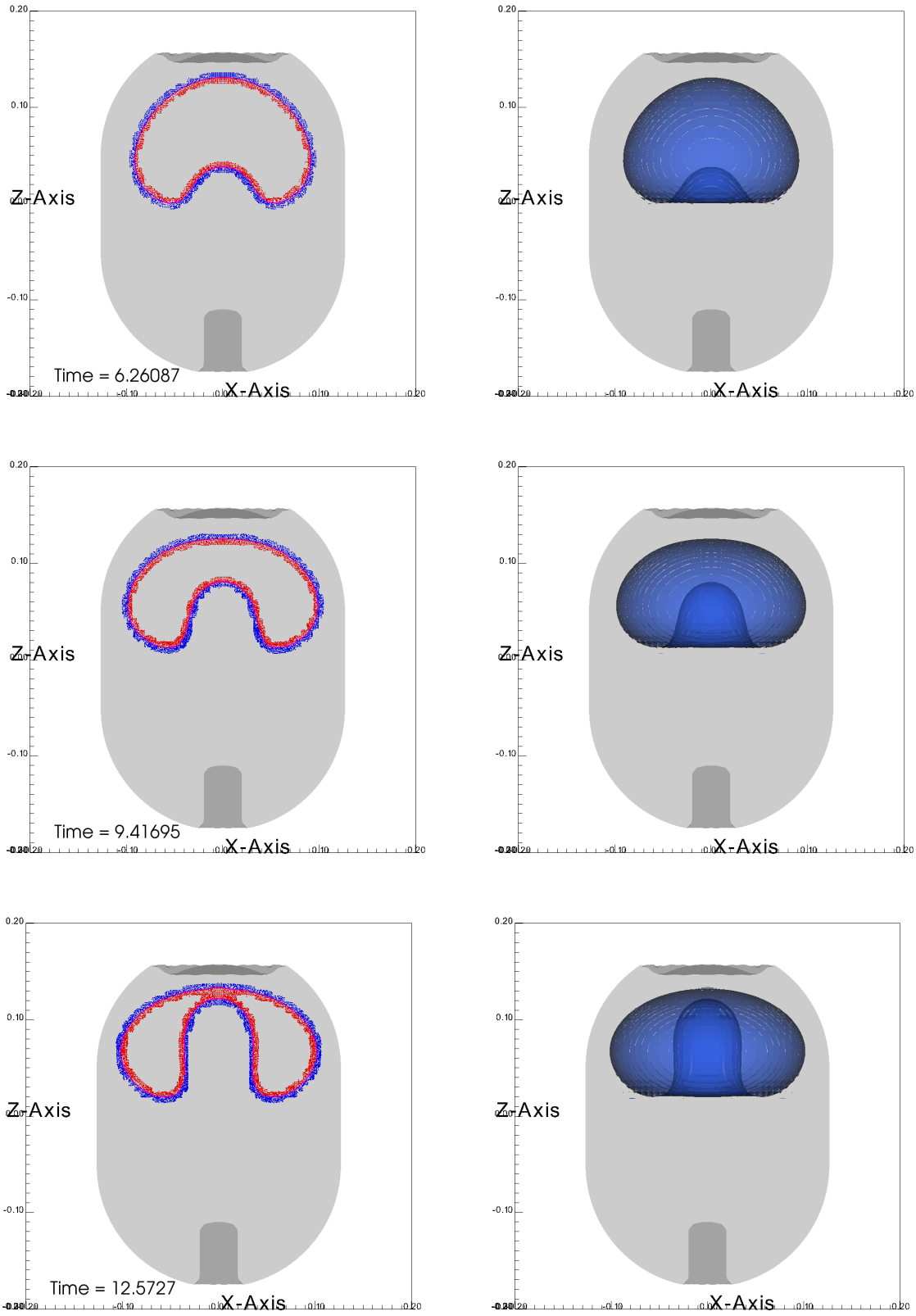


Figure B.7: **PLSMOF**, deformation of spherical ullage due to liquid jet, $We_j = 5.25$ (continued)

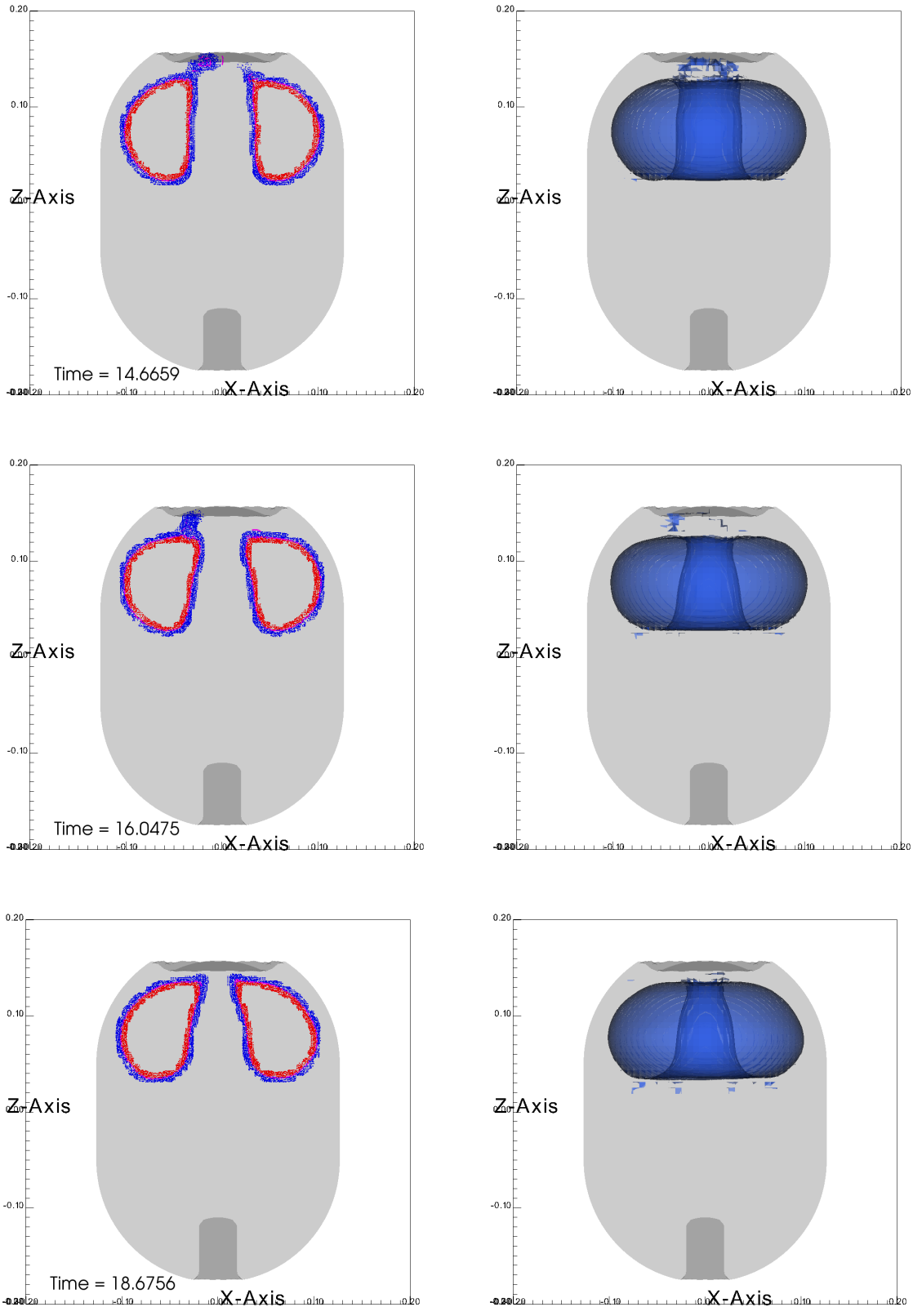


Figure B.8: **PLSMOF**, deformation of spherical ullage due to liquid jet, $We_j = 5.25$ (continued)

BIBLIOGRAPHY

[1] Adalsteinsson, D., and Sethian, J. A. A fast level set method for propagating interfaces. *Journal of computational physics 118*, 2 (1995), 269–277.

[2] Afkhami, S., and Bussmann, M. Height functions for applying contact angles to 2d vof simulations. *International Journal for Numerical Methods in Fluids 57*, 4 (2008), 453–472.

[3] Ahn, H. T., and Shashkov, M. Multi-material interface reconstruction on generalized polyhedral meshes. *Journal of Computational Physics 226*, 2 (2007), 2096–2132.

[4] 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 473* (2023), 111721.

[5] 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.

[6] 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.

[7] Aulisa, E., Manservisi, S., Scardovelli, R., and Zaleski, S. Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry. *Journal of Computational Physics 225*, 2 (2007), 2301–2319.

[8] Balcázar, N., Lehmkuhl, O., Rigola, J., and Oliva, A. A multiple marker level-set method for simulation of deformable fluid particles. *International Journal of Multiphase Flow 74* (2015), 125 – 142.

[9] Barlow, A. J., Maire, P.-H., Rider, W. J., Rieben, R. N., and Shashkov, M. J. Arbitrary lagrangian–eulerian methods for modeling high-speed compressible multimaterial flows. *Journal of Computational Physics 322* (2016), 603–665.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Bukač, Martina, Muha, Boris, and Salgado, Abner J. Analysis of a diffuse interface method for the stokes-darcy coupled problem. *ESAIM: M2AN 57*, 5 (2023), 2623–2658.

[15] Burstedde, C., Calhoun, D., Mandli, K. T., and Terrel, A. R. Forestclaw: Hybrid forest-of-octrees AMR for hyperbolic conservation laws. In *Parallel Computing: Accelerating Computational Science and Engineering (CSE)* (2014), M. Bader, A. Bode, H.-J. Bungartz, M. Gerndt, G. R. Joubert, and F. Peters, Eds., vol. 25 of *Advances in Parallel Computing*, IOS Press, pp. 253–262.

[16] 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.

[17] Can, E., and Prosperetti, A. A level set method for vapor bubble dynamics. *Journal of Computational Physics 231*, 4 (2012), 1533–1552.

[18] Chen, S., Johnson, D. B., Raad, P. E., and Fadda, D. The surface marker and micro cell method. *International Journal for Numerical Methods in Fluids 25*, 7 (1997), 749–778.

[19] Chopp, D. L. Computing minimal surfaces via level set curvature flow. *Journal of Computational Physics 106*, 1 (1993), 77–91.

[20] Chorin, A. J. Numerical solution of the navier-stokes equations. *Mathematics of computation 22*, 104 (1968), 745–762.

[21] Colella, P., Graves, D. T., Ligocki, T., Martin, D., Modiano, D., Serafini, D., and Van Straalen, B. Chombo software package for amr applications design document. *Available at the Chombo website: http://seesar. lbl. gov/ANAG/chombo/(September 2008) 2* (2009).

[22] Coquerelle, M., and Glockner, S. A fourth-order accurate curvature computation in a level set framework for two-phase flows subjected to surface tension forces. *Journal of Computational Physics 305* (2016), 838–876.

[23] Cummins, S. J., Francois, M. M., and Kothe, D. B. Estimating curvature from volume fractions. *Computers & Structures 83*, 6 (2005), 425–434. Frontier of Multi-Phase Flow Analysis and Fluid-Structure.

[24] De Gennes, P.-G., Brochard-Wyart, F., and Quéré, D. *Capillarity and wetting phenomena: drops, bubbles, pearls, waves*. Springer Science & Business Media, 2013.

[25] Denner, F., Evrard, F., Serfaty, R., and van Wachem, B. G. Artificial viscosity model to mitigate numerical artefacts at fluid interfaces with surface tension. *Computers & Fluids 143* (2017), 59–72.

[26] Dyadechko, V., and Shashkov, M. Moment-of-fluid interface reconstruction. *Los Alamos report LA-UR-05-7571* (2005).

[27] Dyadechko, V., and Shashkov, M. Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics 227*, 11 (2008), 5361–5384.

[28] 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.

[29] Enright, D., Losasso, F., and Fedkiw, R. A fast and accurate semi-lagrangian particle level set method. *Computers & Structures 83*, 6 (2005), 479 – 490. Frontier of Multi-Phase Flow Analysis and Fluid-Structure.

[30] Fedkiw, R. P., Aslam, T., Merriman, B., and Osher, S. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of computational physics 152*, 2 (1999), 457–492.

[31] Flynn, S., Egbert, P., Holladay, S., and Oborn, J. Adaptive fluid simulation using a linear octree structure. In *Proceedings of Computer Graphics International 2018* (New York, NY, USA, 2018), CGI 2018, Association for Computing Machinery, p. 217–222.

[32] 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.

[33] Fröde, F., Grenga, T., Le Chenadec, V., Bode, M., and Pitsch, H. A three-dimensional cell-based volume-of-fluid method for conservative simulations of primary atomization. *Journal of Computational Physics 465* (2022), 111374.

[34] Gibou, F., Chen, L., Nguyen, D., and Banerjee, S. A level set based sharp interface method for the multiphase incompressible navier–stokes equations with phase change. *Journal of Computational Physics 222*, 2 (2007), 536–555.

[35] 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.

[36] Guittet, A., Theillard, M., and Gibou, F. A stable projection method for the incompressible navier–stokes equations on arbitrary geometries and adaptive quad/octrees. *Journal of Computational Physics 292* (2015), 215–238.

[37] Harlow, F. H., and Welch, J. E. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *The Physics of Fluids 8*, 12 (12 1965), 2182–2189.

[38] Helmsen, J., Colella, P., and Puckett, E. G. Non-convex profile evolution in two dimensions using volume of fluids. *LBNL Technical Report LBNL-40693* (6 1997).

[39] Henri, F., Coquerelle, M., and Lubin, P. Geometrical level set reinitialization using closest point method and kink detection for thin filaments, topology changes and two-phase flows. *Journal of Computational Physics 448* (2022), 110704.

[40] Hieber, S. E., and Koumoutsakos, P. A lagrangian particle level set method. *Journal of Computational Physics 210*, 1 (2005), 342–367.

[41] Hirt, C., and Nichols, B. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics 39*, 1 (1981), 201–225.

[42] 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.

[43] Huang, M., Chen, B., and Wu, L. A slic–vof method based on unstructured grid. *Microgravity Science and Technology 22* (2010), 305–314.

[44] 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.

[45] 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.

[46] Juric, D., and Tryggvason, G. A front-tracking method for dendritic solidification. *Journal of Computational Physics 123*, 1 (1996), 127–148.

[47] Karnakov, P., Litvinov, S., and Koumoutsakos, P. A hybrid particle volume-of-fluid method for curvature estimation in multiphase flows. *International Journal of Multiphase Flow 125* (2020), 103209.

[48] Kassemi, M., Kartuzova, O., and Hylton, S. Validation of two-phase cfd models for propellant tank self-pressurization: Crossing fluid types, scales, and gravity levels. *Cryogenics 89* (2018), 1–15.

[49] Kim, J. Phase field computations for ternary fluid flows. *Computer methods in applied mechanics and engineering 196*, 45 (2007), 4779–4788.

[50] 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.

[51] Le Chenadec, V., and Pitsch, H. A 3d unsplit forward/backward volume-of-fluid approach and coupling to the level set method. *Journal of Computational Physics 233* (2013), 10–33.

[52] Le Chenadec, V., and Pitsch, H. A monotonicity preserving conservative sharp interface flow solver for high density ratio two-phase flows. *Journal of Computational Physics 249* (2013), 185–203.

[53] Lemoine, A., Glockner, S., and Breil, J. Moment-of-fluid analytic reconstruction on 2d cartesian grids. *Journal of Computational Physics 328* (2017), 131–139.

[54] 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.

[55] Liu, H., Krishnan, S., Marella, S., and Udaykumar, H. Sharp interface cartesian grid method ii: A technique for simulating droplet interactions with surfaces of arbitrary shape. *Journal of Computational Physics 210*, 1 (2005), 32–54.

[56] 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.

[57] Loubère, R., Maire, P.-H., Shashkov, M., Breil, J., and Galera, S. Reale: A reconnection-based arbitrary-lagrangian–eulerian method. *Journal of Computational Physics 229*, 12 (2010), 4724–4761.

[58] 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.

[59] Mauch, S. A fast algorithm for computing the closest point and distance transform. *Go online to http://www. acm. caltech. edu/seanm/software/cpt/cpt. pdf 2* (2000).

[60] Meagher, D. Geometric modeling using octree encoding. *Computer Graphics and Image Processing 19*, 2 (1982), 129–147.

[61] Mihalef, V., Metaxas, D., and Sussman, M. Textured liquids based on the marker level set. *Computer Graphics Forum 26*, 3 (2007), 457–466.

[62] Milcent, T., and Lemoine, A. Moment-of-fluid analytic reconstruction on 3d rectangular hexahedrons. *Journal of Computational Physics 409* (2020), 109346.

[63] Mirjalili, S., Ivey, C. B., and Mani, A. Comparison between the diffuse interface and volume of fluid methods for simulating two-phase flows. *International Journal of Multiphase Flow 116* (2019), 221–238.

[64] Mirjalili, S., and Mani, A. A conservative second order phase field model for simulation of n-phase flows. *Journal of Computational Physics 498* (2024), 112657.

[65] Mukundan, A. A., Ménard, T., de Motta, J. C. B., and Berlemont, A. A 3d moment of fluid method for simulating complex turbulent multiphase flows. *Computers & Fluids 198* (2020), 104364.

[66] Noh, W., and Woodward, P. Simple line interface method. In *A. van de Vooren and P. Zandbergen, editors, Proceedings of the 5th international conference on fluid dynamics, Lecture notes in Physics* (1976), vol. 59.

[67] Ohta, M., Kikuchi, D., Yoshida, Y., and Sussman, M. Direct numerical simulation of the slow formation process of single bubbles in a viscous liquid. *JOURNAL OF CHEMICAL ENGINEERING OF JAPAN 40*, 11 (2007), 939–943.

[68] Osher, S., Fedkiw, R., and Piechor, K. Level set methods and dynamic implicit surfaces. *Appl. Mech. Rev. 57*, 3 (2004), B15–B15.

[69] 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.

[70] Palmore, J., and Desjardins, O. A volume of fluid framework for interface-resolved simulations of vaporizing liquid-gas flows. *Journal of Computational Physics 399* (2019), 108954.

[71] 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.

[72] 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.

[73] Peskin, C. S. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics 10*, 2 (1972), 252–271.

[74] Peskin, C. S. The immersed boundary method. *Acta Numerica 11* (2002), 479–517.

[75] Pilliod Jr, J. E., and Puckett, E. G. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics 199*, 2 (2004), 465–502.

[76] Popinet, S. Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries. *Journal of Computational Physics 190*, 2 (2003), 572–600.

[77] Popinet, S. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics 228*, 16 (2009), 5838–5866.

[78] Qian, L., Wei, Y., and Xiao, F. Coupled thinc and level set method: A conservative interface capturing scheme with high-order surface representations. *Journal of Computational Physics 373* (2018), 284–303.

[79] Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics 378* (2019), 686–707.

[80] Rueda Villegas, L., Alis, R., Lepilliez, M., and Tanguy, S. A ghost fluid/level set method for boiling flows and liquid evaporation: Application to the leidenfrost effect. *Journal of Computational Physics 316* (2016), 789–813.

[81] Saye, R. High-order methods for computing distances to implicitly defined surfaces. *Communications in Applied Mathematics and Computational Science 9*, 1 (2014), 107–141.

[82] Scardovelli, R., and Zaleski, S. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics 31*, 1 (1999), 567–603.

[83] Scardovelli, R., and Zaleski, S. Interface reconstruction with least-square fit and split eulerian–lagrangian advection. *International Journal for Numerical Methods in Fluids 41*, 3 (2003), 251–274.

[84] 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.

[85] 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.

[86] 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.

[87] Schrage, R. W. *A Theoretical Study of Interphase Mass Transfer*. Columbia University Press, 1953.

[88] Sethian, J. A. Fast marching methods. *SIAM review 41*, 2 (1999), 199–235.

[89] Shetabivash, H., Dolatabadi, A., and Paraschivoiu, M. A multiple level-set approach for modelling containerless freezing process. *Journal of Computational Physics 415* (2020), 109527.

[90] 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.

[91] 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.

[92] Son, G., Dhir, V., and Ramanujapu, N. Dynamics and heat transfer associated with a single bubble during nucleate boiling on a horizontal surface. *Journal of Heat Transfer-transactions of The Asme 121* (1999), 623–631.

[93] 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.

[94] Stewart, M. *Pressurization of a Flightweight, Liquid Hydrogen Tank: Evaporation & Condensation at the Liquid/Vapor Interface.*

[95] Strain, J. Semi-lagrangian methods for level set equations. *Journal of Computational Physics 151*, 2 (1999), 498–533.

[96] Strang, G. The finite element method and approximation theory. In *Numerical Solution of Partial Differential Equations–II*. Elsevier, 1971, pp. 547–583.

[97] 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.

[98] Sussman, M., and Ohta, M. High-order techniques for calculating surface tension forces. In *Free Boundary Problems* (Basel, 2007), I. N. Figueiredo, J. F. Rodrigues, and L. Santos, Eds., Birkhäuser Basel, pp. 425–434.

[99] 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.

[100] 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.

[101] Sussman, M., Smith, K. M., Hussaini, M. Y., Ohta, M., and Zhi-Wei, R. A sharp interface method for incompressible two-phase flows. *Journal of computational physics 221*, 2 (2007), 469–505.

[102] Tanguy, S., Sagan, M., Lalanne, B., Couderc, F., and Colin, C. Benchmarks and numerical methods for the simulation of boiling flows. *Journal of Computational Physics 264* (2014), 1–22.

[103] Teyssier, R. Cosmological hydrodynamics with adaptive mesh refinement-a new high resolution code called ramses. *Astronomy & Astrophysics 385*, 1 (2002), 337–364.

[104] 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.

[105] 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.

[106] 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.

[107] Wang, C., Wang, W., Pan, S., and Zhao, F. A local curvature based adaptive particle level set method. *Journal of Scientific Computing 91*, 1 (2022), 3.

[108] Wang, Y., and Li, J. Phase field modeling of defects and deformation. *Acta Materialia 58*, 4 (2010), 1212–1235.

[109] Wang, Y., Simakhina, S., and Sussman, M. A hybrid level set-volume constraint method for incompressible two-phase flow. *Journal of Computational Physics 231*, 19 (2012), 6438–6471.

[110] Wang, Z., Yang, J., and Stern, F. An improved particle correction procedure for the particle level set method. *Journal of Computational Physics 228*, 16 (2009), 5819 – 5837.

[111] Wang, Z., Zheng, X., Chryssostomidis, C., and Karniadakis, G. E. A phase-field method for boiling heat transfer. *Journal of Computational Physics 435* (2021), 110239.

[112] 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.

[113] Weymouth, G., and Yue, D. K.-P. Conservative volume-of-fluid method for free-surface simulations on cartesian-grids. *Journal of Computational Physics 229*, 8 (2010), 2853–2865.

[114] Wissink, A. M., Hornung, R. D., Kohn, S. R., Smith, S. S., and Elliott, N. Large scale parallel structured amr calculations using the samrai framework. In *Proceedings of the 2001 ACM/IEEE Conference on Supercomputing* (New York, NY, USA, 2001), SC '01, Association for Computing Machinery, p. 6.

[115] Yang, X., James, A. J., Lowengrub, J., Zheng, X., and Cristini, V. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics 217*, 2 (2006), 364–394.

[116] Ye, Z., Estebe, C., Liu, Y., Vahab, M., Huang, Z., Sussman, M., Moradikazerouni, A., Shoele, K., Lian, Y., Ohta, M., and Hussaini, M. Y. An Improved Coupled Level Set and Continuous Moment-of-Fluid Method for Simulating Multiphase Flows with Phase Change. *Communications on Applied Mathematics and Computation* (Aug. 2023).

[117] Youngs, D. L. Time-dependent multi-material flow with large fluid distortion. *Numerical methods for fluid dynamics* (1982).

[118] Zalesak, S. T. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of computational physics 31*, 3 (1979), 335–362.

[119] Zhang, W., Almgren, A., Beckner, V., Bell, J., Blaschke, J., Chan, C., Day, M., Friesen, B., Gott, K., Graves, D., et al. Amrex: a framework for block-structured adaptive mesh refinement. *Journal of Open Source Software 4*, 37 (2019), 1370–1370.

[120] Zhang, X., Chen, J.-S., and Osher, S. A multiple level set method for modeling grain boundary evolution of polycrystalline materials. *Interaction and multiscale mechanics 1*, 2 (2008), 178–191.

[121] Zhao, H. A fast sweeping method for eikonal equations. *Mathematics of computation 74*, 250 (2005), 603–627.

[122] Zhao, L., Khuc, H., Mao, J., Liu, X., and Avital, E. One-layer particle level set method. *Computers & Fluids 170* (2018), 141 – 156.

[123] Zhao, L., Mu, K., Mao, J., Hongvan, K., and Peng, D. A three-dimensional one-layer particle level set method. *International Journal of Numerical Methods for Heat & Fluid Flow 30*, 7 (2019), 3653–3684.

[124] Zienkiewicz, O. C., Taylor, R. L., and Zhu, J. Z. *The finite element method: its basis and fundamentals*. Elsevier, 2005.