UNIVERSITY OF CALIFORNIA, MERCED

Numerical simulations of incompressible multi-phase fluid flows with environmental applications

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Applied Mathematics

by

Adam Lewis Binswanger

Committee in charge:

Shilpa Khatri, Chair

Maxime Theillard, Co-Chair

Francois Blanchette

Dustin Kleckner

2024

The dissertation of Adam Lewis Binswanger is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

(Francois Blanchette)

(Dustin Kleckner)

(Maxime Theillard, Co-Chair)

(Shilpa Khatri, Chair)

University of California, Merced

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The material in this dissertation is based upon work supported by the National Science Foundation under Grant No. DMS-1840265, the University of California, Merced through the Chancellor's Graduate Fellowship, and the National GEM Consortium through the GEM Fellowship.

I would first like to thank my advisors, Dr. Shilpa Khatri and Dr. Maxime Theillard, for all their guidance, mentorship, and constant support over the past several years. I have learned so much about how to find joy and purpose in research through working with you.

I would like to thank my other committee members, Dr. Francois Blanchette and Dr. Dustin Kleckner, for your insights and participation with my research.

I would like to thank all of my friends at UC Merced for their kindness, friendship, and support over the years.

Finally, I would like to thank my parents, Lewis and Catherine Binswanger, for all of their love and support. None of this would have been possible without you.

Adam L. Binswanger

School of Natural Sciences
5200 N Lake Rd, University of California, Merced
Merced, CA 95343, USA

720-315-2294
abinswanger@ucmerced.edu

Research Interests

Computational Science, Computational Fluid Dynamics, Numerical Analysis, High-Performance Computing, Multi-Phase Flow, Renewable Energy

Education

University of California, Merced

- Ph.D., Applied Mathematics, August 2019 – December 2024 (expected).
- Advisors: Shilpa Khatri, Maxime Theillard
- Thesis: *Numerical Simulations of Incompressible Multi-Phase Fluid Flows with Environmental Applications.*

University of Colorado Boulder

- M.S., Applied Mathematics, 2018 – 2019.
- Advisor: Mark Hoefer
- M.S. Thesis: *Nonclassical, Oblique Dispersive Shock Waves in Steady Shallow Water Flows.*
- B.S., Applied Mathematics, 2014 – 2019. (Concurrent B.S./M.S. Program)

Technical skills

C++, Python, Git Version Control, Linux, MATLAB, Object-Oriented Programming, Parallel Computing, AMReX, PeleLMeX, Mathematica, Microsoft Excel, Visual Basic for Applications

Publications

1. **Adam L. Binswanger**, Matthew Blomquist, Scott R. West, Shilpa Khatri, and Maxime Theillard. *A Stable Nodal Projection Method for Two-Phase Flows*, in preparation, (2024).
2. Matthew Blomquist, Scott R. West, **Adam L. Binswanger**, and Maxime Theillard. *Stable Nodal Projection Method on Octrees Grids*, Journal of Computational Physics, Volume 499, (2024).
3. **Adam L. Binswanger**, Mark A. Hoefer, Boaz Ilan, and Patrick Sprenger, *Whitham Modulation Theory for Generalized Whitham Equations and a General Criterion for Modulational Instability*, Studies in Applied Mathematics, Volume 147, Issue 2, (2021).

Research Experience

National Renewable Energy Laboratory

- Graduate Summer Intern **Summers 2023, 2022**
- Mentors: Marcus Day, Sreejith Nadakkal Appukuttan

Khatri Lab, University of California, Merced

- Graduate Research Assistant **August 2019 – Present**

Mathematical, Applied, and Computational Sciences (MACS) Group, University of California, Merced, Maxime Theillard

- Graduate Research Assistant **August 2019 – Present**

Dispersive Hydrodynamics Laboratory, University of Colorado Boulder, Mark Hoefer

- Graduate/Undergraduate Research Assistant **January 2017 – August 2019**

Presentations at Professional Meetings

2024

1. **Adam L. Binswanger**, Matthew Blomquist, Scott R. West, Shilpa Khatri, and Maxime Theillard. *Sharp, stable, and collocated numerical simulation of incompressible, multi-phase fluid flows.* 77th Annual Meeting of the American Physical Society Division of Fluid Dynamics, Salt Lake City, Utah, November 24 – 26, 2024.
2. **Adam L. Binswanger**, Matthew Blomquist, Scott R. West, Shilpa Khatri, and Maxime Theillard. *Sharp, stable, and collocated numerical simulation of incompressible, multi-phase fluid flows*, poster. SIAM Northern and Central California Sectional Conference (NCC24), Merced, CA, October 9 – 11, 2024.

2023

1. **Adam L. Binswanger**, Matthew Blomquist, Scott R. West, Shilpa Khatri, and Maxime Theillard. *Sharp, stable, and collocated numerical simulation of incompressible, multi-phase fluid flows.* 76th Annual Meeting of the American Physical Society Division of Fluid Dynamics, Washington, D.C., November 18 – 21, 2023.

2022

1. **Adam Binswanger** and Marc Day. *Validation of exascale combustion code for the simulation of an internal combustion engine.* Rocky Mountain Fluid Mechanics Research Symposium 2022, Boulder, CO, August 9, 2022.

2021

1. **Adam Binswanger**, De Zhen Zhou, Dustin Kleckner, Maxime Theillard, and Shilpa Khatri. *Numerical simulations of oil droplets rising in a sharply stratified fluid.* 74th Annual Meeting of the American Physical Society Division of Fluid Dynamics, Phoenix, AZ, November 21 – 23, 2021.

2020

1. **Adam Binswanger**, De Zhen Zhou, Joshua Roe, Tracy Mandel, Dustin Kleckner, Maxime Theillard, and Shilpa Khatri. *Numerical simulations of oil droplets rising in a sharply stratified fluid.* 73rd Annual Meeting of the American Physical Society Division of Fluid Dynamics, Chicago, IL, November 22 – 24, 2020.
2. **Adam Binswanger**, Mark Hoefer, Boaz Ilan, and Patrick Sprenger. *Whitham modulation theory for generalized Whitham equations and a general criterion for modulational instability.* SIAM Northern States Section Student Chapters Conference, Utah State University, Logan, Utah, October 16 – 18, 2020.

2019

1. **Adam Binswanger**, Patrick Sprenger, and Mark Hoefer. *Oblique Dispersive Shock Waves in Steady Shallow Water Flows.* IMACS International Conference on Nonlinear Evolution Equations and Wave Phenomena: Computation and Theory, University of Georgia, Athens, GA, April 17 – 19, 2019.
2. **Adam Binswanger**. *Oblique Dispersive Shock Waves in Steady Shallow Water Flows.* SIAM Front Range Applied Mathematics Student Conference 2019, Denver, CO, March 2, 2019.

2018

1. **Adam Binswanger**, Patrick Sprenger, and Mark Hoefer. *Experimental Investigation of Oblique Dispersive Shock Waves in Supercritical Shallow Water Flow.* 71st Annual Meeting of the American Physical Society Division of Fluid Dynamics, Atlanta, GA, November 18 – 20, 2018.
2. **Adam Binswanger**. *Experimental Investigation of Oblique Dispersive Shock Waves in Supercritical Shallow Water Flow.* Rocky Mountain Fluid Mechanics Research Symposium 2018, Boulder, CO, August 13 – 14, 2018.
3. **Adam Binswanger** and Maxwell Lambek. *Stationary Oblique Dispersive Shock Waves in Supercritical Shallow Water Flow.* SIAM Front Range Applied Mathematics Student Conference 2018, Denver, CO, March 3, 2018.

2017

1. **Adam Binswanger**, Maxwell Lambek, Patrick Sprenger, and Mark Hoefer. *Experimental Investigation of Dispersive Shock Waves in Shallow Water*, poster. 70th Annual Meeting of the American Physical Society Division of Fluid Dynamics, Denver, CO, November 19 – 21, 2017.

Invited Presentations

2024

1. **Adam Binswanger**. *Collocated numerical simulations of incompressible fluid flows with environmental applications.* Dispersive Hydrodynamics Laboratory Group Meeting, Boulder, CO, February 20, 2024.

Internal Presentations

2024

1. **Adam Binswanger**. *Numerical Simulations of Incompressible Multi-Phase Fluid Flows with Environmental Applications.* UC Merced Energy and the Environment Seminar, Merced, CA,

November 8, 2024.

2. **Adam Binswanger**. *Numerical Simulations of Incompressible Multi-Phase Fluid Flows with Environmental Applications.* UC Merced Scientific Computing and Data Science Seminar, Merced, CA, October 16, 2024.

3. **Adam Binswanger**. *Collocated numerical simulations of incompressible fluid flows.* UC Merced Scientific Computing and Data Science Seminar, Merced, CA, April 3, 2024.

4. **Adam Binswanger**. *Collocated numerical simulations of incompressible fluid flows with environmental applications.* UC Merced Energy and the Environment Seminar, Merced, CA, March 8, 2024.

2023

1. **Adam Binswanger**. *Stable nodal projection method on quadtree grids for incompressible, multi-phase fluid flows.* UC Merced Energy and the Environment Seminar, Merced, CA, October 6, 2023.

2. **Adam Binswanger**. *Stable nodal projection method on quadtree grids for incompressible, multi-phase fluid flows.* UC Merced Scientific Computing and Data Science Seminar, Merced, CA, September 7, 2023.

2022

1. **Adam Binswanger**. *Numerical simulations of incompressible two-phase flows.* UC Merced Energy and the Environment Seminar, Merced, CA, April 15, 2022.

2. **Adam Binswanger**. *Numerical simulations of incompressible two-phase flows.* UC Scientific Computing and Data Science Seminar, Merced, CA, March 10, 2022.

2021

1. **Adam Binswanger**. *Numerical simulations of oil droplets rising in a homogeneous fluid.* UC Merced Energy and the Environment Seminar, Merced, CA, September 17, 2021.

2. **Adam Binswanger**. *Numerical simulations of oil droplets rising in a sharply stratified fluid.* UC Merced Energy and the Environment Seminar, Merced, CA, April 9, 2021.

2020

1. **Adam Binswanger**. *Numerical simulations of oil droplets rising in a sharply stratified fluid.* UC Merced Energy and the Environment Seminar, Merced, CA, September 4, 2020.

2019

1. **Adam Binswanger**. *Non-classical, Oblique Dispersive Shock Waves in Steady Shallow Water Flows.* UC Merced Energy and the Environment Seminar, Merced CA, September 27, 2019.

Workshops and Reading Groups

2024

- San Diego Supercomputer Center HPC and Data Science Summer Institute 2024, August 5 – 9, 2024.
  - Attended session on Parallel Computing using MPI and OpenMP
  - Attended session on GPU Computing and Programming

2021

- University of California, Merced Applied Mathematics Department Deep Dive into: Deep Learning Reading Group, September 2021 – December 2021.

Fellowships

- National GEM Consortium Fellowship **2022**
- University of California, Merced Chancellor's Graduate Fellowship **2019 - 2023**

Teaching Experience

University of California, Merced

- Teaching Assistant, Math 023: Vector Calculus (Spring 2024).
- Teaching Assistant, Math 022: Calculus II for Physical Sciences and Engineering (Spring 2023, Fall 2023).
- Teaching Assistant, Math 131: Numerical Methods for Scientists and Engineers (Spring 2020, Fall 2020).
- Teaching Assistant, Math 125: MATH 125: Intermediate Differential Equations (Fall 2019).

University of Colorado Boulder

- Teaching Assistant, APPM 2360: Introduction to Differential Equations with Linear Algebra (Spring 2019).
- Teaching Assistant, APPM 2350: Calculus 3 for Engineers (Fall 2018).

Service

Yosemite Fluids Conference, 2024

- Organizing committee member **August 2023 – April 2024**

University of California, Merced Graduate Student Association

- Treasurer **August 2023 – Present**
- Applied Mathematics Delegate and Delegate Assembly Chair **August 2020 – May 2021**

University of California, Merced

- Applied Mathematics Summer Bridge 2024 Diagnostic Preliminary Exam Lead **July 2024**
- Graduate Peer Mentor in the Department of Applied Mathematics **August 2021 – May 2022**

Professional Memberships

- Society for Industrial and Applied Mathematics
- American Physical Society

xx

ABSTRACT OF THE DISSERTATION

Numerical simulations of incompressible multi-phase fluid flows with environmental applications

by Adam Lewis Binswanger
Doctor of Philosophy in Applied Mathematics

University of California Merced, 2024

Committee Chair: Shilpa Khatri

Droplets and particles are a physical feature of many atmospheric and oceanic systems. For example, the Deepwater Horizon oil spill in 2010 resulted in large plumes being trapped as they rose through stratified layers in the Gulf of Mexico. To begin understanding how and why these plumes became trapped, we produce numerical simulations of a single oil droplet rising in a stratified ambient flow and develop a force decomposition model to characterize surface forces acting on the droplet. Following this, we shift towards the development of numerical methods capable of simulating incompressible flow. We develop a novel collocated projection method for simulating incompressible multi-phase fluid flows in two and three dimensions. This method uses a modified pressure correction projection to solve the Navier-Stokes equations for the fluid flow. The fluid solver employs an adaptive mesh refinement strategy using non-graded octree/quadtree grids and a finite volume discretization for the viscosity and projection operators. The moving interface between phases is captured using a coupled level set-reference map method, which provides a sharp representation of the interface position. This method and solver are highly adaptable to multi-physics applications due to their simplified code structure and second order accuracy. We demonstrate its capabilities through a variety of density and surface tension driven multi-phase flows, including high fidelity simulations of single and multiple rising bubbles facing weak and strong surface deformations, as well as, flow of rising bubbles past solid obstructions.

Chapter 1

Introduction

Rising droplets and bubbles in stratified fluids, fluids featuring gradients in density and temperature, are a physical feature of many atmospheric and oceanic systems [42]. Such systems are vital to understanding important environmental processes. Some processes, including oil spills in the ocean [20, 35, 59, 44], fine particle pollution in the atmosphere [71], and settling marine snow [53], are important to understand in order to mitigate and prevent global environmental crises.

Oil spills in the ocean are characterized by plumes of oil rising out from a well in the deep ocean. The interaction between these oil plumes and the density variation in the ocean greatly affects the dynamics of rising. During the Deepwater Horizon Oil Spill of 2010, a significant amount of oil became trapped in intrusion layers caused by the presence of density gradients, instead of quickly rising to the surface as would be naively expected due to the oil being lighter than all the surrounding ocean water [44]. It is necessary to understand the dynamics underlying the interaction between the rising oil plume and the density stratification that led to this trapping phenomenon to understand why it occurred. This problem is an example of multi-scale dynamics that occur in environmental systems that feature interactions of droplets and particles and a density stratified ambient.

The rising plume escaping from the underwater well in an oil spill can be thought of as a collection of individual oil droplets, as well as other agents such as surfactants. Thus, both an individual droplet as well as multiple droplets' interactions with the density varying ocean should be considered. This is an example of a multi-phase fluid flow: a fluid flow in which multiple immiscible fluids interact with one another.

In order to begin studying this problem, we first consider the dynamics of a single rising oil droplet as it rises through a sharp, two-layer density stratification, in which two miscible fluids of constant density are layered on top of each other with a transition region in between, see figure 1.1. In the experiment, the droplet experienced a tremendous

slowdown and entrainment of fluid as it passed from the higher density lower layer fluid through the region of density stratification and into the lower density upper layer fluid [43]. In this dissertation, I focus on mathematically modeling and computational simulations of a single droplet with the intention to extend this work to multiple droplets in the future.

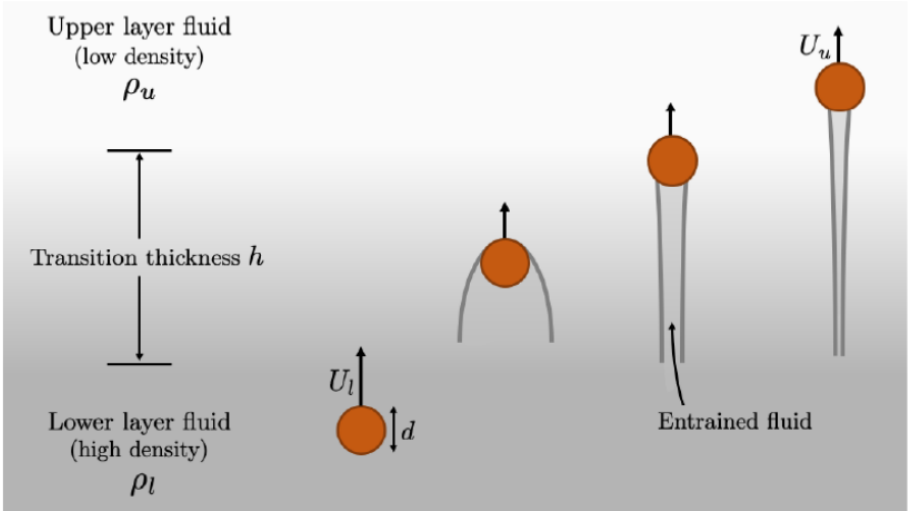

Figure 1.1: Schematic of a single droplet rising through the transition thickness h between two constant density layers, with the densities of the lower and upper layers denoted ρ_l and ρ_u, respectively ($\rho_l > \rho_u$). The entrainment of denser fluid around the droplet is shown. Figure from [43].

In environmental systems such as oil spills, experiments can be costly, difficult to perform, and limited in the quantities of interest that can be measured. In these scenarios, computational simulations are an integral tool because they allow for the flexibility to study a wide range of parameters and quantitatively measure any relevant quantity of interest that is necessary to fully understand and interpret the present dynamics. Since these problems are multi-scale and involve complex physical dynamics, the numerical methods used for these computational simulations must be accurate and efficient.

We began this investigation by modeling a single rising oil droplet as an incompressible multi-phase flow and validating this model using the numerical method for simulating incompressible two-phase flow developed in [69]. We produced simulations that share both qualitative and quantitative agreement with data from the experiment conducted in [43]. These simulations are featured in chapter 2. However, this numerical method was limited in its ability to accurately resolve the forces present on the oil droplet when moving through a stratified ambient, necessitating the use of a different numerical

method with a higher order of accuracy. We developed a novel numerical method for simulating incompressible multi-phase fluid flow that we believe is capable of accurately simulating rising oil droplets in density stratified flows, and that is highly adaptable to studying other similar multi-physics applications. The details of this numerical method and its validation are described in chapters 5 and 6. This method is an extension of a similar method for simulating single phase fluid flow, which is described in chapter 4. We aim to do a full numerical investigation of oil droplets in the future using the new numerical method.

1.1 Model equations for rising oil droplets in density stratified flows

We consider a domain Ω consisting of two fluid phases: Ω^+ the droplet phase and Ω^- the fluid phase. The two phases are separated by an interface Γ, upon which there is a curvature κ and interfacial tension γ, see figure 1.2. The fluid velocity and pressure are

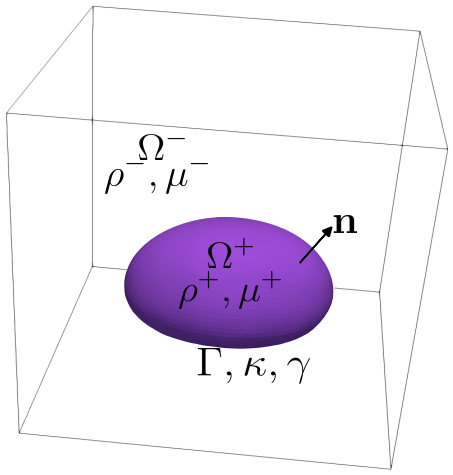

Figure 1.2: Problem schematic where the densities in each phase are denoted by $\rho^{\pm}$ and the viscosities in each phase are denoted by $\mu^{\pm}$. The interface Γ has a curvature κ and interfacial tension γ.

modeled by the incompressible Navier-Stokes equations with the Boussinesq approximation, given by

$$\rho_0 \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \boldsymbol{\Delta} \mathbf{u} + \rho \mathbf{g} \qquad \forall \mathbf{x} \in \Omega \setminus \Gamma, \tag{1.1}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \forall \mathbf{x} \in \Omega \setminus \Gamma, \tag{1.2}$$

where $\mathbf{u}(\mathbf{x}, t)$ is the fluid velocity, $p(\mathbf{x}, t)$ is the pressure, $\mathbf{g}$ is the acceleration due to gravity, $\rho(\mathbf{x}, t)$ is the density, and μ is the viscosity. The density ρ_0 is a constant value in each phase, and is defined as the density of the droplet in Ω^+ and an average of the density in Ω^-. Equations (1.1) and (1.2) represent conservation of momentum and mass, respectively. Due to the presence of the interface Γ, the model for fluid velocity and pressure has the following interface jump conditions

$$[\![\mathbf{u}]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{1.3}$$

$$[\![\sigma \cdot \mathbf{n} - p\mathbf{n}]\!] = \gamma\kappa\mathbf{n} \qquad \forall \mathbf{x} \in \Gamma, \tag{1.4}$$

where $[\![\chi]\!] = \chi^+ - \chi^-$ is the jump of the quantity χ across the interface Γ, γ is the coefficient of interfacial tension, κ is the curvature of the interface Γ, $\mathbf{n}$ is the outward facing normal vector, and $\sigma = \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)$ is the viscous stress tensor. These jump conditions represent the continuity of velocity across the interface and the jump in stress due to interfacial tension, respectively.

Additionally, because the density in Ω^- features a density gradient, we model the variable density in Ω^- with the following advection-diffusion equation

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla\rho = D_\rho \boldsymbol{\Delta}\rho \qquad \forall \mathbf{x} \in \Omega^-, \tag{1.5}$$

where D_ρ is the diffusion coefficient. This diffusion coefficient is chosen in practice to be on the order of $D_\rho \approx 10^{-5}$ cm^2/s, as this is the diffusion coefficient of salt in water. In practice, this is smaller than the numerical diffusion of the method used to solve Eq. (1.5), but this is not an issue because we expect the effects of advection to primarily drive the evolution of the density field in the ambient fluid over time.

Equations (1.1)-(1.5) are nonlinear vector equations that involve solving for multiple, coupled variables, and feature jump conditions connecting the two fluid phases. This makes solving this set of equations numerically challenging. Projection methods [22] are a widely used class of numerical methods that numerically solve the Navier-Stokes equations (1.1)-(1.5) in a splitting approach in which the momentum equation Eq. (1.1) is solved first and then corrected to account for the divergence free condition Eq. (1.2). The numerical method [69] that we began this investigation with is a modified pressure correction projection method, which handles the jump conditions Eqs. (1.3) and (1.4) through an iterative approach. Similarly, the novel numerical method we developed during this work, presented in chapter 5, is also a type of modified pressure correction projection method.

Given that multi-phase fluid flows model systems with complex dynamics that occur across a range of relevant scales, adaptive mesh refinement (AMR) must be employed as

part of the numerical method to ensure computational efficiency. The method that we developed uses adaptive octree/quadtree grids [55] in 3D and 2D, respectively, as the AMR strategy. With projection methods specifically, one primary concern that must be addressed is the storage location of all computational variables on the computational grid. Staggered grids, such as the widely used Marker and Cell (MAC) grid [30], which stagger the storage locations of the velocity components and pressure, are often preferred because they can be readily shown to be stable, even on adaptive grids, see for example [69]. The method we developed instead collocates all computational variables at the nodes of the computational grid. In doing so, we derive a nodal projection operator that is used to project the velocity field found from solving the momentum equation Eq. (1.1) into the divergence free space. This choice allows us to obtain a higher order of accuracy and is shown to be numerically stable under a range of boundary conditions, though the projection into the space of divergence free velocity fields is only approximate.

This dissertation is organized as follows: In chapter 2, we present the comparison between simulations of rising oil droplets in a sharply density stratified flow produced using the numerical method from [69] and the experimental data of [43]. In chapter 3, we introduce a novel model for the decomposition of surface forces acting on the rising oil droplet and present preliminary results of this model using the method of [69]. In chapter 4 we given an overview of the nodal projection operator that is the foundation of our numerical method for simulating incompressible two-phase flow and summarize the numerical method for incompressible single phase flow that our two-phase flow method extends. Chapter 5 is a thorough overview of all details of the numerical method for simulating incompressible two-phase flow, and chapter 6 details the verification and validation of this method.

Chapter 2

Rising oil droplets in stratification

In this work, I developed a framework to computationally study a single rising oil droplet as it rises through a sharp, two-layer density stratified fluid, as described in figure 1.1. This framework involves computing the velocity and pressure of the oil and ambient fluid, by numerically solving Eqs. (1.1)-(1.5), as the droplet travels through a region of density stratified fluid. Then we use the velocity and pressure to compute several interfacial forces that dictate the effect of stratification on the droplet's rising behavior.

This problem was studied experimentally at UC Merced in [43], where time scales of the entrainment process were quantified for a range of experimental parameters, such as Reynolds numbers ranging from 5.4 - 1060. While these experiments were able to quantify aspects of the process of entrainment and slowdown as the droplet passes through the transition region, quantities such as forces and energies were not capable of being measured. To further the investigation of this experiment, I use numerical simulations to capture the experiment in a framework where all numerical quantities are available and capable of being used to solve for relevant forces and energies. The primary goal of my investigation has been to use simulations to measure and understand the forces that play the largest roles in driving droplet entrainment and slowdown.

The subject of the rising and settling of particles, drops, and bubbles in homogeneous density and density stratified fluids has been an area of active experimental and computational research. The incorporation of density stratification in particular has led to numerous studies into how to quantify the effects of stratification on rising and settling by means of decomposing the effects of stratification into multiple, distinct forces. Some experimental investigations include [32], where settling and rising spheres were studied in density homogeneous flows and the vortex shedding and wake structure were investigated in a range of moderate Reynolds number regimes and [74, 6], where the terminal drop rise velocity of several types of droplets including toluene and n-butanol in water were

studied. These studies helped establish the general velocity profiles of rising and settling particles and drops present in regimes in the absence of any density stratification.

In addition, there have been experimental studies that focused on the increased complexity that density stratification adds to the rising and settling dynamics of solid particles. In [61], the gravitational settling of solid particles in a two layer density stratification demonstrated that particles experience an additional drag force and deceleration due to stratification. Similary, [73] also found that settling particles passing through a two layer stratification region reach a velocity minimum as the particle exits from the density transition region. This phenomenon was also observed in [1], in which settling spheres were studied in a strongly stratified two layer regime, where the stratification forces were also observed to have caused a reversal in the sphere's direction of motion. Finally, settling spheres in low Reynolds number, or Stokes flow, regimes with both two layer and linear stratification were studied in [77, 19, 45], quantifying the entrainment in this additionally physically relevant regime. Finally, [43] studied rising oil droplets in a sharp two layer density stratification. They quantified two timescales describing the droplet's delay, as well as estimations on stratification forces contributing to the delay. It is this experiment that this work aims to further through numerical simulations.

There similarly exists a comparable body of numerical studies of rising and settling particles, spheres, and drops in both homogeneous and density stratified fluids. In [5], the wake dynamics of very light rising spheres under buoyancy in a density homogeneous flow were studied. Both [6, 11] featured numerical investigations into the rising of n-butanol droplets in density homogeneous water, which agreed with corresponding experimental investigations. For drops in density stratified fluids, [7] observed that the drag coefficient of settling spherical drops was enhanced in linearly stratified flows under a range of Froude numbers. This study also considered sharp two-layer stratified fluids, and saw a similar drag enhancement. The study [58], also in a linearly stratified flow, observed an increase in drag in a low Reynolds number regime due to stratification and inertial forces. Finally, [13] considered simulations of oil droplets in two layer stratification, and found that Marangoni forces, forces arising due to surface tension gradients, also play a significant role in stratification induced drag enhancement.

In this chapter, I give an overview of the means by which the model equations are numerically solved and show validation of the simulations with available experimental data. In chapter 3 of this dissertation, interfacial forces are identified and solved for using a decomposition of the pressure field corresponding to distinct physical effects such as viscosity and buoyancy. While this decomposition proved capable of solving for distinct interfacial forces over time, a preliminary numerical investigation in both a homogeneous and stratified ambient yielded noisy and inaccurate results that were not sufficient to

perform a full investigation on the role of each force in driving droplet entrainment. This limitation in accuracy inspired the development of a novel numerical method to solve Eqs. (1.1)-(1.5) that has a higher order of accuracy, the subject of chapters 4 - 6 of this dissertation.

2.1 Model equations

As previously stated, the velocity, pressure, and density field of a single rising oil droplet in stratification are modeled using Eqs. (1.1)-(1.5). The momentum equation (1.1) has been simplified using the Boussinesq approximation to have any variable density effects be considered only in the external body forces. This approximation is valid for density variations in the ambient fluid that are near the range of $5 - 10\%$ [60], which is true in the experiments these simulations are modeling.

To focus more closely on the interfacial effects that are induced by density stratification, we rewrite these equations by rescaling the pressure to remove the lower layer hydrostatic pressure, $p_{\text{hydrostatic}} = \rho_l g_y y$, where g_y is the non-zero component of the acceleration due to gravity, y is the coordinate in the rising direction, and ρ_l is the lower-layer density. We define the rescaled pressure $\widetilde{p}$ as

$$\widetilde{p} = p - \rho_l g_y y. \tag{2.1}$$

As such, Eq. (1.1) can be written in terms of $\widetilde{p}$ as

$$\frac{1}{\rho_0}\nabla\widetilde{p} = -\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) + \frac{\mu}{\rho_0}\boldsymbol{\Delta}\mathbf{u} + \frac{(\rho-\rho_l)}{\rho_0}\mathbf{g}, \tag{2.2}$$

with corresponding jump conditions at the interface

$$[\![\mathbf{u}]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{2.3}$$

$$[\![\sigma\cdot\mathbf{n} - \widetilde{p}\mathbf{n}]\!] = \gamma\kappa\mathbf{n} \qquad \forall \mathbf{x} \in \Gamma, \tag{2.4}$$

We also define a rescaled density profile, $\widetilde{\rho}$,

$$\widetilde{\rho} = \rho - \rho_l. \tag{2.5}$$

As such, we can write Eq. (2.2) as

$$\frac{1}{\rho_0}\nabla\widetilde{p} = -\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) + \frac{\mu}{\rho_0}\boldsymbol{\Delta}\mathbf{u} + \frac{\widetilde{\rho}}{\rho_0}\mathbf{g}. \tag{2.6}$$

Finally, we note the wall boundary conditions for $\mathbf{u}$ and $\widetilde{p}$,

$$\frac{\partial \mathbf{u}}{\partial n} = 0 \qquad y = Y_{max}, \tag{2.7a}$$

$$\mathbf{u} = \mathbf{0} \qquad \forall \mathbf{x} \in \partial\Omega : y \neq Y_{max}, \tag{2.7b}$$

$$\widetilde{p} = 0 \qquad y = Y_{max}, \tag{2.8a}$$

$$\frac{\partial \widetilde{p}}{\partial n} = 0 \qquad \forall \mathbf{x} \in \partial\Omega : y \neq Y_{max}, \tag{2.8b}$$

where we consider the boundary of the domain to be $\partial\Omega = [X_{min}, X_{max}] \times [Y_{min}, Y_{max}] \times [Z_{min}, Z_{max}]$.

2.2 Numerical method

The model equations, Eqs. (2.3)-(2.8), are solved using the numerical method presented in [69]. This numerical method is a pressure correction projection method on adaptive octree grids, which uses a staggered Marker and Cell (MAC) grid for storing the computational variables. The details of this numerical method will be elucidated on further in chapter 5, as the new numerical method developed in this dissertation is based on this method. The primary difference between these methods is the grid on which computational variables are stored and the number of interpolations required due to the new grid layout. In short, these numerical methods solve the incompressible two-phase Navier-Stokes equations Eqs. (1.1)-(1.4) through a splitting approach where the momentum equation Eq. (1.1) is solved first and then corrected to account for the divergence free condition Eq. (1.2). The advection is treated explicitly and diffusion is treated implicitly. The adaptive octree grid is refined in areas near the interface and where there are high variations in density, velocity, and vorticity.

Additionally, the interface between oil and water, Γ, is represented using the level-set method [48] as the zero-contour of an auxiliary function ϕ. The details of how this level set function is advanced in time are also explained further in chapter 5, although this method does not use the volume preserving projection to the reference map that is incorporated in the new numerical method.

2.3 Comparison with experiment

Using the numerical method of [69], several comparisons between simulations and experiments were performed. Shown in figure 2.1 is an image from a 3D simulation of a

2D slice of the density field in the ambient fluid surrounding a droplet that is passing through the transition region. This figure demonstrates that the simulation captures the entrainment of higher density fluid as the droplet passes from the lower layer to the upper layer. Shown in figure 2.2 is a comparison between the shadowgraph of a simulation and

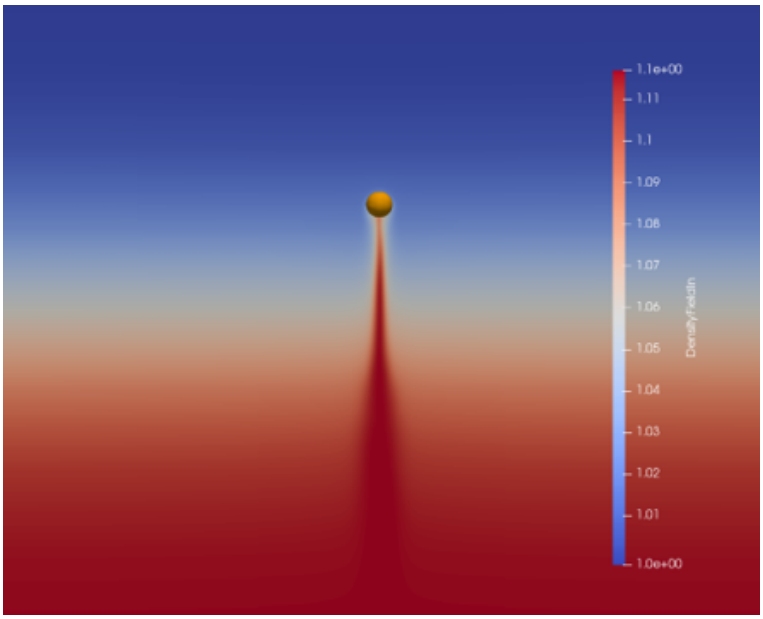

Figure 2.1: Simulation of rising oil droplet in sharp two layer stratification. A 2D slice of the 3D simulation is shown, with the color representing the density in the ambient fluid. The lower-layer density depicted in red is $\rho_l = 1.1\ \mathrm{g/cm^3}$ and the upper-layer density depicted in blue is $\rho_u = 1\ \mathrm{g/cm^3}$. Here, the density and viscosity of the droplet are $\rho_d = 0.9972\ \mathrm{g/cm^3}$ and $\mu_d = 0.098\ \mathrm{g/(cm \cdot s)}$, respectively. Additionally, the Reynolds number in the lower layer is $Re = 370$. The entrainment of density from the lower layer is depicted.

experiment. The shadowgraph is a quantity computed as $\nabla^2\widetilde{\rho}$, and is used to visualize the drop's wake structure. Though the simulation does not fully capture the unsteady nature of the wake structure far from the drop, it does capture the near drop wake fairly close to experiment. Additionally, we note that this unsteadiness could be better captured with the addition of some sort of destabilization mechanism or symmetry breaking in the model.

Shown in figures 2.3 and 2.4 are a comparison between the droplet's vertical position and velocity over time. We observe some agreement with the experiment, in particular with capturing the trends in the rising velocity. In both the experiment and simulation, we observe that the droplet rises with an asymptotic rising velocity in the lower layer,

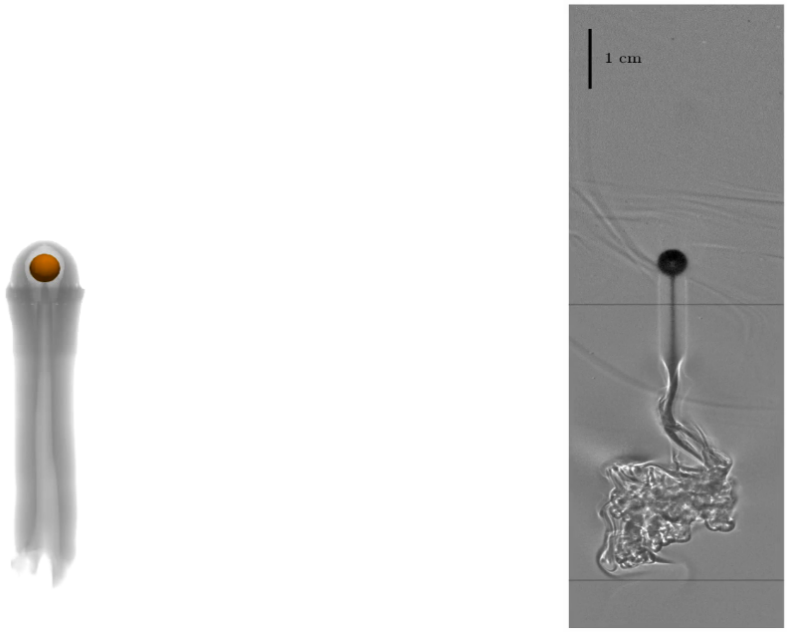

Figure 2.2: Comparison between the shadowgraph of a simulation (left) and the corresponding experiment (right) as the droplet travels through the transition region. The parameters in the simulation were set to be equal to those in the experiment, where the lower-layer density is $\rho_l = 1.1\ \mathrm{g/cm^3}$ and the upper-layer density is $\rho_u = 1\ \mathrm{g/cm^3}$. Here, the density and viscosity of the droplet are $\rho_d = 0.9972\ \mathrm{g/cm^3}$ and $\mu_d = 0.098\ \mathrm{g/(cm \cdot s)}$, respectively, and the Reynolds number in the lower layer is $Re = 370$.

experiences a tremendous drop in velocity as it travels through the transition region, reaches a velocity minimum shortly after entering the upper layer, and finally accelerates until achieving an asymptotic rising velocity in the upper layer. While our simulations produced similar orders of magnitude in position and velocity compared with the experiment, there is a noticeable discrepancy between the two results. We believe that this is due to limitations in the model itself, as, for example, we do not model the asymmetry in the wake that was observed in the experiment shown in figure 2.2. Nevertheless, these comparisons demonstrate validation of the mathematical model and of the numerical method itself in being able to reproduce the velocity trends observed and measured in the experiments.

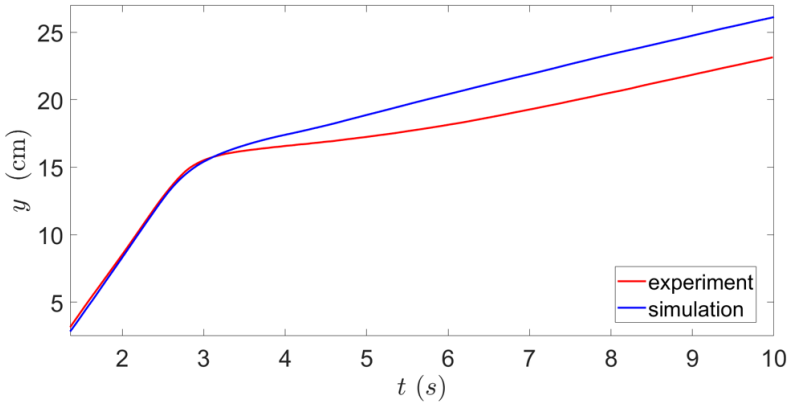

Figure 2.3: Droplet rising vertical position vs time, comparison between experiment (red) and simulation (blue).

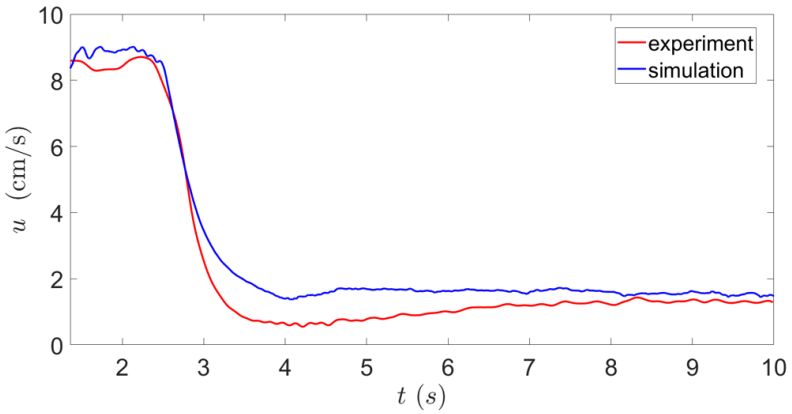

Figure 2.4: Droplet rising vertical velocity vs time, comparison between experiment (red) and simulation (blue).

Chapter 3

Investigation of interfacial forces

In this chapter, we present a model for the interfacial forces impacting a rising oil droplet as it rises through a region of density stratified ambient fluid. Inspired by the work of [78], this model is a force decomposition model, obtained from decomposing the fluid pressure into distinct components that each derive from a physical effect such as the gravitational force and interfacial tension and that are all impacted by the presence of density stratification. We present a preliminary numerical investigation of this model using the model and numerical method [69] presented in chapter 2, in both a homogeneous and stratified ambient.

There have been several studies focused on quantifying the multiple forces that arise in stratified flows in order to understand the role each plays on rising and settling particle and droplet drag enhancement. In [23], a distinct stratification force on settling spheres was identified in a linearly stratified flow due to vorticity resulting from baroclinic torque, present due to a misalignment of density and pressure gradients. This work was expanded on in [78], in which a force decomposition was proposed including an additional buoyant entrainment force and a force due to baroclinic torque. This force decomposition approach serves as the inspiration for the force decomposition model presented in this chapter.

3.1 Net force acting on droplet

The net force acting on the droplet is the sum of body forces plus the drag force,

$$\mathbf{F}_{\text{net}} = \rho_d \mathbf{g} V + \mathbf{F}_{\text{drag}}, \tag{3.1}$$

where ρ_d and V are the density and volume of the droplet, respectively, and

$$\mathbf{F}_{\text{drag}} = \int_\Gamma \mathbf{n} \cdot (-(\widetilde{p} + \rho_l g_y y))\mathbb{I} + \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}\right))\mathrm{d}S, \tag{3.2}$$

where $\mathbb{I}$ is the identity matrix. Then as such, the net force is given as

$$\mathbf{F}_{\text{net}} = \rho_d \mathbf{g} V + \int_\Gamma \mathbf{n} \cdot (-(\widetilde{p} + \rho_l g y)\mathbb{I} + \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}\right))\mathrm{d}S. \tag{3.3}$$

And we can simplify this by integrating the hydrostatic pressure term and obtain

$$\mathbf{F}_{\text{net}} = (\rho_d - \rho_l)\mathbf{g} V + \int_\Gamma \mathbf{n} \cdot (-\widetilde{p}\,\mathbb{I} + \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}\right))\mathrm{d}S. \tag{3.4}$$

Then Eq. (3.4) can be used to compute the net force on the droplet. We wish to split up the integral term in Eq. (3.4) by decomposing $\widetilde{p}$ into multiple components that represent different physical effects.

3.2 Identification of interfacial forces

Considering Eqs. (2.6), (2.3), and (2.4), we note that there are four distinct effects comprising the pressure: buoyancy effects due to terms containing $\widetilde{\rho}\mathbf{g}$, viscous effects due to terms containing μ, interfacial tension effects due to terms containing γ, and remaining inertial effects due to the remaining terms. This approach is similar to [78] but now in the context of drops and interfacial tension forces instead of solid spheres. This observation allows us to decompose the rescaled pressure as

$$\widetilde{p} = p_\mu + p_\rho + p_u + p_\gamma, \tag{3.5}$$

where p_μ is the pressure due to viscous effects, p_ρ is the pressure due to buoyant effects, p_γ is the pressure due to interfacial tension effects, and p_u is the pressure due to remaining inertial effects.

3.3 Derivation of Poisson jump problems

Given the pressure decomposition Eq. (3.5), we now wish to derive a set of equations to solve for each component, allowing us to isolate each and study their relative effects on droplet entrainment over time. Poisson problems can be derived for each pressure term by

first taking the divergence of Eq. (2.2) and applying the divergence free condition $\nabla \cdot \mathbf{u} = 0$, resulting in

$$\frac{1}{\rho_0}\Delta\widetilde{p} = -\left(\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u})\right) + \frac{1}{\rho_0}\left(\frac{\partial}{\partial y}\widetilde{\rho}\right) g_y. \tag{3.6}$$

We can also take the dot product of Eq. (2.6) with the normal vector $\mathbf{n}$ and obtain

$$\frac{1}{\rho_0}\nabla\widetilde{p} \cdot \mathbf{n} = -\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) \cdot \mathbf{n} + \frac{\mu}{\rho_0}\Delta\mathbf{u} \cdot \mathbf{n} + \frac{\widetilde{\rho}}{\rho_0}\mathbf{g} \cdot \mathbf{n}. \tag{3.7}$$

We will use Eq. (3.7) to determine the jump conditions for the pressure when $\mathbf{x} \in \Gamma$. In addition, we note that the jump condition Eq. (2.4) can be re-written as

$$[\![\widetilde{p}]\!] = [\![\mathbf{n} \cdot \sigma \cdot \mathbf{n}]\!] - \gamma\kappa. \tag{3.8}$$

Then using Eqs. (3.6), (3.7), and (2.8) we define the Poisson problems as

$$\frac{1}{\rho_0}\Delta p_\mu = 0 \tag{3.9a}$$

$$[\![p_\mu]\!] = [\![\mathbf{n} \cdot \sigma \cdot \mathbf{n}]\!] \qquad \forall \mathbf{x} \in \Gamma, \tag{3.9b}$$

$$\left[\!\left[\frac{1}{\rho_0}\mathbf{n} \cdot \nabla p_\mu\right]\!\right] = \left[\!\left[\frac{\mu}{\rho_0}\Delta\mathbf{u} \cdot \mathbf{n}\right]\!\right] \qquad \forall \mathbf{x} \in \Gamma, \tag{3.9c}$$

$$p_\mu = 0 \qquad y = Y_{max}, \tag{3.9d}$$

$$\frac{\partial p_\mu}{\partial n} = 0 \qquad \forall \mathbf{x} \in \partial\Omega : y \neq Y_{max}, \tag{3.9e}$$

$$\frac{1}{\rho_0}\Delta p_\rho = \frac{1}{\rho_0}\left(\frac{\partial}{\partial y}\widetilde{\rho}\right) g_y \tag{3.10a}$$

$$[\![p_\rho]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{3.10b}$$

$$\left[\!\left[\frac{1}{\rho_0}\mathbf{n} \cdot \nabla p_\rho\right]\!\right] = \left[\!\left[\frac{\rho}{\rho_0}\mathbf{n} \cdot \mathbf{g}\right]\!\right] \qquad \forall \mathbf{x} \in \Gamma, \tag{3.10c}$$

$$p_\rho = 0 \qquad y = Y_{max}, \tag{3.10d}$$

$$\frac{\partial p_\rho}{\partial n} = 0 \qquad \forall \mathbf{x} \in \partial\Omega : y \neq Y_{max}, \tag{3.10e}$$

$$\frac{1}{\rho_0}\Delta p_u = -\left(\nabla\cdot(\mathbf{u}\cdot\nabla\mathbf{u})\right) \tag{3.11a}$$

$$[\![p_u]\!] = 0 \qquad \forall \mathbf{x}\in\Gamma, \tag{3.11b}$$

$$\left[\!\left[\frac{1}{\rho_0}\mathbf{n}\cdot\nabla p_u\right]\!\right] = 0 \qquad \forall \mathbf{x}\in\Gamma, \tag{3.11c}$$

$$p_u = 0 \qquad y = Y_{max}, \tag{3.11d}$$

$$\frac{\partial p_u}{\partial n} = 0 \qquad \forall \mathbf{x}\in\partial\Omega : y \neq Y_{max}, \tag{3.11e}$$

$$\frac{1}{\rho_0}\Delta p_\gamma = 0 \tag{3.12a}$$

$$[\![p_\gamma]\!] = -\gamma\kappa \qquad \forall \mathbf{x}\in\Gamma, \tag{3.12b}$$

$$\left[\!\left[\frac{1}{\rho_0}\mathbf{n}\cdot\nabla p_\gamma\right]\!\right] = 0 \qquad \forall \mathbf{x}\in\Gamma, \tag{3.12c}$$

$$p_\gamma = 0 \qquad y = Y_{max}, \tag{3.12d}$$

$$\frac{\partial p_\gamma}{\partial n} = 0 \qquad \forall \mathbf{x}\in\partial\Omega : y \neq Y_{max}. \tag{3.12e}$$

Eq. (3.9) is for the viscous pressure, Eq. (3.10) is for the buoyancy pressure, Eq. (3.11) is for the remaining inertial pressure, and Eq. (3.12) is for the interfacial tension pressure. We note that in Eq. (3.10c), the flux jump is in ρ and not in $\widetilde{\rho}$ due to the fact that the jump in ρ_l is zero. We also note in Eq. (3.11c) the flux jump is zero due to continuity of $\mathbf{u}$ across Γ.

Additionally, we have another viscous force,

$$\mathbf{F}_{\text{visc,2}} = \int_\Gamma \mathbf{n}\cdot(\mu\left(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}}\right))\mathrm{d}S, \tag{3.13}$$

though this force is not due to the pressure, but rather the velocity field alone. The individual forces are computed by integrating the pressure components over the interface, as

$$\mathbf{F}_\mu = \int_\Gamma \mathbf{n}\cdot(-p_\mu\mathbb{I})\mathrm{d}S, \tag{3.14}$$

$$\mathbf{F}_\rho = \int_\Gamma \mathbf{n}\cdot(-p_\rho\mathbb{I})\mathrm{d}S, \tag{3.15}$$

$$\mathbf{F}_u = \int_\Gamma \mathbf{n}\cdot(-p_u\mathbb{I})\mathrm{d}S, \tag{3.16}$$

$$\mathbf{F}_\gamma = \int_\Gamma \mathbf{n} \cdot (-p_\gamma \mathbb{I}) \mathrm{d}S. \tag{3.17}$$

As such, the net interfacial force (the net force from Eq. (3.4) minus the first term) can be computed as

$$\mathbf{F}_{\text{I,net}} = \int_\Gamma \mathbf{n} \cdot (-(p_\mu + p_\rho + p_u + p_\gamma)\mathbb{I}) \mathrm{d}S + \int_\Gamma \mathbf{n} \cdot (\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}} \right)) \mathrm{d}S. \tag{3.18}$$

3.4 Preliminary results of droplet in homogeneous and stratified fluids

Using the Poisson problems Eqs. (3.9)-(3.12), interfacial forces were computed using pressure components for a droplet in a homogeneous ambient and a droplet in a sharp two layer density stratified ambient. The numerical method used to solve these Poisson problems is the Poisson solver described in chapter 5.9.1.

3.4.1 Homogeneous ambient

To verify the accuracy of the Poisson jump problems Eqs. (3.9)-(3.12) as a decomposition of the full pressure, first a comparison between the sum of the pressure components and the full pressure was performed in a homogeneous ambient fluid. This comparison is shown in figure 3.1, and aside from deviations at early times at the start of the simulation, the pressure decomposition matches fairly well with the full pressure, demonstrating that the Poisson jump problems were formulated correctly in the constant density ambient fluid configuration. The discrepancy between the full pressure and the sum of the pressure components is explained by limitations in accurately computing terms in Eqs. (3.9)-(3.12) that involve multiple derivatives of $\mathbf{u}$, as $\mathbf{u}$ is obtained from a first order accurate numerical method.

A full force investigation was additionally performed on a drop rising in a homogeneous flow, shown in figure 3.2. In this figure, interfacial forces from the pressure decomposition are shown with solid lines, and the remaining forces and net force are shown with dashed lines. In this scenario, the droplet rises until it reaches a terminal velocity, and thus rises until it has zero net force. This is correctly demonstrated in the simulation (see dashed red line). This also demonstrates the capability of the individual forces to be studied over the course of a simulation.

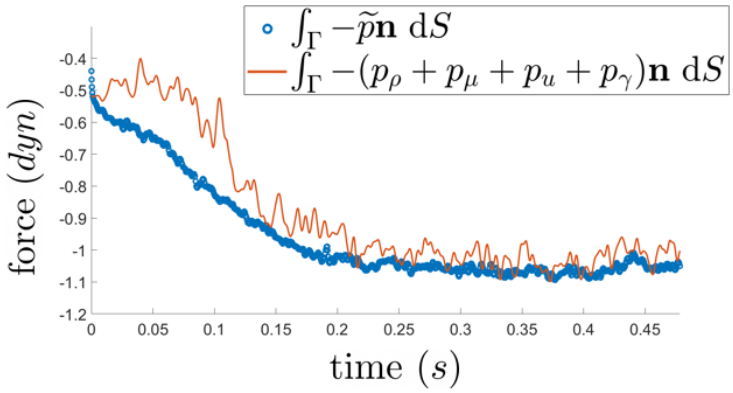

Figure 3.1: Pressure decomposition (orange) compared to the full pressure (blue) for a droplet rising in a homogeneous flow. In this case, the interfacial forces due to pressure were computed for the pressure decomposition and for the full pressure for the comparison.

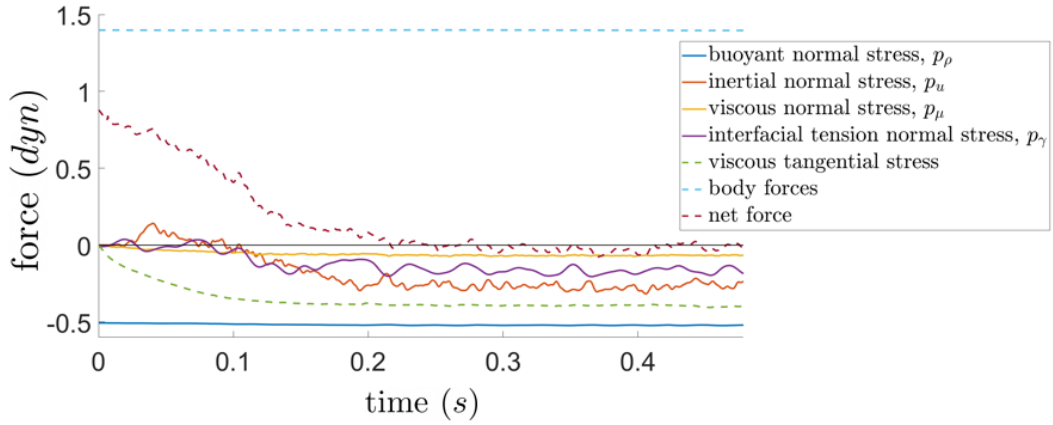

Figure 3.2: Forces acting on a rising droplet in homogeneous flow vs time. The interfacial forces due to the pressure decomposition are shown as solid lines, and the remaining forces and net force are shown as dashed lines.

3.4.2 Sharp two layer stratified ambient

Finally, the force decomposition is applied to a simulation of a drop rising in a sharp two layer density stratification, with the same parameters as the case shown in figures 2.1 and 2.2. Equations (3.9)-(3.12) are solved and the resulting forces per volume are shown in figures (3.3) and (3.4). Note that the buoyant force is shown with a full, unscaled pressure with the lower layer hydrostatic pressure still present. While the resulting buoyant force is reasonable and matches the expected behavior while the droplet is in the lower layer, the remaining forces are extremely noisy and likely quite inaccurate, even in the lower layer. We believe again that this noise and inaccuracy is due to the velocity field not being obtained in manner that allows for accurate calculations of its second derivatives, given that the velocity field is computed using the first order accurate method [69]. While these inaccuracies were more manageable in the constant density ambient fluid case shown in section 3.4.1, the droplet experiences far more dramatic changes in its velocity when in the presence of density stratification. As a result, there is a much greater need in the fully density stratified case to accurately compute the inertial and viscous effects modeled through two derivatives of the velocity field. Attempts to remedy this noisy and inaccurate behavior proved to be insufficient, leading to the realization that we should use a different numerical method with higher accuracy in the velocity field to solve the model equations Eqs. (1.1)-(1.2). The velocity field and its derivatives are the primary tool needed to compute these remaining forces, as seen in Eqs. (3.9) and (3.11), and thus need to be computed with the highest accuracy possible. A novel numerical method that fits this criteria is the focus of the rest of this dissertation.

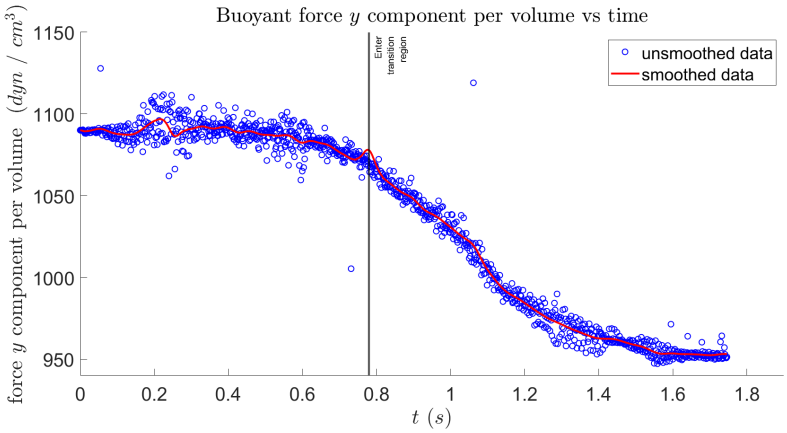

Figure 3.3: Buoyancy force per volume vs time (blue) with a smoothed curve (red). This additionally accounts for the lower layer hydrostatic pressure. The vertical black line signifies the time at which the droplet enters the transition region from the lower layer.

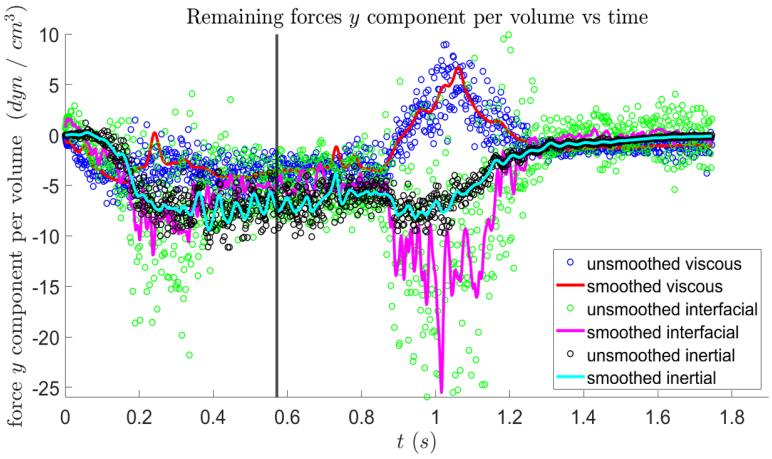

Figure 3.4: Viscous, remaining inertial, and interfacial tension forces per volume vs time (circles) along with smoothed curves (solid lines). The vertical black line signifies the time at which the droplet enters the transition region from the lower layer.

3.5 Conclusions

In this project, we successfully modeled and numerically simulated an oil droplet rising in a sharp two layer stratification. These simulations were validated against available experimental data, and show both qualitative and some quantitative agreement. From this starting point, we developed a model for forces acting on the surface of the droplet as it rises through the region of density stratification. We validated this model using a rising oil droplet in a homogeneous ambient layer without the effects of density stratification. However, when applied to a case with density stratification, limitations in the accuracy of the numerical method used led to an inability to perform a full quantification of the relative effects of each interfacial force on droplet slowdown and entrainment.

To address these limitations, we develop a new numerical method for simulating incompressible two-phase flow that shares similarities with the method [69] but is an order of accuracy higher in the velocity. With this method implemented and validated for examples featuring a constant density flow, the future direction for this work first begins with an incorporation of density stratification to this method. Following this, the validation tests shown in chapter 2 will be performed to ensure that these simulations still match the behavior of the experiment. After, the same analysis will be performed on interfacial forces as outlined in this chapter, with efforts made to further tune the

numerical method to produce the most accurate interfacial forces possible. Past studies [7, 58, 13] show that stratification induces additional interfacial forces that enhance the drag on rising and settling particles, and this future direction will allow us to quantify these interfacial forces on the rising drop.

Another direction of future work is the incorporation of additional forces that were not included in the model Eqs. (1.1)-(1.5). For example, the study [13] found that the Marangoni effect plays a significant role in drag enhancement on rising and settling particles. Incorporating the Marangoni effect into the model involves an additional term in the jump condition Eq. (1.4), which will need to be addressed with additions to the numerical method.

Chapter 4

Stable nodal projection methods

The primary challenge with numerically solving the incompressible Navier-Stokes equations, Eqs. (1.1) and (1.2) is the coupling between the conservation of mass and momentum, given by the incompressibility constraint on the velocity field, Eq. (1.2), that must be satisfied at all times. Multi-phase fluid flow additionally provides the challenge of mathematically representing the interface, tracking its evolution, and in capturing the discontinuity in quantities between the distinct fluid phases across the interface, see figure 1.2.

Given that multi-phase fluid flows are the focal point of numerous applications in environmental, biological, and industrial applications, the development of numerical methods to simulate the multi-phase incompressible Navier-Stokes equations has been an area of tremendous research. Peskin's immersed boundary method [49], originally developed for simulating blood flow in the heart, was first used to simulate incompressible multiphase fluid flow with a front tracking method in [70, 72] and with the level-set method in [65, 64]. The level-set approach was also coupled with the volume of fluid method in [63, 62] to improve mass conservation. These methods handle the discontinuity in quantities across the interface through the use of numerical δ-functions, which leads to an artificial smearing of quantities across the interface due to numerical diffusion.

Sharp capturing methods, methods that treat the interface as having zero thickness, have been developed to improve the capturing of the discontinuity in quantities across the interface. The immersed interface method was developed by [39] to improve on the immersed boundary method as a sharp interface capturing method. This method was originally applied to solving the incompressible Stokes equations [38] and the Navier Stokes equations [37, 40] with fluids that have identical densities and viscosities. Later, this method was generalized for the Stokes equations to account for discontinuous viscosities with an augmented variable approach. This approach has also shown success in

handling non-smooth velocities and pressures for the Navier-Stokes equations [66, 67], but has had less work developed regarding the handling of discontinuous densities.

Another example of a sharp capturing method is the ghost fluid method, originally developed in [24]. This method preserves the sharpness of quantities across the interface in the normal direction, but still leads to the smearing of quantities in the tangential direction. This method was used to simulate incompressible multi-phase flow in [34, 62] using the projection method of [22] to solve for the fluid velocity.

There has been a variety of approaches developed focusing on ways to correctly enforce the jump conditions at the interface. One example is the virtual node method, originally developed in [8, 31] and used for simulating incompressible flow in [4, 57], which takes on a monolithic approach and discretizes the mass and momentum equations together to solve for velocity and pressure and uses an embedded boundary approach to represent the interface. Another example is the gauge method, used in methods such as [56, 18], where the jump conditions are reformulated as auxiliary and gauge variables.

Level-set based methods have also been developed using a variety of strategies to capture the sharpness of quantities across the interface. One example is the numerical method of [69], which uses a modified pressure correction projection method with an iterative correction procedure to preserve sharpness and ensure that jump conditions were properly satisfied. This method is the basis for the method presented here, and was used to produce the results presented in chapters 2 and 3. Another recent example is the method of [21], which uses a fully implicit treatment of the jump conditions.

I developed a numerical method that is capable of simulating the multi-phase incompressible Navier-Stokes equations Eqs. (1.1)-(1.3) with higher accuracy in the velocity field, to be able to study applications such as rising oil droplets in stratified flows. This is a projection method where all of the computational variables are collocated at the nodes of the computational mesh, instead of using the more common staggered grid layout. This avoids numerous expensive interpolations between different grid locations, avoids needing an iterative correction step that introduces additional error, and allows for the use of a more accurate nodal solver, similar to the supra-convergent solver [47], during the projection operation step. Even though this fully collocated projection is not a true projection, it is stable and approaches a true projection; if iterated a sufficient number of times.

This chapter is organized as follows: First, I present an overview on the background of the various components of this numerical method. Next, I describe a projection numerical method for single-phase incompressible fluid flow that is entirely collocated. This method was developed primarily by fellow graduate students Matt Blomquist and Scott West, though I contributed to the conceptualization of this method, in particular to the structure of the collocated projection operator. This numerical method is the subject of a

publication [14], on which I serve as a contributing author. This method and the collocated projection operator in particular that it features forms the basis of the collocated projection method that I developed for multi-phase fluid flow, which is the subject of chapter 5.

4.1 Background

4.1.1 Projection method

Fractional step or splitting methods are among the most common methods used to solve the incompressible Navier-Stokes equations. The chief example is the projection method, first proposed by Chorin [22], in which the momentum equation (1.1) is solved first and the mass equation (1.2) is later enforced by projecting the solution of the momentum equation into the space of divergence free velocities. Care must be taken when selecting the boundary conditions for each step of this splitting, and past studies, such as the class of projection methods known as Gauge Methods, have found choices on the boundary conditions to lead to fully second order methods for the velocity and pressure [18].

Another class of projection methods is approximate projection methods [3], in which the discrete gradient and divergence operators do not lead to a true projection into the space of divergence free velocity fields. Instead, the projected velocity field only has approximately zero divergence to second order in the mesh spacing. This approach allows for velocity components to be collocated, allowing for easier implementation of higher order upwinding time-stepping schemes, enabling the numerical method to achieve second order convergence in both space and time. The approach we take is a type of approximate projection method, in which the projection step is not a true projection discretely, but converges to one under repeated iteration. We select boundary conditions for the splitting steps by an iterative correction procedure.

4.1.2 Adaptive mesh refinement

In addition, because of the inherent multi-scale nature of multi-phase incompressible fluid flows, adaptive mesh refinement (AMR) is used in order to achieve computational efficiency. One of the earliest examples of AMR is the Block Structure AMR [10], where finer grids are adaptively placed over a coarse grid covering the domain. Block Structure AMR is popular in simulating the Navier-Stokes equations because each of the separately layered grids are allowed to be uniform, simplifying the construction of discretization methods.

Another popular technique is the use of octree/quadtree grids [55], which have been used in numerous numerical methods for simulating both single and multi-phase incompressible fluid flows [51, 29, 69]. Octree/quadtree grids are refined by recursively splitting computational cells into four (2D) or eight (3D) equally sized smaller cells. In Block Structure AMR, all quantities are defined on multiple grids that are synchronized for consistency, whereas in octree/quadtree grids, all quantities are defined on a single grid. In this work, we choose to use octree/quadtree grids to incorporate AMR into our method, as they are straightforward to work with given the need to only handle quantities on a single grid.

4.1.3 Interface representation

Additionally, when simulating incompressible fluid flow a choice needs to be made about how to simulate the irregular geometries that represent solid objects or moving interfaces. One common technique for handling the representation of moving irregular geometries are Front Tracking methods [27], in which the flow field is discretized on a stationary grid and the interface is explicitly represented by a separate, unstructured grid that moves through the stationary grid. This technique has been used very successfully in many applications [72], but has the drawback of required an additional unstructured grid to couple with the flow grid.

Another widely used method for tracking an interface sharply is the Level-Set method [48], in which irregular geometries are tracked through the advection of an implicit, higher dimensional level-set function which is discretized on the same grid as the fluid. The Level-Set method is widely used to simulate incompressible fluid flows [29, 69], but has inherent mass loss issues. The mass loss issue has been mitigated through coupling the level-set method with the Reference Map method [50, 2, 76], a method in which advection of a quantity of interest is achieved through considering the deformation of the entire computational domain rather than the single quantity itself. We choose to not only use a coupled Level-Set/Reference Map [9] in this work for interface representation, but also incorporate an additional, volume-preserving projection of the reference map to improve volume conservation [68], see section 5.5. In this way, we capture the sharpness of quantities across the interface and preserve volume well without needing an additional grid to track the evolution of the interface.

4.1.4 Storage of computational variables

Finally, when using a projection method, consideration must be given on where computational variables are stored on the computational mesh, in particular, the velocities,

pressure, and any intermediate variables used to project the momentum equation velocity into the space of divergence free velocities. Analytically, the gradient operator is equal to the negative transpose of the divergence operator (*i.e.* $\nabla = -\nabla^{\mathrm{T}}\cdot$). For a detailed proof of this property, see Appendix A. If this is also true for the chosen discrete gradient operator, $\mathcal{G}$, and the discrete divergence operator, $\mathcal{D}$, (*i.e.* $\mathcal{G} = -\mathcal{D}^{\mathrm{T}}$), then the projection step will also be stable at the discrete level. On a uniform Cartesian grid, this property is not satisfied if all variables are collocated and standard central differencing is used to discretize the gradient and divergence operations, and this construction is in fact unstable. As such, a staggered grid or Marker and Cell (MAC) grid [30] is often used, where velocities are stored at cell faces and the pressure at the cell centers. In this layout, the chosen discrete divergence can be shown to be the negative transpose of the chosen discrete gradient ($\mathcal{G} = -\mathcal{D}^{\mathrm{T}}$), as has been shown in numerous works such as in [30] on uniform Cartesian grids and in [29, 69] on adaptive octree/quadtree grids. A figure of the MAC grid layout on a quadtree grid is shown in figure 4.1.

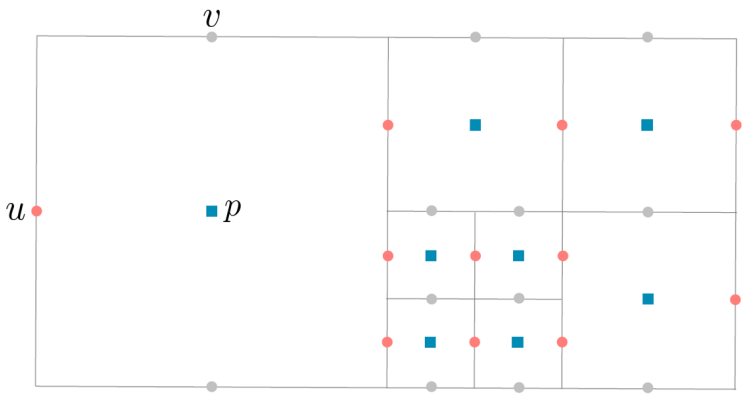

Figure 4.1: MAC grid layout on a quadtree grid in 2D. The red circles represent the x-direction velocity, u, the grey circles represent the y-direction velocity, v, and the blue squares represent the pressure, p.

The MAC grid does introduce some challenges, particularly when applied to octree/quadtree grids. Constructing differential operators that satisfy the negative transpose property is a highly non-trivial process on non-uniform grids. For example, in [29], the authors developed a stable projection method for the incompressible Navier-Stokes equations on octree/quadtree grids with the MAC grid layout. This layout led to the need for a Voronoi-based Finite-Volume approach to deal with the spatial discretization of the momentum equation and expensive Least Squares interpolations for the temporal discretization of the momentum equation. Expensive interpolations are

needed in order to perform calculations between quantities stored at different locations, which can introduce error as well as increase computational cost. This method was later generalized to simulate multi-phase fluid flows in [69], the method used to present preliminary results in chapter 2 of this dissertation.

Other studies have found success using different storage approaches than the MAC grid. For example, in [28] a different staggered grid layout was developed and used in which the pressure was stored at the nodes and velocity stored at the cell faces. This layout had several advantages over the standard MAC grid such as allowing for finite difference methods to be used given the better alignment of the pressure and velocity components. Others have found success with using a fully collocated approach, such as in [46], in which the authors created an orthogonal projection operator entirely defined on nodes, which was shown to be stable. This method also produced second order accuracy in the velocity field as it was able to take advantage of a supra-convergent solver [47] during the projection operation step. However, their method only allowed for the boundary condition of the velocity in the normal direction being zero, limiting the number of applications that could be simulated.

In this work, we wish to develop a numerical method that can produce more accurate velocity fields than our existing method [69] in order to be able to fully study rising oil droplets in stratified flows. To this end, we choose to instead collocate all of our variables at the nodes of the computational domain, instead of utilizing the MAC grid layout. This choice is made to create a numerical method with higher accuracy in the velocity field and a considerably simplified implementation.

4.2 Nodal numerical method for single-phase flow

The content of this section has been published in Blomquist, M., West, S. R., Binswanger, A. L., & Theillard, M. (2024). Stable nodal projection method on octree grids. Journal of Computational Physics, 499, 112695. https://doi.org/10.1016/j.jcp.2023.112695

Here, I summarize the contributions I made on the development of a collocated projection method to simulate the single-phase incompressible Navier-Stokes equations. This is a project that I collaborated on with fellow graduate students Matt Blomquist and Scott West. As they were developing this method, I was simultaneously developing the extension of this method to multi-phase fluid flows, which is the subject of chapters 5 - 6. My contribution to this project was assisting in the development of the collocated projection operator, as it its the same for both single and multi-phase formulations and in writing, editing, and providing feedback as a contributing author for the publication [14]

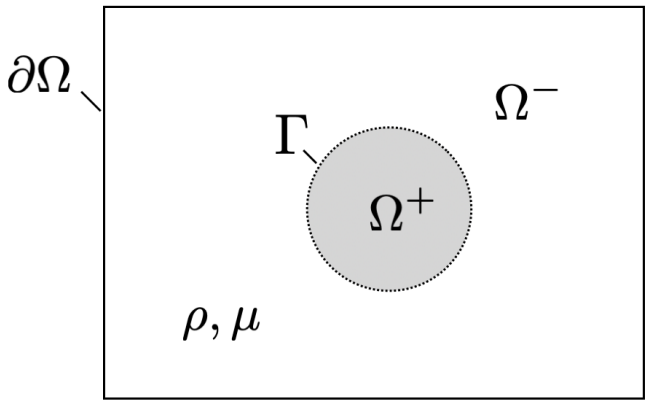

Figure 4.2: Computational domain shown in two dimensions. The fluid domain, Ω^-, is enclosed by the domain boundary, $\partial\Omega$, and the interface, Γ. Fluid properties, ρ and μ, are constant throughout the fluid domain. An arbitrary solid domain, Ω^+ is shown as a shaded region.

on this method.

This numerical method served as the foundation on which the collocated multi-phase projection method was established, and the methods and results developed in this work were extended to multi-phase flow in chapters 5-6.

4.2.1 Model equations

We consider a fluid flow in a computational domain $\Omega = \Omega^+ \cup \Omega^- \subset \mathbb{R}^{2,3}$ whose dynamics are modeled by the incompressible Navier-Stokes equations

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\Delta\mathbf{u} + \mathbf{f} \quad \forall \mathbf{x} \in \Omega^-, \tag{4.1}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \forall \mathbf{x} \in \Omega^-, \tag{4.2}$$

where $\mathbf{u}$ is the fluid velocity, p is the pressure, ρ is the constant density, μ is the constant viscosity, and $\mathbf{f}$ are external forces such as the gravitational force. By convention, the Navier-Stokes equations (4.1)-(4.2) are specifically solved in the subdomain Ω^-, see figure 4.2. We denote the boundary of Ω^- by Γ and the boundary of the computational domain Ω by $\partial\Omega$.

Additionally, we consider a no slip boundary condition for any solid object placed in

the flow, denoted mathematically as

$$\mathbf{u} = \mathbf{u}|_{\Gamma} \quad \forall \mathbf{x} \in \Gamma, \tag{4.3}$$

where $\mathbf{u}|_{\Gamma}$ is equal to the velocity of the solid object.

4.2.2 Computational grids

The equations presented here will be solved on non-graded adaptive octree/quadtree grids. In an adaptive octree/quadtree grid, each cell is split recursively as desired into either four cells in a quadtree or eight cells in an octree. The number of times a cell is split is called its level. In a non-graded tree, there are no restrictions on the differences in level between adjacent cells [55].

On the octree/quadtree grids, all computational variables, namely the velocity components and pressure, will be fully collocated and stored at the nodes of the grid. An example of a highly non-graded quadtree grid along with a schematic of where variables are stored is shown in figure 4.3.

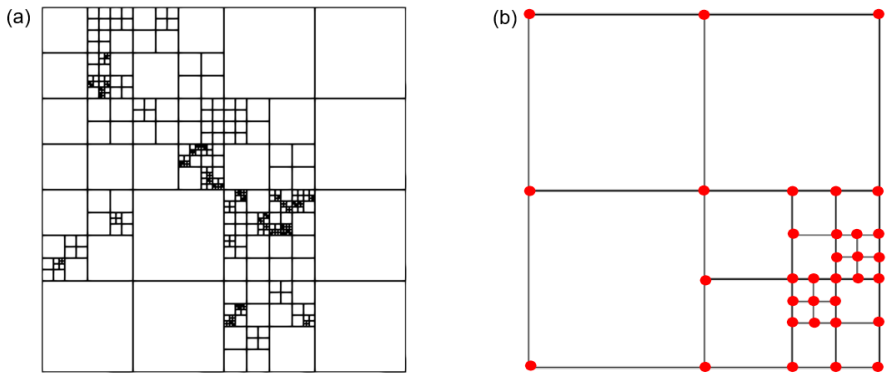

Figure 4.3: (a) Highly non-graded quadtree grid, (b) Example of fully collocated quadtree grid. The velocity components and pressure are stored at the nodes, represented in red.

4.2.3 General Projection Method

In this section, the general steps of the projection method [22] are detailed, as this numerical method for incompressible single-phase fluid flow is constructed as an approximate projection method.

The projection method is a fractional-step scheme for solving Eqs. (4.1) and (4.2). In the first step, referred to in this work as the viscosity step, we advance the velocity field $\mathbf{u}^n$ at time step t_n to an intermediate velocity field $\mathbf{u}^*$ by solving the momentum equation Eq. (4.1), where the pressure term is omitted. The viscosity step, using a second order, semi-implicit, Semi-Lagrangian, Backward Difference Formula scheme, has $\mathbf{u}^*$ constructed as the solution of

$$\rho\left(\alpha\frac{\mathbf{u}^* - \mathbf{u}_d^n}{\Delta t_n} + \beta\frac{\mathbf{u}_d^n - \mathbf{u}_d^{n-1}}{\Delta t_{n-1}}\right) = \mu\Delta\mathbf{u}^* + \mathbf{f}, \tag{4.4}$$

where α and β are coefficients given in terms of time steps t_n and t_{n-1}, $\Delta t_n = t_{n+1} - t_n$ and $\Delta t_{n-1} = t_n - t_{n-1}$, $\mathbf{u}_d^n$ and $\mathbf{u}_d^{n-1}$ are the velocities $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ ($\mathbf{u}$ at time steps t_n and t_{n-1}) evaluated at the departure points $\mathbf{x}_d^n$ and $\mathbf{x}_d^{n-1}$ respectively, which are the departure points that follow to $\mathbf{u}^{n+1}$ at the point $\mathbf{x}^{n+1}$ if the corresponding characteristic is traced back in time from t_{n+1} to t_n and t_{n-1}. The details of this temporal discretization will be elaborated on in chapter 5.

The next step of the projection method is to take the intermediate velocity $\mathbf{u}^*$ and project it onto the divergence-free space to enforce the incompressibility condition Eq. (4.2). This step is referred to as the projection step. This is done by first using the Helmholtz-Hodge decomposition to separate the intermediate velocity into curl-free and divergence-free components as

$$\mathbf{u}^* = \mathbf{u}^{n+1} + \nabla\Phi \tag{4.5}$$

where $\mathbf{u}^{n+1}$ is the divergence-free velocity field at time t_{n+1} and Φ is the Hodge variable. Taking the divergence of Eq. (4.5) and using the fact that $\mathbf{u}^{n+1}$ is divergence free, we obtain a Poisson equation for the Hodge variable,

$$\Delta\Phi = \nabla\cdot\mathbf{u}^*. \tag{4.6}$$

This Poisson equation is solved for Φ and the divergence-free velocity $\mathbf{u}^{n+1}$ at time t_{n+1} is recovered as

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \nabla\Phi. \tag{4.7}$$

We note that Eq. (4.7) can be rewritten as an operator applied to $\mathbf{u}^*$,

$$\mathbf{u}^{n+1} = \left(\mathcal{I} - \nabla\Delta^{-1}\nabla\cdot\right)\mathbf{u}^*. \tag{4.8}$$

Thus, we define the general projection operator P as,

$$P = \mathcal{I} - \nabla\Delta^{-1}\nabla\cdot. \tag{4.9}$$

4.2.4 Stable collocated projection operator

In the numerical methods presented in this work, the discrete collocated projection operator $\mathcal{P}_N$ is defined as

$$\mathcal{P}_N = \mathcal{I} - \mathcal{G}_N \mathcal{L}_N^{-1} \mathcal{D}_N, \tag{4.10}$$

where $\mathcal{G}_N$, $\mathcal{D}_N$, and $\mathcal{L}_N$ are the discrete nodal gradient, divergence, and Laplacian operators, respectively, constructed using standard central differences [14, 47]. This operator is used to perform the projection step in the numerical methods presented in this work.

As we established in our manuscript [14], the discrete collocated projection operator $\mathcal{P}_N$ is not actually a true projection. This is due to the fact that the composition of the discrete nodal divergence and gradient operators is not equivalent to the discrete nodal Laplacian operator (*i.e.* $\mathcal{D}_N \mathcal{G}_N \neq \mathcal{L}_N$). As a result,

$$\begin{aligned} \mathcal{P}_N^2 &= (\mathcal{I} - \mathcal{G}_N \mathcal{L}_N^{-1} \mathcal{D}_N)(\mathcal{I} - \mathcal{G}_N \mathcal{L}_N^{-1} \mathcal{D}_N) \\ &= \mathcal{I} - 2\mathcal{G}_N \mathcal{L}_N^{-1} \mathcal{D}_N + \mathcal{G}_N \mathcal{L}_N^{-1} \mathcal{D}_N \mathcal{G}_N \mathcal{L}_N^{-1} \mathcal{D}_N \neq \mathcal{P}_N. \end{aligned} \tag{4.11}$$

However, if the operator $\mathcal{P}_N$ is applied to a velocity field $\mathbf{u}$ iteratively as

$$\lim_{k \to \infty} \mathcal{P}_N^k \mathbf{u}, \tag{4.12}$$

then it was shown that $\mathcal{P}_N$ dampens all compressible modes of $\mathbf{u}$ and keeps only the incompressible ones, meaning that the collocated projection operator $\mathcal{P}_N$ converges to a true projection when iterated sufficiently many times [14]. In practice, we found that only a handful of projections, generally fewer than 10, were enough to produce accurate results in a range of validation examples.

Furthermore, in our manuscript [14], we prove the stability of the collocated projection analytically on uniform grids with periodic boundary conditions, and numerically on adaptive grids for a range of other boundary conditions. The stability of the collocated projection operator in the context of two-phase flow will be explored later in chapter 5.

4.2.5 Numerical method

The numerical method is a fully collocated approximate projection method where the viscosity and projection steps are successively iterated until the prescribed boundary conditions on $\mathbf{u}$ are sufficiently satisfied on the velocity at the next time step, $\mathbf{u}^{n+1}$, and where the projection step is itself successively iterated until the compressible modes are sufficiently removed from the velocity computed in the viscosity step. We use the

projection method presented in section 4.2.3 with the iterative approach shown in section 4.2.4 to carry out the projection step Eq. (4.8). The interface representing objects in the domain is expressed as a level-set function. The algorithm used to construct the solution $\mathbf{u}^{n+1}$ at time t_{n+1} from the solution $\mathbf{u}^n$ at time t_n is described in figure 4.4.

---

1 – Initialization

Initialize the corrective velocity boundary condition $\mathbf{c}$ as the final one from the previous iteration.

2 – Repeat until $\left\|\mathbf{u}^{n+1} - \mathbf{u}|_\Gamma\right\|_\infty < \epsilon_o$

2a – Viscosity step

Compute the intermediate velocity field $\mathbf{u}^*$ as the numerical solution of

$$\rho\frac{D\mathbf{u}^*}{Dt} = \mu\triangle\mathbf{u}^* + \mathbf{f} \qquad \forall\mathbf{x}\in\Omega^-/\Gamma, \tag{4.13}$$

$$\mathbf{u}^* = \mathbf{u}|_\Gamma + \mathbf{c} \qquad \forall\mathbf{x}\in\Gamma, \tag{4.14}$$

Initialize $\mathbf{u}^{n+1} = \mathbf{u}^*$

2b – Projection step

Repeat until convergence by iterating in p , $\left\|\mathbf{u}_p^{n+1} - \mathcal{P}_N\mathbf{u}_p^{n+1}\right\|_\infty < \epsilon_i \left\|\mathbf{u}_p^{n+1}\right\|_\infty$

Project the velocity field

$$\mathbf{u}_{p+1}^{n+1} = \mathcal{P}_N\mathbf{u}_p^{n+1} \tag{4.15}$$

By computing the Hodge variable Φ_p as the solution of

$$\boldsymbol{\Delta}\Phi_p = \nabla\cdot\mathbf{u}_p^{n+1} \qquad \forall\mathbf{x}\in\Omega^-/\Gamma, \tag{4.16}$$

$$\nabla\Phi_p\cdot\mathbf{n} = 0 \qquad \forall\mathbf{x}\in\Gamma, \tag{4.17}$$

and using it to project the intermediate velocity on the divergence-free space:

$$\mathbf{u}_{p+1}^{n+1} = \mathbf{u}_p^{n+1} - \nabla\Phi_p. \tag{4.18}$$

Compute the new correction

$\mathbf{c} = \mathbf{c} - \omega\left(\mathbf{u}^{n+1} - \mathbf{u}|_\Gamma\right)$

3 – Update

Adapt the mesh to $\mathbf{u}^{n+1}$ and update all the variables accordingly.

Figure 4.4: Outline of the single-phase flow algorithm for the construction of the solution $\mathbf{u}^{n+1}$ at time t_{n+1} from the solution $\mathbf{u}^n$ at the previous time step t_n.

We advance our our velocity field from $\mathbf{u}^n$ to $\mathbf{u}^{n+1}$ using a standard two step projection method. In the viscosity step, we compute the intermediate velocity field, $\mathbf{u}^*$, using a semi-Lagrangian Backward Difference Formula scheme for the temporal integration of the

momentum equation where the viscous terms are treated implicitly. The details of this discretization are described in chapter 5, as this discretization is shared between the single-phase and multi-phase numerical methods.

In the projection step, we repeatedly apply our nodal projection operator until the stopping criteria

$$\left\|\mathbf{u}^{n+1} - \mathcal{P}_N \mathbf{u}^{n+1}\right\|_\infty < \epsilon_i \left\|\mathbf{u}^{n+1}\right\|_\infty, \tag{4.19}$$

where ϵ_i is a small, positive number, is satisfied, or a predefined maximum number of iterations, K_{max}, has been reached. Typically, we choose $\epsilon_i = 10^{-3}$, set $K_{max} = 5$, and only need a few iterations to reach convergence.

To ensure that $\mathbf{u}^{n+1}$ satisfies the prescribed boundary conditions on $\mathbf{u}$, we iterate the viscosity and projection steps using a boundary correction procedure. In general, the boundary conditions of $\mathbf{u}^{n+1}$ will not satisfy the prescribed boundary conditions on $\mathbf{u}$, such as Eq. (4.3), because $\mathbf{u}^{n+1}$ is constructed entirely from $\mathbf{u}^*$ and $\nabla\Phi$, as seen in Eq. (4.7). To satisfy the correct boundary conditions on $\mathbf{u}^{n+1}$, we choose the boundary conditions on $\mathbf{u}^*$ to be the prescribed boundary conditions on $\mathbf{u}$ plus a corrective velocity $\mathbf{c}$, shown in Eq. (4.14). This corrective velocity $\mathbf{c}$ is initialized at $t = 0$ as $\mathbf{c} = 0$ and as the final $\mathbf{c}$ from the previous time step for later times, and at each time step is updated for every iteration of the viscosity and projection steps. The corrective velocity $\mathbf{c}$ is updated as

$$\mathbf{c} = \mathbf{c} - \omega \left(\mathbf{u}^{n+1} - \mathbf{u}|_\Gamma\right), \tag{4.20}$$

where $\omega \in (0, 1)$. It is designed so that when the correction reaches convergence, the boundary condition on the solid object is satisfied (*i.e.* $\mathbf{u}^{n+1} = \mathbf{u}$ on Γ). The parameter ω controls the convergence rate and must be chosen in the range $0 < \omega < 1$. Similar corrections are performed on the wall of the computational domain $\partial\Omega$. Typically, we terminate the outer iterations when the error in the interface's velocity ($\|\mathbf{u}^{n+1} - \mathbf{u}|_\Gamma\|_\infty$) is less than $\epsilon_o = 10^{-3}$.

4.3 Conclusions

I contributed to the development of a novel projection method for the simulation of single-phase incompressible flows in arbitrary domains using quad/octrees, where all of the variables are collocated at the grid nodes. By design, our collocated projection operator is an iterative procedure. If it exists, the limit of this iterated procedure is the canonical projection on the incompressible space. This numerical method achieves second order accuracy and through our validation is shown to be a competitive computational fluid dynamics tool for studying complex fluid flows. We extended this method to

incompressible two-phase flow as part of the project presented in chapter 5, in order to have a method capable of studying rising oil droplets in density stratified flows.

Chapter 5

Stable nodal projection method for two-phase fluid flows

In this chapter, I present a fully collocated nodal projection method to solve the multi-phase incompressible Navier-Stokes equations. This numerical method extends the approach presented in chapter 4 to multi-phase fluid flow. This numerical method is specifically designed for simulating two-phase flow, though it is generalizable to more than two fluid phases.

5.1 Governing equations of incompressible two-phase fluid flow

We consider a domain Ω consisting of two incompressible, immiscible, Newtonian fluid phases: Ω^+ and Ω^-, each with a respective constant density $\rho^\pm$ and constant viscosity $\mu^\pm$. The two phases are separated by an interface Γ, upon which there is a curvature κ and interfacial tension γ. Here, the $\pm$ superscript denotes the fluid phase. A schematic of this problem domain is given in figure 5.1. The fluid velocity and pressure are modeled by the incompressible Navier-Stokes equations

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \nabla\cdot\sigma + \mathbf{f} \qquad \forall \mathbf{x} \in \Omega \setminus \Gamma, \tag{5.1}$$

$$\nabla\cdot\mathbf{u} = 0 \qquad \forall \mathbf{x} \in \Omega \setminus \Gamma, \tag{5.2}$$

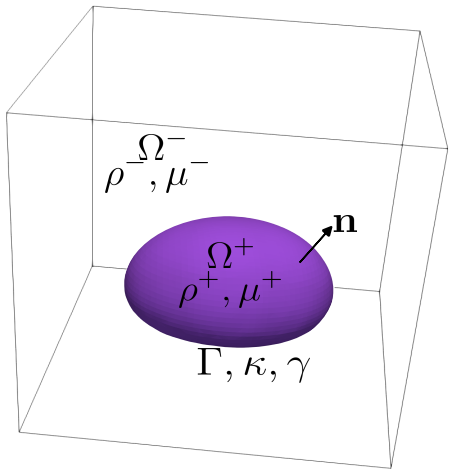

Figure 5.1: Problem schematic where the density in each phase is denoted by $\rho^{\pm}$ and the viscosity in each phase is denoted by $\mu^{\pm}$. The interface Γ has a curvature κ, normal $\mathbf{n}$, and interfacial tension γ.

where $\mathbf{u}$ is the fluid velocity, p is the pressure, $\mathbf{f}$ are external forces, $\sigma = \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)$ is the stress tensor, ρ is the density, and μ is the viscosity. The first equation Eq. (5.1) represents conservation of momentum and the second Eq. (5.2) represents conservation of mass. Jump conditions representing continuity of the velocity and traction across the interface are defined as

$$[\![\mathbf{u}]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{5.3}$$

$$[\![\sigma \cdot \mathbf{n} - p\mathbf{n}]\!] = \gamma\kappa\mathbf{n} + \mathbf{q} \qquad \forall \mathbf{x} \in \Gamma, \tag{5.4}$$

where $[\![\chi]\!] = \chi^+ - \chi^-$ is the jump of the quantity χ across the interface Γ, γ is the coefficient of interfacial tension, κ is the curvature of the interface Γ, $\mathbf{n}$ is the outward facing normal vector, and $\mathbf{q}$ are any additional interfacial stresses.

5.2 Numerical method overview

The two-phase numerical method presented in this section can be seen as a generalization of the method presented in chapter 4 to multiple fluid phases. We collocate all computational variables at the grid nodes and use adaptive quad/octree grids, as in section 4.2.2.

This approach is a traditional pressure correction projection method [22], where at every step a pressure guess needs to be constructed and the viscosity and projection steps

are iterated until the interface jump conditions and wall boundary conditions are satisfied within a desired tolerance. The projection step is a fully collocated approximate nodal projection, where the projection step is itself successively iterated until the compressible modes are sufficiently removed from the intermediate velocity computed during the viscosity step.

The interface Γ is represented as the zero contour of the scalar level-set function $\phi(\mathbf{x})$, where ϕ is a signed distance function with the following convention

$$\Gamma = \{\mathbf{x} \in \Omega \ : \ \phi(\mathbf{x}) = 0\}, \tag{5.5}$$

$$\Omega^- = \{\mathbf{x} \in \Omega \ : \ \phi(\mathbf{x}) < 0\}, \tag{5.6}$$

$$\Omega^+ = \{\mathbf{x} \in \Omega \ : \ \phi(\mathbf{x}) > 0\}, \tag{5.7}$$

see figure 5.1. The level-set function ϕ can be differentiated to compute the normal vector $\mathbf{n}$ and the curvature κ.

The algorithm used to construct the solution $(\phi^{n+1}, \mathbf{u}^{n+1})$ at time t_{n+1} from the solution $(\phi^n, \mathbf{u}^n)$ at time t_n is shown in figure 5.2, and each step of this method will be elaborated on further. In the description of each step of the numerical method, wall boundary conditions on the boundary of the domain $\partial\Omega$ will be given for each quantity being solved for. Generically, these boundary conditions will be denoted as $\mathbf{B}(\cdot)$.

In the following sections, we integrate the nodal projection operator into a general framework for solving the incompressible two-phase Navier-Stokes equations. We begin by giving an overview of the convergence criteria used for the iterations present in the numerical method. Next, we present an overview of the interface representation method we employ as part of the numerical method, followed by an overview of the adaptive mesh refinement strategy used. Subsequent sections then discuss details of the viscosity, pressure guess, and projection steps of the numerical method.

(0) Initialization of the corrective terms
Initialize the corrective velocity jump $\mathbf{X}_0$.

(1) Pressure guess
Construct the pressure guess $\widetilde{p}$ as the solution of the Poisson problem

$$\begin{aligned} \boldsymbol{\Delta}\widetilde{p} &= 0 \qquad \forall \mathbf{x} \in \Omega \setminus \Gamma, && (5.8)\\ \left[\!\left[\widetilde{p}\right]\!\right] &= -\gamma\kappa - \mathbf{q}\cdot\mathbf{n} \qquad \forall \mathbf{x} \in \Gamma, && (5.9)\\ \left[\!\left[\frac{1}{\rho}\nabla\widetilde{p}\cdot\mathbf{n}\right]\!\right] &= 0 \qquad \forall \mathbf{x} \in \Gamma. && (5.10) \end{aligned}$$

(2) Repeat until convergence by iterating in k , $\left\|\mathbf{u}_{k+1}^{n+1} - \mathbf{u}_{k}^{n+1}\right\|_\infty < \epsilon_o$

2a – Viscosity step
Compute the intermediate velocity field $\mathbf{u}_k^*$ as the solution of

$$\begin{aligned} \rho\frac{D\mathbf{u}_k^*}{Dt} = \nabla\cdot\left(\mu\left(\nabla\mathbf{u}_k^* + (\nabla\mathbf{u}_k^*)^{\mathrm{T}}\right)\right) - \nabla\widetilde{p} + \mathbf{f} \qquad \forall \mathbf{x} \in \Omega/\Gamma, && (5.11)\\ \left[\!\left[\mathbf{u}_k^*\right]\!\right] = \mathbf{X}_k \qquad \forall \mathbf{x} \in \Gamma, && (5.12)\\ \left[\!\left[\left(\nabla\mathbf{u}_k^* + (\nabla\mathbf{u}_k^*)^{\mathrm{T}}\right)\cdot\mathbf{n}\right]\!\right] = \mathbf{q} - \mathbf{n}(\mathbf{q}\cdot\mathbf{n}) \qquad \forall \mathbf{x} \in \Gamma. && (5.13) \end{aligned}$$

2b – Projection step
Initialize the velocity to be projected $\mathbf{u}_p^{n+1} = \mathbf{u}_k^*$.

Repeat until convergence by iterating in p , $\left\|\mathbf{u}_p^{n+1} - \mathcal{P}_N\mathbf{u}_p^{n+1}\right\|_\infty < \epsilon_i \left\|\mathbf{u}_p^{n+1}\right\|_\infty$
Project the velocity field

$$\mathbf{u}_{p+1}^{n+1} = \mathcal{P}_N\mathbf{u}_p^{n+1} \tag{5.14}$$

By computing the Hodge variable Φ_p as the solution of

$$\begin{aligned} \boldsymbol{\Delta}\Phi_p &= \nabla\cdot\mathbf{u}_p^{n+1} \qquad \forall \mathbf{x} \in \Omega/\Gamma, && (5.15)\\ [\![\rho\Phi_p]\!] &= 0 \qquad \forall \mathbf{x} \in \Gamma, && (5.16)\\ [\![\nabla\Phi_p\cdot\mathbf{n}]\!] &= 0 \qquad \forall \mathbf{x} \in \Gamma, && (5.17) \end{aligned}$$

and using it to project the intermediate velocity on the divergence-free space:

$$\mathbf{u}_{p+1}^{n+1} = \mathbf{u}_p^{n+1} - \nabla\Phi_p. \tag{5.18}$$

Compute the new correction $\mathbf{X}_{k+1}$ from the above velocity

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \omega\left[\!\left[\mathbf{u}_k^{n+1}\right]\!\right],\ 0 < \omega < 1. \tag{5.19}$$

(3) Interface evolution
Construct the new level set ϕ^{n+1} and the new reference map ξ^{n+1} from the reference map ξ^n and $\mathbf{u}^{n+1}$ by solving Eqs. (5.34), (5.35), and (5.37).

(4) Update
Adapt the mesh to ϕ^{n+1} and $\mathbf{u}^{n+1}$ and update all the variables accordingly.

Figure 5.2: Outline of the algorithm for the construction of the solution $(\phi^{n+1}, \mathbf{u}^{n+1})$ at time t_{n+1} from the solution $(\phi^n, \mathbf{u}^n)$ at the previous time step t_n.

5.3 Interfacial and boundary convergence conditions

The corrective term $\mathbf{X}_k$ as seen in Eq. (5.12) is used to correctly satisfy the continuity of velocity across the interface in $\mathbf{u}^{n+1}$. It is iterated such that

$$\lim_{k\to\infty} \left[\!\left[\mathbf{u}_k^{n+1}\right]\!\right] = \mathbf{0}. \tag{5.20}$$

The iteration is initialized at $t = 0$ as $\mathbf{X}_0 = 0$ and as the final $\mathbf{X}_k$ from the previous time step for later times, and at each time step is updated for every iteration of the viscosity and projection steps. The iteration procedure on $\mathbf{X}_k$ is

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \omega \left[\!\left[\mathbf{u}_k^{n+1}\right]\!\right], \tag{5.21}$$

where ω is a small, strictly positive constant. This iteration choice was shown in [69] to converge to the correct jump condition Eq. (5.3) as $k \to \infty$ for any $\omega \in (0, 1)$. We do not need an additional correction on the stress jump Eq. (5.13) like was included in [69] due to our full treatment of the stress tensor in the Navier-Stokes momentum equation Eq. (5.1).

Additionally, during this iteration process, the boundary conditions of $\mathbf{u}^*$ are continually updated and corrected in the same way as the single-phase flow method 4.4 as

$$\mathbf{B}(\mathbf{u}_k^*) = \mathbf{B}(\mathbf{u}_k^{n+1}) + \mathbf{c}, \tag{5.22}$$

where the corrective velocity $\mathbf{c}$ is computed as in Eq. (4.20).

5.4 Convergence criteria

We end the iteration of the successive viscosity and projection steps when the convergence criteria

$$\left\|\mathbf{u}_{k+1}^{n+1} - \mathbf{u}_k^{n+1}\right\|_\infty < \epsilon_o, \tag{5.23}$$

is fulfilled. In this case, ϵ_o is a small positive number, in practice typically taken to be $\epsilon_o = 10^{-3}$. This convergence criteria will be satisfied when $\mathbf{X}_{k+1} \approx \mathbf{X}_k$, as then the velocity jump condition should be satisfied within a desired tolerance, as opposed to the single-phase flow method 4.4, which terminates its iteration when the wall boundary condition is satisfied within a desired tolerance.

5.5 Interface representation

In this numerical method, we handle the interface representation between the two fluid phases using the Coupled Level-Set Reference Map Method [9], as well as the improved Volume Preserving Reference Map correction [68].

5.5.1 Level-Set method

As previously stated, the interface Γ is represented as the zero contour of the scalar level-set function $\phi(\mathbf{x})$, where ϕ is a signed distance function with the following convention

$$\Gamma = \{\mathbf{x} \in \Omega \ : \ \phi(\mathbf{x}) = 0\}, \tag{5.24}$$
$$\Omega^- = \{\mathbf{x} \in \Omega \ : \ \phi(\mathbf{x}) < 0\}, \tag{5.25}$$
$$\Omega^+ = \{\mathbf{x} \in \Omega \ : \ \phi(\mathbf{x}) > 0\}, \tag{5.26}$$

see figure 5.3. The level-set function ϕ can be differentiated to compute the normal vector

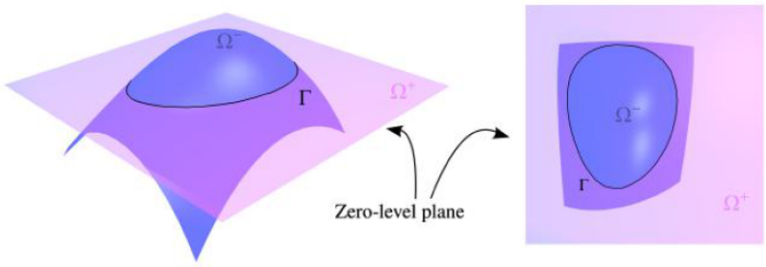

Figure 5.3: Level-set representation of interface Γ as contour of level-set function ϕ, which separates the domain into Ω^- and Ω^+ in 2D [9].

$\mathbf{n}$ and the curvature κ as

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \ \kappa = \nabla \cdot \mathbf{n} = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right). \tag{5.27}$$

To ensure that ϕ is unique, ϕ is chosen to be a signed-distance function

$$|\nabla\phi| = 1. \tag{5.28}$$

Then given an advecting velocity that is the solution to the incompressible Navier-Stokes equation Eqs. (5.1) - (5.4), the position of the interface can be tracked in time by solving the advection initial value problem

$$\frac{\partial\phi}{\partial t} + \mathbf{u} \cdot \nabla\phi = 0 \quad t \geq 0, \ \forall \mathbf{x} \in \Omega, \tag{5.29a}$$
$$\phi(t = 0, \mathbf{x}) = \phi_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega, \tag{5.29b}$$

where $\phi_0(\mathbf{x})$ is the initial level-set function at $t = 0$. As this advection is carried out, the level-set function can gradually lose the signed distance property (5.28) due to numerical error, and so the level-set function must be reinitialized by solving the reinitialization equation for a sufficient number of fictitious τ time steps:

$$\frac{\partial \phi}{\partial \tau} + \text{sign}(\phi)(|\nabla \phi| - 1) = 0 \quad \forall \mathbf{x} \in \Omega. \tag{5.30}$$

As a result of the inherent numerical errors that accumulate through the advection and reinitialization procedures, the level-set method is not conservative of mass, which can pose significant issues when using it to resolve long time dynamics of interface driven flows like the oil droplets in density stratified flows of chapter 2. As such, we choose to use a Coupled Level-Set Reference Map Method to address this mass loss issue by instead using a reference map to handle interface advection.

5.5.2 Reference map method

The Reference map method [33, 9] handles the advection of quantities in a given domain by a deforming field through an invertible mapping between the initial domain and the domain at a future time. We consider the advection of a scalar quantity ψ under the velocity $\mathbf{u}$ given by the following initial value problem

$$\frac{\partial \psi}{\partial t} + \mathbf{u} \cdot \nabla \psi = 0 \quad t \geq 0, \ \forall \mathbf{x} \in \Omega, \tag{5.31a}$$

$$\psi(t = 0, \mathbf{x}) = \psi_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega, \tag{5.31b}$$

the same as in Eq. (5.29). The Reference Map method works by deforming the entire computational domain by the advecting velocity $\mathbf{u}$ and tracking the mapping from the deformed domain at a future time $t > 0$ back to the initial domain with an invertible reference map [33, 9]. A schematic of this method is shown in figure 5.4.

We consider the original, undeformed domain at $t = 0$ to be $\mathcal{B}_0 \in \mathbb{R}^{2,3}$ and the future, deformed domain to be $\mathcal{B}(t) \in \mathbb{R}^{2,3}$. The mapping that takes any point $\mathbf{x}_0 \in \mathcal{B}_0$ and maps it to its corresponding point $\mathbf{x}(t) \in \mathcal{B}(t)$ is called the motion map, defined

$$\mathbf{x}(t) = \chi(t, \mathbf{x}_0), \ t \geq 0, \ \ \mathbf{x}_0 \in \mathcal{B}_0. \tag{5.32}$$

The reference map, denoted as $\xi(t, \mathbf{x})$, is the inverse of the motion map, and maps any point $\mathbf{x}(t) \in \mathcal{B}(t)$ to its corresponding point $\mathbf{x}_0 \in \mathcal{B}_0$,

$$\mathbf{x}_0 = \xi(t, \mathbf{x}(t)), \ t \geq 0, \ \ \mathbf{x} \in \mathcal{B}(t). \tag{5.33}$$

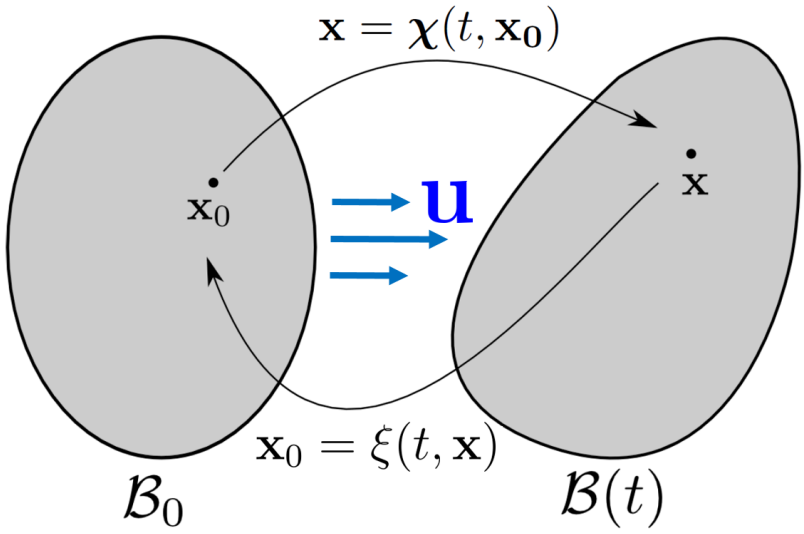

Figure 5.4: Schematic of the effect of the motion map χ and the reference map ξ in mapping points $\mathbf{x}_0 \in \mathcal{B}_0$ and $\mathbf{x} \in \mathcal{B}(t)$ between each other [9]. The deformation of the domain here is due to the advecting velocity $\mathbf{u}$.

In the reference map framework, quantities are advected by advecting the reference map itself by solving the following advection equation initial value problem

$$\frac{\partial \xi}{\partial t} + \mathbf{u} \cdot \nabla \xi = 0, \ \forall t \geq 0, \ \forall \mathbf{x} \in \mathcal{B}(t), \tag{5.34}$$

$$\xi(t = 0, \mathbf{x}) = \mathbf{x}, \ \mathbf{x} \in \mathcal{B}_0. \tag{5.35}$$

With the reference map obtained, we can solve for the advection of any scalar quantity ψ also advected by the velocity $\mathbf{u}$ by plugging the reference map into the initial condition as

$$\psi(t, \mathbf{x}) = \psi_0(\xi(t, \mathbf{x})), \ t \geq 0, \ \ \mathbf{x} \in \mathcal{B}(t). \tag{5.36}$$

5.5.3 Coupled method

In the Coupled Level-Set Reference Map method, the level-set function ϕ is advected using the reference map, following the procedure outlined in Eqs. (5.34)-(5.36) [9]. The initial domain $\mathcal{B}_0$ and the deformed domain $\mathcal{B}(t)$ are both chosen to be the computational domain at the corresponding time $t \geq 0$, and the level-set function at time $t \geq 0$ is evaluated as

$$\phi(t, \mathbf{x}) = \phi_0(\xi(t, \mathbf{x})), \ t \geq 0, \ \ \mathbf{x} \in \mathcal{B}(t). \tag{5.37}$$

In order for the Reference Map method to work, the motion map χ and thus the reference map ξ must be bijective. It is possible, especially for large deformations due to $\mathbf{u}$ [33], for ξ to gradually lose its bijectivity due to numerical error. As such, a restarting procedure is performed if ξ is shown to be losing bijectivity. The following criteria based on the gradient of ξ is evaluated at each time step in a user-defined region around the interface to see if ξ has lost bijectivity [9]:

$$\max\left\{\left\|\frac{\nabla\xi_x}{\|\xi_x\|_\infty}\cdot\frac{\nabla\xi_y}{\|\xi_y\|_\infty}\right\|_\infty,\ \left\|\frac{\nabla\xi_x}{\|\xi_x\|_\infty}\cdot\frac{\nabla\xi_z}{\|\xi_z\|_\infty}\right\|_\infty,\ \left\|\frac{\nabla\xi_y}{\|\xi_y\|_\infty}\cdot\frac{\nabla\xi_z}{\|\xi_z\|_\infty}\right\|_\infty\right\} > \cos\left(\theta_{crit}\right), \tag{5.38}$$

where ξ_x, ξ_y, and ξ_z are the components of ξ and $\theta_{crit} \in \left[0, \frac{\pi}{2}\right]$ is an arbitrary user-defined critical angle. If there is a particular time t_c where the restarting criteria is met, then the following restarting procedure is performed

$$\phi_0(\mathbf{x}) = \mathcal{R}(\phi_0(\xi(t_c, \mathbf{x}))),\ \xi(t_c, \mathbf{x}) = \mathbf{x},\ \mathbf{x} \in \mathcal{B}(t_c), \tag{5.39}$$

where $\mathcal{R}(\cdot)$ denotes the reinitialization of ϕ_0, performed by solving Eq. (5.30). In practice, far fewer reinitializations are needed in this coupled method than when directly advecting the level-set function per Eq. (5.29), resulting in better mass conservation [9]. Reinitialization of the level-set function only needs to be done during the restarting process, as opposed to every time step when directly advecting the level-set function, and when reinitialization does take place, only 20 reinitialization iterations are used in the coupled method compared to 50 in the direct level-set method.

Volume-preserving projection

In addition, our numerical method uses a volume-preserving correction to the coupled method [9]. Due to numerical error, we assume that the reference map, now defined as ξ^* and its inverse, defined as the motion map χ^*, do not preserve volume. The volume-preserving projection method [68] measures the deviation of the reference map, solved for in Eqs. (5.34) and (5.35), from a volume-preserving map and uses a correction to project the reference and motion map pairing into the space of volume-preserving diffeomorphisms.

This projection is done through composing χ^* and ξ^* with a diffeomorphism γ such that

$$\chi(\mathbf{x}_0) = \gamma(\chi^*(\mathbf{x}_0)),\ \det(\chi(\mathbf{x}_0)) = 1 \qquad \forall \mathbf{x}_0 \in \mathcal{B}_0, \tag{5.40}$$

and

$$\xi(\mathbf{x}) = \xi^*(\gamma^{-1}(\mathbf{x})),\ \det(\xi(\mathbf{x})) = 1 \qquad \forall \mathbf{x} \in \mathcal{B}, \tag{5.41}$$

where $\mathcal{B}$ is the deformed domain at any specified $t > 0$. The determinant criteria in Eqs. (5.40) and (5.41) hold at any given point in the domain for a volume-preserving map

[68]. To ensure that this projection does not impact the bijectivity of the reference map, the correction is only imposed in a thin user-defined shell $\mathcal{S}$ around the interface. It was shown in [68] using variational calculus that the correction to the reference map γ^{-1} is computed as

$$\gamma^{-1}(\mathbf{x}) = \mathbf{x} - \nabla\lambda, \tag{5.42}$$

where λ satisfies the Poisson problem

$$-\boldsymbol{\Delta}\lambda = 1 - \det(\nabla\xi^*) \qquad \forall \mathbf{x} \in \mathcal{S}, \tag{5.43}$$

$$\lambda = 0 \qquad \forall \mathbf{x} \in \partial\mathcal{S}, \tag{5.44}$$

where ξ^* is the unprojected reference map that is the solution to Eqs. (5.34) and (5.35). A schematic detailing the volume-preserving method is shown in figure 5.5. This additional correction further improves the volume conservation of the interface representation in our numerical method.

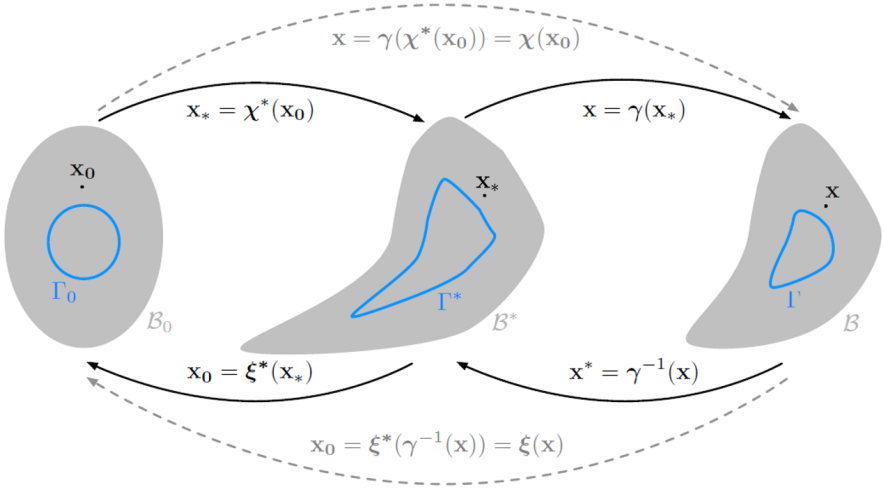

Figure 5.5: Schematic of the volume-preserving correction to the reference map. We assume that at some given time the maps (χ^*, ξ^*) are not volume-preserving due to numerical errors. These maps are corrected into a volume-preserving pair (χ, ξ) by composition with the diffeomorphism γ [68].

5.6 Sampling and data structures

Our numerical method is implemented on non-graded octree (3D) and quadtree (2D) grids. The computational grid is initialized as a single root cell that represents the entire

computational domain and is split into eight (3D) or four (2D) children cells. These children cells are then split in the same manner based on user-specified splitting criteria, with the splitting process continuing recursively on new cells until the splitting criteria are fulfilled.

As was done in previous studies [69, 14, 46, 29], the mesh is dynamically refined near the fluid-fluid interfaces and in areas of high velocity or vorticity gradients. At each iteration, we recursively apply all chosen splitting criteria at each cell.

For adaptive mesh refinement near the fluid-fluid interfaces, we split each cell $\mathcal{C}$ if the following criterion is met

$$\min_{n \in \text{nodes}(\mathcal{C})} |\phi(n)| \leq B \cdot \text{Lip}(\phi) \cdot \text{diag}(\mathcal{C}) \quad \text{and} \quad \text{level}(\mathcal{C}) \leq \max_{\text{level}}, \tag{5.45}$$

where $\text{Lip}(\phi)$ is an upper estimate of the minimal Lipschitz constant of the level-set function ϕ, $\text{diag}(\mathcal{C})$ is the length of the diagonal of cell $\mathcal{C}$, B is the user-specified width of the uniform band around the interface, where finer resolution is desired, and $\max_{\text{level}}$ is the user-specified maximum grid level. Since the level set used to create the mesh will be reinitialized and $|\nabla\phi| = 1$, we use $\text{Lip}(\phi) = 1.2$.

The refinement based criterion for areas of high velocity gradients is the following

$$\min_{n \in \text{nodes}(\mathcal{C})} \text{diag}(\mathcal{C}) \cdot \frac{\|\nabla \mathbf{u}(n)\|}{\|\mathbf{u}\|_\infty} \geq T_V \quad \text{and} \quad \text{level}(\mathcal{C}) \leq \max_V, \tag{5.46}$$

where T_V is the user-specified threshold on the velocity gradient and $\max_V$ is the maximum grid level allowed for velocity gradient-based refinement. We typically choose the velocity gradient-based refinement maximum level $\max_V$ to be one level lower than the maximum grid level $\max_{\text{level}}$ due to the most significant dynamics occuring near the interface.

Finally, we ensures that a minimum resolution of $\min_{\text{level}}$ is maintained through the following refinement criterion:

$$\text{level}(\mathcal{C}) \leq \min_{\text{level}}. \tag{5.47}$$

If none of these criteria are met, we merge $\mathcal{C}$ by removing all its descendants.

5.6.1 Collocated nodal grid layout

In figure 5.6 we illustrate the grid layout of all quantities of interest for the numerical method (velocity and the Hodge variable) on adaptive quad/octree grids. Like in our numerical method for incompressible single-phase flow [14], we collocate all quantities at the nodes of our computational grid. This results in a stable, approximate projection method that achieves second order convergence in the velocity.

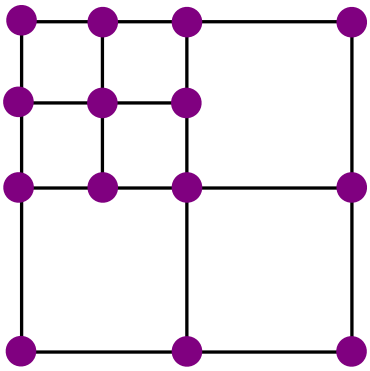

Figure 5.6: Collocated nodal grid in 2D: the velocity components, pressure, Hodge variable, level-set function, and reference map are all stored at the nodes (●) of the grid.

5.7 Viscosity step: temporal discretization

We choose to treat the advective terms in the Navier-Stokes momentum equation Eq. (5.11) explicity and the diffusive terms implicitly. This allows us to employ different strategies for the temporal and spatial discretizations. In this section, we discuss the temporal discretization of the Navier-Stokes momentum equation Eq. (5.11).

5.7.1 Phase accounting SLBDF

The left hand side of Eq. (5.11) is discretized using a combination of a second order Semi-Lagrangian (SL) method with a second order Backward Difference Formula (BDF) scheme [41, 29, 69] with adaptive time steps. This SLBDF scheme leads to the following discretization of the the momentum equation Eq. (5.11):

$$\rho\left(\alpha\frac{\mathbf{u}^{*}-\mathbf{u}_{d}^{n}}{\Delta t_{n}}+\beta\frac{\mathbf{u}_{d}^{n}-\mathbf{u}_{d}^{n-1}}{\Delta t_{n-1}}\right)=-\nabla\widetilde{p}+\nabla\cdot\left(\mu\left(\nabla\mathbf{u}^{*}+(\nabla\mathbf{u}^{*})^{\mathrm{T}}\right)\right)+\mathbf{f}, \tag{5.48}$$

where $\Delta t_n = t_{n+1} - t_n$ and $\Delta t_{n-1} = t_n - t_{n-1}$ are the adaptive time steps and where

$$\alpha=\frac{2\Delta t_{n}+\Delta t_{n-1}}{\Delta t_{n}+\Delta t_{n-1}},\ \beta=-\frac{\Delta t_{n}}{\Delta t_{n}+\Delta t_{n-1}}. \tag{5.49}$$

The terms $\mathbf{u}_d^n$ and $\mathbf{u}_d^{n-1}$ are the velocities $\mathbf{u}^n$ and $\mathbf{u}^{n-1}$ evaluated at the departure points $\mathbf{x}_d^n$ and $\mathbf{x}_d^{n-1}$ respectively, which are the departure points that follow to $\mathbf{u}^{n+1}$ at the point $\mathbf{x}^{n+1}$ if the corresponding characteristic is traced back in time from t_{n+1} to t_n and t_{n-1}. We find

the departure points $\mathbf{x}_d^n$ and $\mathbf{x}_d^{n-1}$ through the backward RK2 scheme

$$\hat{\mathbf{x}} = \mathbf{x}^{n+1} - \frac{\Delta t_n}{2}\mathbf{u}^{n+1}(\mathbf{x}^{n+1}), \tag{5.50}$$

$$\hat{\mathbf{u}} = \left(1 + \frac{\Delta t_n}{2\Delta t_{n-1}}\right)\mathbf{u}^n(\hat{\mathbf{x}}) - \frac{\Delta t_n}{2\Delta t_{n-1}}\mathbf{u}^{n-1}(\hat{\mathbf{x}}), \tag{5.51}$$

$$\mathbf{x}_d^n = \mathbf{x}^{n+1} - \Delta t_n \hat{\mathbf{u}}, \tag{5.52}$$

and

$$\hat{\mathbf{x}} = \mathbf{x}^{n+1} - \Delta t_n \mathbf{u}^n(\mathbf{x}^{n+1}), \tag{5.53}$$

$$\hat{\mathbf{u}} = \mathbf{u}^{n+1}(\hat{\mathbf{x}}), \tag{5.54}$$

$$\mathbf{x}_d^{n-1} = \mathbf{x}^{n+1} - (\Delta t_n + \Delta t_{n-1})\hat{\mathbf{u}}. \tag{5.55}$$

The interpolations of the velocity fields at the interpolation points are done using a stable weighed essentially non-oscillatory (WENO) scheme, maintaining a second order accurate discretization.

We use the improved trajectory reconstruction formulated in our nodal projection method for single phase flows [14] and replace the $\mathbf{u}^{n+1}$ in Eqs. (5.50)-(5.55) with the expansion

$$\mathbf{u}^{n+1}(\mathbf{x}^{n+1}) = \mathbf{u}^n(\mathbf{x}^{n+1}) + \frac{\Delta t_n}{\Delta t_{n-1}}\left(\mathbf{u}^n(\mathbf{x}^{n+1}) - \mathbf{u}^{n-1}(\mathbf{x}^{n+1})\right) + \mathcal{O}(\Delta t^2), \tag{5.56}$$

instead of simply using $\mathbf{u}^n$ to represent the unknown $\mathbf{u}^{n+1}$.

We additionally note that care is taken to ensure that for each point $\mathbf{x}^{n+1}$, the fluid phase that $\mathbf{x}^{n+1}$ belongs to is checked and the corresponding velocity field is used for $\mathbf{u}^n$ to construct the intermediate points $\hat{\mathbf{x}}$. The procedure is also done to ensure that once each $\hat{\mathbf{x}}$ is computed, its corresponding fluid phase is checked and the the corresponding velocity field is used for $\mathbf{u}^n$ to construct the departure points $\mathbf{x}_d^n$ and $\mathbf{x}_d^{n-1}$. This phase accounting is done by taking into account the value of the level set function at all evaluation points and using the corresponding velocity field depending on if the level set function at each evaluation point is positive or negative.

5.7.2 Local temporal limiter

Although the semi-Lagrangian and second order backward difference formula schemes are unconditionally stable independently, it is not true that the SLBDF scheme presented above is also unconditionally stable for all fields $\mathbf{u}$, see [16, 15] and the references therein.

For our work, we use quadratic interpolation in the construction of our finite difference and finite volume stencils, which can introduce spurious oscillations in regions of high curvature, leading to instabilities. For a means of visualizing this, consider the example of advection in 1D with no viscosity and a sharply peaked initial profile. The SLBDF scheme applied to this scenario rapidly blows up, as shown in figure 5.7.

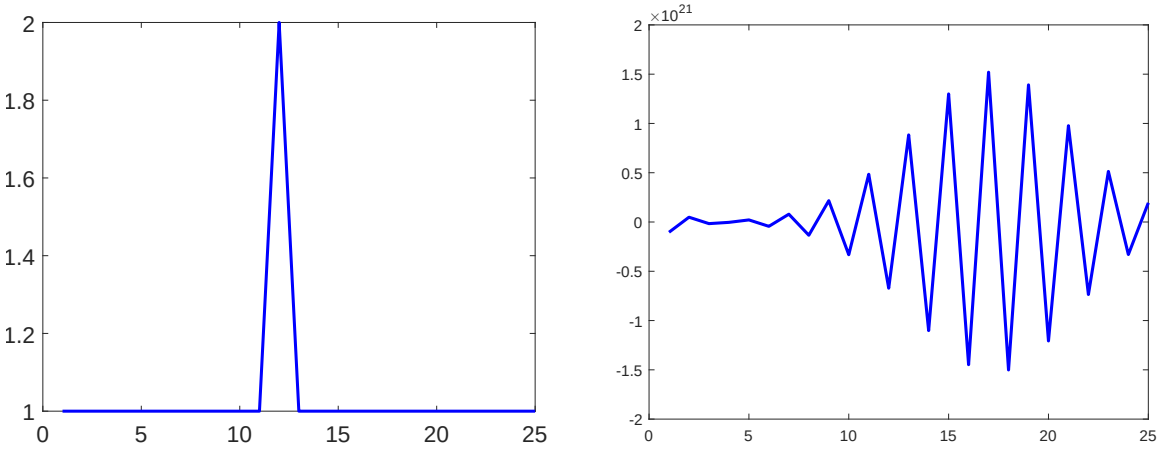

Figure 5.7: Rapid blow up of the SLBDF scheme for 1D advection with no viscosity and a sharply peaked initial profile. The left plot shows the solution at the initial time and the right plot shows the solution at a later time after blow up has occured.

However, the first order Semi-Lagrangian Backward Difference Formula discretization of the momentum equation:

$$\rho\left(\frac{\mathbf{u}^{n+1}-\mathbf{u}_d^n}{\Delta t_n}\right) = -\nabla\widetilde{p} + \nabla\cdot\left(\mu\left(\nabla\mathbf{u}^{n+1} + (\nabla\mathbf{u}^{n+1})^T\right)\right) + \mathbf{f}, \tag{5.57}$$

is unconditionally stable [75]. As such, to avoid a potential instability, we included the ability to use a local limiter which drops the SLBDF scheme to first order should it appear that the velocity at a particular node on the computational grid at the current time would be unstable under the second order SLBDF scheme.

The criteria that the local limiter uses in 2D is as follows. Given Eq. (5.48), the following quantity is computed for each velocity component

$$\mathrm{SL}_{\mathbf{u}} = \left(1 - \frac{\beta}{\alpha}\frac{\Delta t_n}{\Delta t_{n-1}}\right)\mathbf{u}_d^n - \frac{\beta}{\alpha}\frac{\Delta t_n}{\Delta t_{n-1}}\mathbf{u}_d^{n-1}. \tag{5.58}$$

After this, a check is made to see if any of these quantities exceeds the maximum or is smaller than the minimum of the velocity at the departure points, *i.e.*

$$\text{if } \mathrm{SL}_{\mathbf{u}} > \mathrm{MAX}(\mathbf{u}_d^n, \mathbf{u}_d^{n-1}) \text{ or if } \mathrm{SL}_{\mathbf{u}} < \mathrm{MIN}(\mathbf{u}_d^n, \mathbf{u}_d^{n-1}). \tag{5.59}$$

If this criteria is met for a particular velocity component at that particular node, then the discretization used for that component at that node is the first order scheme Eq. (5.57) to ensure stability. In practice, this temporal limiter is a crucial element in ensuring the robustness of our solver, and is used at nodes near the interface in cases where significant deformation of the interface occurs (see chapter 6 for examples of rising bubbles in 3D where the local temporal limiter is used to maintain numerical stability).

5.7.3 Time step restriction

For the momentum equation Eq. (5.11), following the numerical method [69], the time step restriction we use is the following

$$\Delta t \leq \Delta t_{\text{max}} = \text{MIN}(\Delta t_{\text{CFL}}, \Delta t_{\text{GV}}). \tag{5.60}$$

The first term in Eq. (5.60) is from the CFL condition, that being

$$\Delta t_{\text{CFL}} = \frac{c_0 \Delta x}{\max \|\mathbf{u}\|_\infty}, \tag{5.61}$$

where c_0 is the pre-set CFL number and Δx is the smallest spatial grid size on the entire quad/octree grid. Given that our method uses a Semi-Lagrangian method for handling advection, the CFL number c_0 in general can be chosen larger than 1. The second term in Eq. (5.60) is the generalization of the time step restriction of Galusinski and Vigneaux [26] presented in [69], that being

$$\Delta t_{\text{GV}} = \frac{c_1 \mu \Delta x}{\gamma} + \sqrt{\left(\frac{c_1 \mu \Delta x}{\gamma}\right)^2 + c_2 \frac{\rho \Delta x^3}{2\pi\gamma}}, \tag{5.62}$$

where c_1 and c_2 are constant pre-set coefficients that typically are set such that $0 < c_1, c_2 < 1$. This time step restriction is a generalization of the Brackbill stability condition [17], which is the time scale necessary to resolve capillary wave propagation. We also multiply the maximum allowable time step Δt_{max} from Eq. (5.60) by a coefficient t_{GV}, referred to as the GV coefficient, and obtain a time step $\Delta t = t_{\text{GV}} \Delta t_{\text{max}}$. To ensure numerical stability, we restrict $t_{\text{GV}} \in (0, 1]$, though for some examples we may choose a GV coefficient larger than 1 and still maintain stability (see chapter 6).

5.8 Viscosity step: spatial discretization

In this section, we discuss the implicit treatment of the diffusive terms in the Navier-Stokes momentum equation Eq. (5.11). We use a finite volume discretization to treat the stress tensor.

5.8.1 Coupled Jump Solver

With the explicit terms computed in Eq. (5.48), we are left with needing to solve for $\mathbf{u}^*$. This equation can be posed as a Helmholtz type equation of the form

$$\eta\mathbf{v} - \nabla\cdot\sigma = \mathbf{r} \qquad \forall\mathbf{x}\in\Omega\setminus\Gamma, \tag{5.63}$$

$$[\![\mathbf{v}]\!] = \mathbf{k} \qquad \forall\mathbf{x}\in\Gamma, \tag{5.64}$$

$$[\![\sigma\cdot\mathbf{n}]\!] = \mathbf{h} \qquad \forall\mathbf{x}\in\Gamma, \tag{5.65}$$

where

$$\sigma = \mu\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathrm{T}}\right), \tag{5.66}$$

$$\eta = \frac{\rho\alpha}{\Delta t_n}, \tag{5.67}$$

and $\mathbf{u}^*$ is replaced with $\mathbf{v}$ for readability. The right-hand side $\mathbf{r}$ is equal to the previous right-hand side for Eq. (5.11) plus the explicit temporal terms computed in the viscosity step. The jump conditions $\mathbf{k}$ and $\mathbf{h}$ are equal to the jump conditions in Eqs. (5.12) and (5.13).

To solve Eqs. (5.63)-(5.65), we use a finite volume discretization, which enables us to handle points near the interface with ease and to ensure second order accuracy at the walls if a Neumann boundary condition is present on $\mathbf{u}$. We consider a control volume C_{ijk} (C_{ij} in 2D), a cube (square in 2D) centered at the node (i,j,k) ((i,j) in 2D) on the computational mesh. An example of a control volume in 2D with a uniform grid near an interface is shown in figure 5.8. We integrate Eq. (5.63) over the control volume and obtain

$$\int_{C_{ijk}}\eta\mathbf{v}\mathrm{d}V - \int_{C_{ijk}}\nabla\cdot\sigma\mathrm{d}V = \int_{C_{ijk}}\mathbf{r}\mathrm{d}V. \tag{5.68}$$

The Divergence Theorem is then applied to rewrite this equation as

$$\int_{C_{ijk}}\eta\mathbf{v}\mathrm{d}V - \int_{\partial C_{ijk}}\sigma\cdot\mathbf{n}\mathrm{d}S = \int_{C_{ijk}}\mathbf{r}\mathrm{d}V, \tag{5.69}$$

where ∂C_{ijk} represents the boundary of cell C_{ijk}. If the cell crosses the interface Γ, then this can further be rewritten by splitting the integrals into parts corresponding to the two fluid phases $\Omega^{\pm}$ as

$$\int_{C^+_{ijk}}\eta\mathbf{v}\mathrm{d}V + \int_{C^-_{ijk}}\eta\mathbf{v}\mathrm{d}V - \int_{\partial C^+_{ijk}}\sigma\cdot\mathbf{n}\mathrm{d}S - \int_{\partial C^-_{ijk}}\sigma\cdot\mathbf{n}\mathrm{d}S = \int_{C^+_{ijk}}\mathbf{r}\mathrm{d}V + \int_{C^-_{ijk}}\mathbf{r}\mathrm{d}V. \tag{5.70}$$

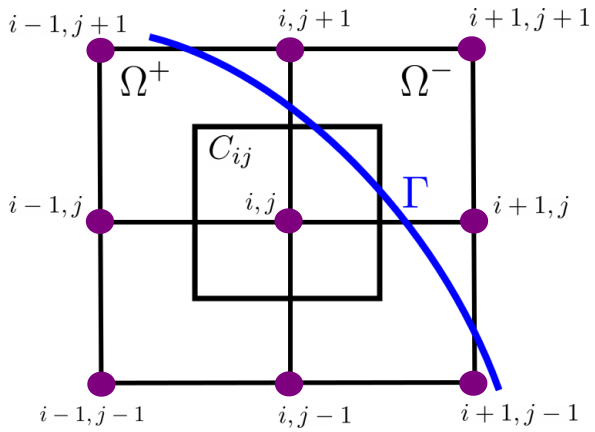

Figure 5.8: Control volume C_{ij} near the interface Γ in the two dimensional case.

Additionally, because the interface Γ crosses C_{ijk}, the surface integral terms are rewritten using the stress jump condition Eq. (5.65) as

$$-\int_{\partial C^+_{ijk}} \sigma \cdot \mathbf{n} \mathrm{d}S - \int_{\partial C^-_{ijk}} \sigma \cdot \mathbf{n} \mathrm{d}S = -\int_{\partial C^+_{ijk} \backslash \Gamma} \sigma \cdot \mathbf{n} \mathrm{d}S - \int_{\partial C^-_{ijk} \backslash \Gamma} \sigma \cdot \mathbf{n} \mathrm{d}S - \int_{\Gamma} \mathbf{h} \mathrm{d}S. \quad (5.71)$$

From this point, the derivatives present in Eq. (5.71) need to be approximated. The details of how this discretization is done will be fully explained in 2D, as the discretization in 3D is a relatively straightforward generalization.

5.8.2 Discretization in 2D

To discretize the integrals of Eq. (5.71), we begin by splitting these up along the four walls of the control volume C_{ij}: R, L, T, B (right, left, top, and bottom)

$$\int_{\partial C^{\pm}_{ij} \backslash \Gamma} = \int_{R^{\pm} \backslash \Gamma} + \int_{L^{\pm} \backslash \Gamma} + \int_{T^{\pm} \backslash \Gamma} + \int_{B^{\pm} \backslash \Gamma}. \quad (5.72)$$

The derivatives present in Eq. (5.71) are present in the stress tensor σ found in Eq. (5.66). To handle the derivatives present in the $\nabla \mathbf{v}$ term in σ, a standard, second order central difference formula is used. However, in the case of adaptive octree/quadtree grids, the direct neighbor to the center node may not exist in a particular direction. To handle this, we follow [47] and define ghost values of desired nodal quantities at T-junctions to circumvent the lack of direct neighbor. These ghost values are constructed using a third order accurate interpolation scheme. An example of this scenario is shown in figure 5.9. Here, node n_0

does not have a direct neighbor to the right, so we introduce a ghost node n_r on the face delimited by nodes n_{rt} and n_{rb}. For any nodal quantity ϕ, sampled at the existing nodes, we can calculate a third-order accurate ghost value ϕ_r using the information at n_0, at its direct neighbors in all three other directions (*i.e.* n_l, n_t, n_b), and at the neighboring nodes n_{rt} and n_{rb} as

$$\phi_r = \frac{r_b\phi_{r_t} + r_t\phi_{t_b}}{r_t + r_b} - \frac{r_t r_b}{t+b}\left(\frac{\phi_t - \phi_0}{t} - \frac{\phi_0 - \phi_b}{b}\right). \tag{5.73}$$

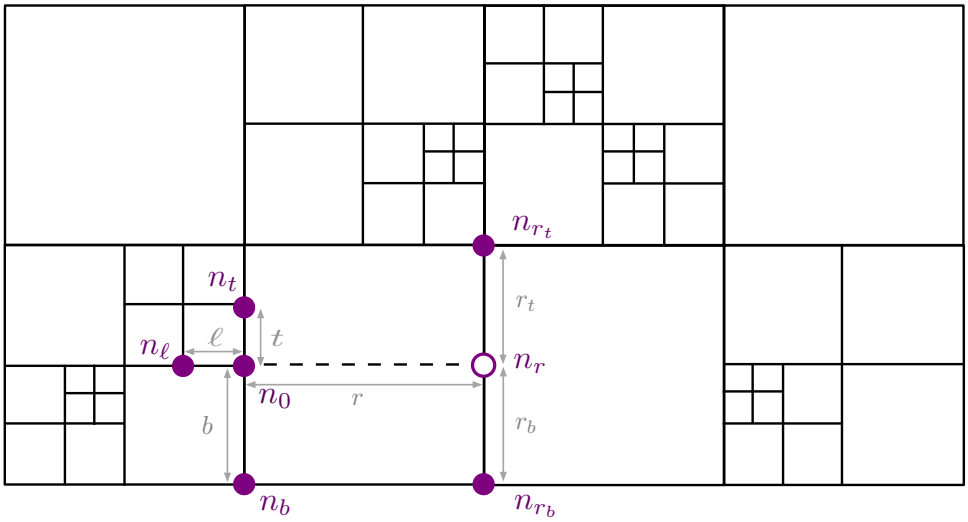

Figure 5.9: Finite difference discretization on quadtree grids. Here, node n_0 has no direct neighbor to the right, and thus a ghost node n_r (○) must be constructed using the existing neighboring nodes (●). Standard central discretizations can then be constructed using this ghosted neighborhood [14].

This approach requires solving a linear system for every component of the velocity field $\mathbf{v}$ at once because the $\nabla\mathbf{v}^{\mathrm{T}}$ term in the stress tensor σ removes the ability to decouple the velocity components. This results in several cross-coupling terms that require an additional approach to discretize. The full Helmholtz system Eq. (5.71) has two components in 2D, those being

$$-\int_{\partial C_{ij}^{+}\setminus\Gamma} \mu\left(2u_x N_1 + (v_x + u_y)N_2\right)\mathrm{d}S - \int_{\partial C_{ij}^{-}\setminus\Gamma} \mu\left(2u_x N_1 + (v_x + u_y)N_2\right)\mathrm{d}S - \int_{\Gamma} h_1 \mathrm{d}S, \tag{5.74}$$

and

$$-\int_{\partial C_{ij}^{+}\backslash\Gamma}\mu\left((u_y+v_x)N_1+2v_yN_2\right)\mathrm{d}S-\int_{\partial C_{ij}^{-}\backslash\Gamma}\mu\left((u_y+v_x)N_1+2v_yN_2\right)\mathrm{d}S-\int_{\Gamma}h_2\mathrm{d}S, \tag{5.75}$$

where $\mathbf{u}=(u,v)^{\mathrm{T}}$, $\mathbf{h}=(h_1,h_2)^{\mathrm{T}}$, and $\mathbf{n}=(N_1,N_2)^{\mathrm{T}}$. In Eq. (5.74), the v_x term here represents a coupling of the velocity component v with the velocity component u. The v_x terms here are non-zero at the top and bottom walls of the cell C_{ij}. In Eq. (5.75), the u_y term represents a coupling of the velocity component u with the velocity component v. The u_y terms here are non-zero on the left and right walls of the cell C_{ij}. These cross-coupling terms are more challenging to discretize because they are derivatives in the direction orthogonal to the normal of the face of the control volume that they are defined at.

The discretization of these cross-coupling terms is handled through averaging derivatives in neighboring quadrants. Each node n_{ij} has four neighboring quadrants. The v_x terms across the top and bottom walls of the control volume C_{ij} are discretized through averaging the v_x derivatives in either the upper or lower right and left neighboring quadrants depending on if the discretization is being performed on the top or bottom wall, respectively. These averages are also weighted by the area of their respective quadrant. For example, given the setup in figure 5.8, the discretization of v_x on the top wall of the control volume is

$$\begin{aligned}v_x|_T=\frac{W_L}{2}\left(\frac{v_{i-1,j+1}-v_{i,j+1}}{\Delta x_L}+\frac{v_{i-1,j}-v_{i,j}}{\Delta x_L}\right)\\+\frac{W_R}{2}\left(\frac{v_{i,j+1}-v_{i+1,j+1}}{\Delta x_R}+\frac{v_{i,j}-v_{i+1,j}}{\Delta x_R}\right)+\mathcal{O}\left(\Delta x^2\right),\end{aligned} \tag{5.76}$$

where Δx_L and Δx_R are the lengths of the left and right quadrants, respectively. The weights W_L and W_R are defined as

$$W_L=\frac{\text{Area}(C_R)}{\text{Area}(C_R)+\text{Area}(C_L)},\, W_R=\frac{\text{Area}(C_L)}{\text{Area}(C_R)+\text{Area}(C_L)}, \tag{5.77}$$

where C_L and C_R are the left and right quadrants, respectively. A similar procedure is done for discretizing the u_y terms on the left and right walls of the control volume. For example, given the setup in figure 5.8, the discretization of u_y on the right wall of the control volume is

$$\begin{aligned}u_y|_R=\frac{W_T}{2}\left(\frac{u_{i,j+1}-u_{i,j}}{\Delta y_T}+\frac{u_{i+1,j+1}-u_{i+1,j}}{\Delta y_T}\right)\\+\frac{W_B}{2}\left(\frac{u_{i,j}-u_{i,j-1}}{\Delta y_B}+\frac{u_{i+1,j}-u_{i+1,j-1}}{\Delta y_B}\right)+\mathcal{O}\left(\Delta y^2\right),\end{aligned} \tag{5.78}$$

where Δy_T and Δy_B are the lengths of the top and bottom quadrants, respectively. The weights W_T and W_B are defined as

$$W_T = \frac{\text{Area}(C_B)}{\text{Area}(C_T) + \text{Area}(C_B)}, W_B = \frac{\text{Area}(C_T)}{\text{Area}(C_T) + \text{Area}(C_B)}, \tag{5.79}$$

where C_T and C_B are the top and bottom quadrants, respectively. We note that the two quadrants used for the averaging need not be unique, as is the case for the node n_0 in figure 5.9, where the top and bottom neighboring quadrants of the right wall of the control volume are the same. The remaining terms in $\nabla \mathbf{v}^{\mathrm{T}}$ are handled through the same second order central difference scheme used for those in $\nabla \mathbf{v}$.

This coupled approach is much more easily achievable in this fully collocated numerical method because all of the velocity components are stored at the same location. In a MAC grid layout, this is challenging due to the different velocity components being stored at adjacent faces. By solving the viscosity step in this coupled approach, we are able to solve Eqs. (5.11)-(5.13) without needing an additional iterative correction on the stress jump condition (5.13), as was needed in [69].

5.8.3 Boundary conditions on the domain boundary

As previously stated, the boundary conditions for $\mathbf{u}^*$ are

$$\mathbf{B}(\mathbf{u}^*) = \mathbf{B}(\mathbf{u}^{n+1}) + \mathbf{c}. \tag{5.80}$$

The wall boundary condition $\mathbf{B}(\mathbf{u}^{n+1})$ can be either Dirichlet or Neumann which respectively can be expressed as

$$\mathbf{B}(\mathbf{u}^{n+1}) = \mathbf{u}|_{\partial\Omega}, \tag{5.81}$$

$$\mathbf{B}(\mathbf{u}^{n+1}) = (\nabla \mathbf{u} \cdot \mathbf{n})|_{\partial\Omega}. \tag{5.82}$$

In the case of a Dirichlet boundary condition, we also prescribe a Dirichlet boundary condition for $\mathbf{u}^*$ as

$$\mathbf{u}^*|_{\partial\Omega} = \mathbf{u}|_{\partial\Omega} + \mathbf{c}. \tag{5.83}$$

In the case of a Neumann boundary condition, we also prescribe a Neumann boundary condition for $\mathbf{u}^*$, that being

$$(\nabla \mathbf{u}^* \cdot \mathbf{n})|_{\partial\Omega} = (\nabla \mathbf{u} \cdot \mathbf{n})|_{\partial\Omega}. \tag{5.84}$$

5.8.4 Treatment of the velocity jump condition

While the flux jump $\mathbf{h}$ is treated naturally by the finite volume discretization as is seen in Eq. (5.71), the actual velocity jump $\mathbf{k}$ needs to be treated with a different approach. We handle the velocity jump by extending it to the entire domain. We introduce the extension $\widetilde{\mathbf{k}}$ of $\mathbf{k}$ to the entire domain such that $\widetilde{\mathbf{k}}$ satisfies

$$\eta\widetilde{\mathbf{k}} - \mu^{-}\nabla\cdot(\nabla\widetilde{\mathbf{k}}) = 0 \qquad \forall \mathbf{x}\in\Omega^{-}, \tag{5.85}$$

$$\widetilde{\mathbf{k}} = 0 \qquad \forall \mathbf{x}\in\Omega^{+}, \tag{5.86}$$

$$\widetilde{\mathbf{k}} = \mathbf{k} \qquad \forall \mathbf{x}\in\Gamma. \tag{5.87}$$

We introduce the function $\psi = \mathbf{v} - \widetilde{\mathbf{k}}$ which by Eqs. (5.63), (5.64) and (5.85) satisfies

$$\eta\psi - \nabla\cdot\widetilde{\sigma} = \widetilde{\mathbf{r}} \qquad \forall \mathbf{x}\in\Omega\setminus\Gamma, \tag{5.88}$$

$$[\![\psi]\!] = 0 \qquad \forall \mathbf{x}\in\Gamma, \tag{5.89}$$

$$[\![\widetilde{\sigma}\cdot\mathbf{n}]\!] = \mathbf{h} - \left[\!\left[\sigma_{\widetilde{\mathbf{k}}}\cdot\mathbf{n}\right]\!\right] \qquad \forall \mathbf{x}\in\Gamma, \tag{5.90}$$

where

$$\widetilde{\mathbf{r}} = \mathbf{r} - \mu^{-}\nabla\cdot((\nabla\widetilde{\mathbf{k}})^{\mathrm{T}}), \tag{5.91}$$

$$\widetilde{\sigma} = \mu\left(\nabla\psi + (\nabla\psi)^{\mathrm{T}}\right), \tag{5.92}$$

$$\sigma_{\widetilde{\mathbf{k}}} = \mu\left(\nabla\widetilde{\mathbf{k}} + (\nabla\widetilde{\mathbf{k}})^{\mathrm{T}}\right). \tag{5.93}$$

The jump extension $\widetilde{\mathbf{k}}$ is solved for using the supra-convergent Poisson equation solver [47] on Eqs. (5.85)-(5.87), which is second order accurate in both the solution and its gradient. Then, ψ is solved for using the method of section 5.8. However, given that the right hand side of $\widetilde{\mathbf{r}}$ in Eq. (5.91) contains two derivatives of $\widetilde{\mathbf{k}}$, the solution ψ to Eq. (5.88) will in general be first order accurate. In practice, for small velocity jumps $\mathbf{k}$, this does not significantly impact the overall order of convergence for the coupled jump solver.

5.8.5 Coupled jump solver convergence results

The convergence of the coupled jump solver in 3D with test solution

$$u^{-} = \cos(x)\sin(z),\ u^{+} = \cos(x)\sin(z) + z^{2}, \tag{5.94}$$

$$v^{-} = \sin(y)\cos(x),\ v^{+} = \sin(y)\cos(x) + xy, \tag{5.95}$$

$$w^{-} = \sin(xy),\ w^{+} = \sin(xy) + y\sin(x), \tag{5.96}$$

and viscosities of $\mu^- = 1$ and $\mu^+ = 0.5$, test domain of $[-1, 1]^3$, and interface represented by the level set function $\phi = x^2 + y^2 + z^2 - 0.52^2$ is shown in table 5.1. We observe convergence between first and second order, as expected given how the solution jump condition is treated.

Level (max:min)	u L^∞ error	Order	v L^∞ error	Order	w L^∞ error	Order
3:1	0.0411	-	0.0521	-	0.0383	-
4:2	0.0116	1.82	0.0161	1.69	0.0128	1.58
5:3	0.0117	-0.0118	0.0101	0.680	0.0103	0.313
6:4	4.11e-3	1.51	3.80e-3	1.41	4.56e-3	1.17
7:5	1.44e-3	1.51	1.40e-3	1.44	1.91e-3	1.25

Table 5.1: Convergence of coupled jump solver in 3D.

5.9 Projection and pressure guess steps

In this section, we discuss the procedures by which the pressure guess, step 1 in figure 5.2, and projection, step 2b in figure 5.2, steps are performed. Both steps require solving a Poisson problem with jump conditions and both are treated using the same Poisson jump solver that uses a finite volume discretization, similar to that of the coupled jump solver of section 5.8.1. Additionally, we demonstrate numerical evidence of the stability of the nodal projection operator for two-phase flow.

We use the particular pressure guess shown here as this was shown in [69] to enable the use of the less restrictive time step constraint given in Eq. (5.62). This pressure guess resolves capillary effects before the viscosity step, something that is not applicable in our single-phase flow method 4.4.

5.9.1 Single value finite volume Poisson jump solver

The Poisson problems Eq. (5.8) and Eq. (5.15) are solved using a single valued finite volume discretization of the Poisson equation with jump conditions. This discretization is identical to how the non cross-coupling terms were handled in the coupled jump solver presented in section 5.8.1, including how the control volumes are constructed, how the discretization matrix is built, and how the jump conditions are handled.

We test the convergence of this solver by solving the following Poisson problem for $u(x, y, z)$ with jump conditions

$$\eta u - \mu \boldsymbol{\Delta} u = r \qquad \forall \mathbf{x} \in \Omega \setminus \Gamma, \tag{5.97}$$

$$[\![u]\!] = k \qquad \forall \mathbf{x} \in \Gamma, \tag{5.98}$$

$$[\![\nabla u \cdot \mathbf{n}]\!] = h \qquad \forall \mathbf{x} \in \Gamma. \tag{5.99}$$

We select the test solution

$$u^- = xy + xy^2, u^+ = \cos(z)\sin(y), \tag{5.100}$$

with $\eta = 0$, viscosities of $\mu^- = 10$ and $\mu^+ = 1$, in a test domain of $[-2, 2]^3$, and with the interface represented by the level set function $\phi = x^2 + y^2 + z^2 - 0.5889^2$. The right-hand side r is defined as the exact right-hand sided needed for the test solution to satisfy the Poisson equation and the jump conditions k and h are defined as the exact jump conditions for the given test solution Eq. (5.100). The convergence of the L^∞ error for this test example is shown in table 5.2. We observe second order convergence for this example.

Level (max:min)	$\|\|\mathbf{u}\|\|_\infty$ error	Order
3:1	0.0239	-
4:2	0.1762	-2.88
5:3	0.0810	1.12
6:4	0.0174	2.22
7:5	3.69e-3	2.23

Table 5.2: Convergence of single value finite volume Poisson jump solver in 3D.

5.9.2 Pressure guess

In the pressure guess step, a pressure guess $\widetilde{p}$ is solved for as the pressure induced by the jump in interfacial forces across the interface

$$\boldsymbol{\Delta}\widetilde{p} = 0 \qquad \forall \mathbf{x} \in \Omega \setminus \Gamma \tag{5.101a}$$

$$[\![\widetilde{p}]\!] = -\gamma\kappa - \mathbf{q}\cdot\mathbf{n} \qquad \forall \mathbf{x} \in \Gamma, \tag{5.101b}$$

$$\left[\!\left[\frac{1}{\rho}\mathbf{n}\cdot\nabla\widetilde{p}\right]\!\right] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{5.101c}$$

$$\mathbf{B}(\widetilde{p}) = 0 \qquad \forall \mathbf{x} \in \partial\Omega. \tag{5.101d}$$

$$\tag{5.101e}$$

The wall boundary condition $\mathbf{B}(\widetilde{p})$ are homogeneous and can be either Dirichlet or Neumann which respectively can be expressed for $\mathbf{x} \in \partial\Omega$ as

$$\widetilde{p} = 0, \tag{5.102}$$

$$(\nabla\widetilde{p}\cdot\mathbf{n})|_{\partial\Omega} = 0. \tag{5.103}$$

We solve for the pressure guess $\widetilde{p}$ using the solver described in section 5.9.1.

5.9.3 Projection

In the projection step, the Hodge variable Φ is solved for as the solution to the following homogeneous, jump Poisson boundary value problem

$$\Delta\Phi = \nabla\cdot\mathbf{u}^* \qquad \forall \mathbf{x} \in \Omega \setminus \Gamma \tag{5.104a}$$

$$[\![\rho\Phi]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{5.104b}$$

$$[\![\nabla\Phi\cdot\mathbf{n}]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{5.104c}$$

$$\mathbf{B}(\Phi) = \mathbf{B}(p) \qquad \forall \mathbf{x} \in \partial\Omega. \tag{5.104d}$$

$$\tag{5.104e}$$

The wall boundary condition $\mathbf{B}(\Phi)$ can be either Dirichlet or Neumann which respectively can be expressed as

$$\mathbf{B}(p) = p|_{\partial\Omega}, \tag{5.105}$$

$$\mathbf{B}(p) = (\nabla p\cdot\mathbf{n})|_{\partial\Omega}. \tag{5.106}$$

We arrive at having homogeneous jump conditions for Φ because of our choice of pressure guess $\widetilde{p}$. The wall boundary conditions are taken from the pressure wall boundary

conditions. We note that when solving for the Hodge variable Φ, we choose to rescale the Hodge variable Φ by the density ρ to remove ρ from the jump condition in Eq. (5.104). We define

$$\Phi = \frac{1}{\rho}\varphi, \tag{5.107}$$

and arrive at the following Poisson jump problem

$$\frac{1}{\rho}\Delta\varphi = \nabla \cdot \mathbf{u}^* \qquad \forall \mathbf{x} \in \Omega \setminus \Gamma \tag{5.108a}$$

$$[\![\varphi]\!] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{5.108b}$$

$$\left[\!\left[\frac{1}{\rho}\nabla\varphi \cdot \mathbf{n}\right]\!\right] = 0 \qquad \forall \mathbf{x} \in \Gamma, \tag{5.108c}$$

Once φ is solved for, the rescaling is undone and Φ is used to project $\mathbf{u}^*$ into the divergence free space. This Poisson boundary value problem is also solved using the nodal single value finite volume Poisson jump solver from section 5.9.1, which while not fully second order in both Φ and $\nabla\Phi$ due to the treatment of the jump condition, still results in second order convergence of the overall method as will be demonstrated in chapter 6. This was not achieved in the existing MAC grid solver [69], demonstrating one way in which this nodal solver achieves higher accuracy in the velocity field.

Once Φ is obtained as the solution to Eq. (5.104), then $\mathbf{u}^{n+1}$ is constructed as

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \nabla\Phi. \tag{5.109}$$

As was established in our single phase work [14], this projection operation, entirely collocated at the nodes, is not a true projection, and so we must iterate on this process of successively projecting the velocity $\mathbf{u}^*$ produced from the viscosity step. We use an iterative procedure to remove compressible modes through repeated projection, performing successive projections until

$$\left\|\mathbf{u}^{n+1} - \mathcal{P}_N\mathbf{u}^{n+1}\right\|_\infty < \epsilon_i \left\|\mathbf{u}^{n+1}\right\|_\infty, \tag{5.110}$$

or a predefined maximum number of iterations, K_{max}, has been reached. Typically, we choose $\epsilon_i = 10^{-3}$, set $K_{max} = 3$, and only a few $(1-3)$ iterations are required to reach convergence. This process is detailed in figure 5.2 in Eqs. (5.15)-(5.18).

Stability of the Projection Step

As was established in our single phase work [14], this collocated projection operator is shown to be stable under iteration for a range of boundary conditions. Here, we demonstrate

numerically in 2D that this stability property still holds in a two-phase context when jump conditions are present.

Using an initially divergence free velocity field, we successively apply the projection operator to this field and monitor the norm of the variation between the velocity at the current projection iteration and the previous projection iteration. For a stable operator, this variation will tend to zero, up to the solver tolerance. For this investigation, the initial velocity field chosen is

$$u^{\pm} = \sin(x)\cos(y), \tag{5.111}$$
$$v^{\pm} = -\cos(x)\sin(y), \tag{5.112}$$

with the interface Γ defined as the zero contour of the level set function

$$\phi(x, y) = 0.1 - \sin(x)\sin(y). \tag{5.113}$$

Additionally, we consider densities of $\rho^{\pm} = 1$ and domain $\Omega = \left[-\frac{\pi}{3}, \frac{4\pi}{3}\right]^2$. We choose a Quadtree grid with a minimum level of 4, a maximum level of 8, and choose to refine the grid near the interface. A picture of this domain is shown in figure 5.10. For the boundary

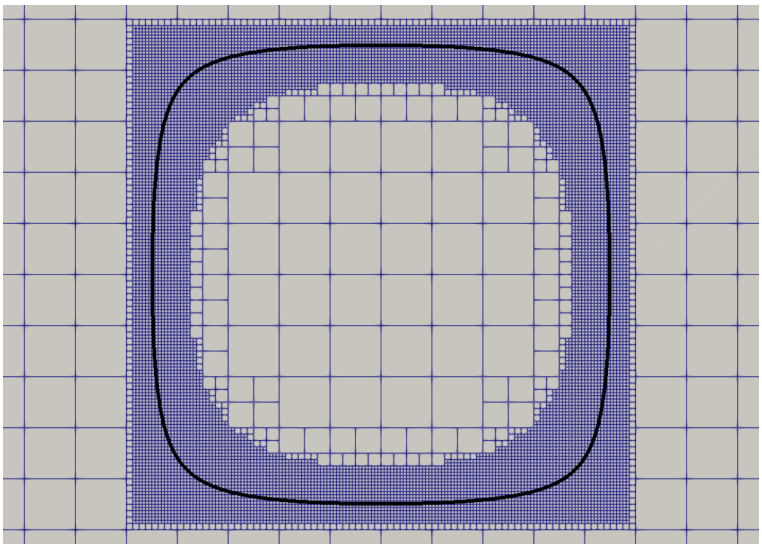

Figure 5.10: Representation of the Quadtree grid of Ω and the interface Γ (shown in black), for the example given in Eqs. (5.111) and (5.112) to show the stability of the collocated projection operator.

condition on the Hodge variable, we consider homogeneous Dirichlet, Neumann, and a

combination of Dirichlet and Neumann boundary conditions to demonstrate the robustness of the collocated projection operation.

The results of this test are shown in Figure 5.11. As expected, the variation in the norm of the velocity between successive projections decreases to machine precision as we successively apply our projection operator. For these examples, the tolerance of our linear solver was set to 10^{-12}. As expected, we see that our collocated projection operator is numerically stable for all boundary conditions tested. We also note that in practice, only a small number of iterations $(1-3)$ of the projection operator will be used and, as we show in chapter 6, second order accuracy in the velocity can be achieved with only a single projection applied.

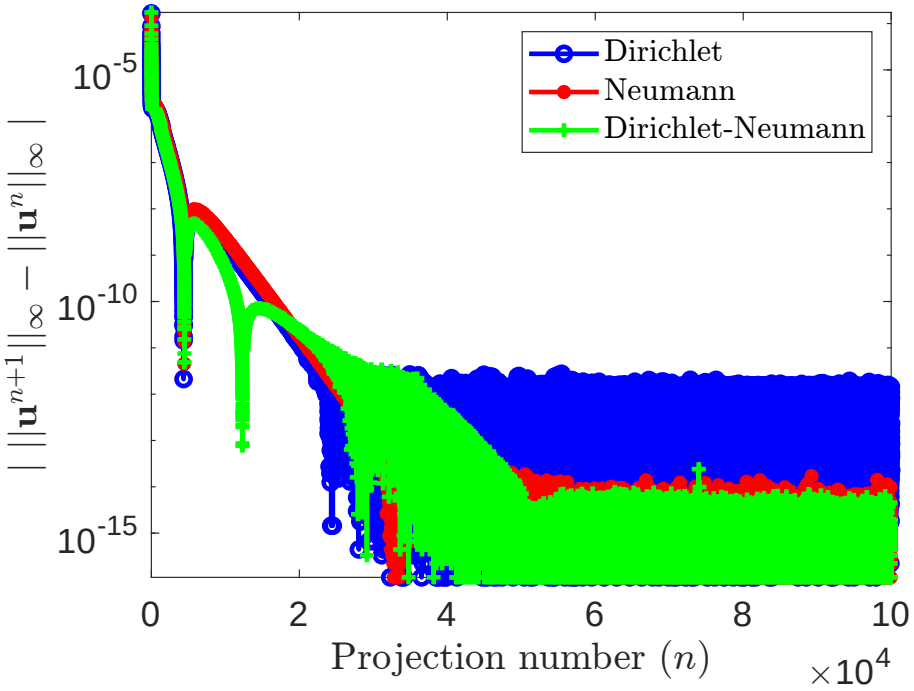

Figure 5.11: Difference between successive projections of an incompressible field, with different boundary conditions on the Hodge variable.

5.9.4 Pressure reconstruction

In our formulation of the projection method, pressure is never directly computed. If desired, the pressure p in the numerical method can be reconstructed using the pressure guess $\widetilde{p}$, Hodge variable Φ, and $\mathbf{u}^*$, and follows from the choice of temporal discretization.

Following [69], which has an identical procedure on a MAC grid layout, the pressure reconstruction formula is

$$p = \widetilde{p} + \frac{\alpha\rho}{\Delta t_n}\Phi - \mu\nabla\cdot\mathbf{u}^*. \tag{5.114}$$

5.10 Conclusions

We presented a novel projection method for the simulation of incompressible two-phase flow in arbitrary domains using quadtrees/octrees, where all of the variables are collocated at the grid nodes. This method extends our projection method for incompressible single-phase flow [14] and uses the same collocated projection operator. This projection operator is an approximate projection that converges to the canonical projection on the incompressible space when iterated.

Our numerical method discretizes the momentum equation Eq. (5.1) by treating the advection explicitly and the diffusive terms implicitly. The temporal discretization is a second order Semi-Lagrangian Backward Difference Formula scheme that is augmented with a local temporal limiter to ensure numerical stability. The spatial discretization of the momentum equation is handled with a finite volume scheme that easily treats the full stress tensor with jump conditions, and a similar single valued finite volume scheme is used to discretize the projection and pressure guess steps. We verified the convergence of these finite volume schemes and additionally numerically verified the stability of our collocated projection operator in the context of two-phase flow across a range of boundary conditions. In chapter 6, we demonstrate that our solver converges with second order accuracy and validate its capabilities to simulate applications of multi-phase flow through several canonical validation examples.

The collocated nodal framework employed by our solver dramatically improves on its ability to be applied to studying scientific and engineering applications of multi-phase flows. The collocation of variables simplifies the implementation cost by limiting the number of data structures needed to store the computational variables, reducing the number of complex interpolation operations needed in a staggered grid framework, and allowing for an easier treatment of the stress tensor through the finite volume coupled jump solver presented in section 5.8. As such, we believe that our solver serves as the ideal tool for scientists and engineers wishing to develop simulations of multi-phase flow applications, as well as the best tool for us to simulate the rising oil droplets experiment presented in chapters 2 and 3, due to the combination of its accuracy and its ease of implementation.

Chapter 6

Numerical method verification and validation

In this chapter, we validate our nodal incompressible two-phase Navier-Stokes solver presented in chapter 5 using several canonical problems in two and three spatial dimensions. We begin by considering an analytical solution to verify its convergence, then demonstrate its accuracy with several canonical two-phase flow test cases. We conclude this chapter by demonstrating the versatility and capabilities of our solver to accurately simulate numerous rising bubbles and interactions of bubbles with solid flow obstructions.

6.1 Analytic vortex

We verify the convergence of our method by solving Eqs. (5.1)-(5.4) for the following exact analytical solution

$$u(x, y, t) = \sin(x)\cos(y)\cos(t), \tag{6.1a}$$
$$v(x, y, t) = -\cos(x)\sin(y)\cos(t), \tag{6.1b}$$
$$p = 0, \tag{6.1c}$$
$$\phi(x, y) = 0.1 - \sin(x)\sin(y), \tag{6.1d}$$

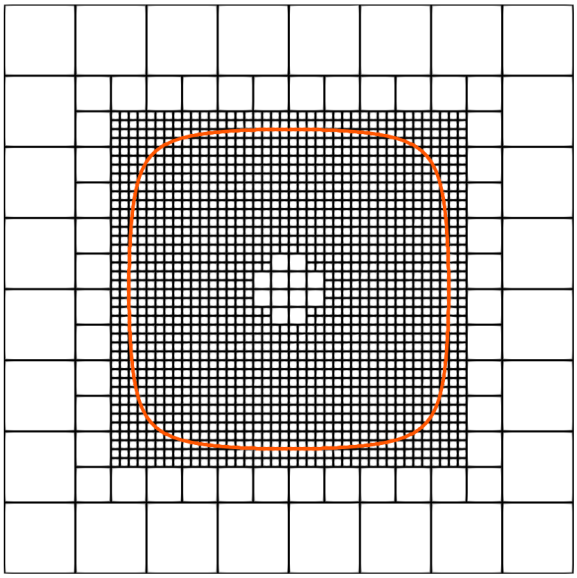

Figure 6.1: Representation of the quadtree discretization of Ω and the interface Γ (shown in orange).

for the two-phase incompressible Navier-Stokes equations with an external body force $\mathbf{f}^{\pm}(x, y, t)$ and interfacial stress $\mathbf{q}(x, y, t)$ given by:

$$\mathbf{f}^{\pm}(x, y, t) = \rho^{\pm}\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) - \mu^{\pm}\Delta \mathbf{u},$$
$$\mathbf{q}(x, y, t) = -\gamma \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} + \left[\!\left[\mu \nabla \mathbf{u} \cdot \frac{\nabla \phi}{|\nabla \phi|}\right]\!\right].$$

This solution is constructed from an analytic solution of the single-phase case. Additionally, we set the following parameters as:

$$\mu^{+}, \rho^{+} = 1, \qquad \mu^{-}, \rho^{-} = 10, \qquad \gamma = 0.1, \qquad \Omega = \left[-\frac{\pi}{3}, \frac{4\pi}{3}\right]^2, \qquad t_{\text{final}} = \pi.$$

No-slip boundary conditions are imposed on the walls of the computational domain and the mesh is represented using a quadtree data structure (see Figure 6.1). We note that the mesh is only refined near the interface location (*i.e.* no vorticity based refinement). We perform the computations on a coarse mesh and then recursively produce a mesh where the min and max level of the quadtree is increased by one. We see in table 6.1 that if we use SLBDF we get second order convergence in the L^{∞}-norm for the velocity. For quadtree

levels up to 7:3, the solution seems to be underresolved and our order of accuracy is far less than desired. For quadtree levels of 7:3 to 9:5, we do see convergence. With the local temporal limiter, convergence drops to first order. In both cases, only a single projection was used, demonstrating that we can obtain second order convergence even with only a single projection, consistent with what we observed in the nodal projection method for single phase flows [14].

	Without temporal limiter		With temporal limiter	
Level (max:min)	$\mathbf{u}$ L^∞ error	Order	$\mathbf{u}$ L^∞ error	Order
4:0	2.71e-01	-	3.25e-01	-
5:1	7.38e-02	1.88	8.48e-02	1.94
6:2	4.57e-02	0.69	4.67e-02	0.86
7:3	5.01e-02	-0.13	4.38e-02	0.09
8:4	1.18e-02	2.08	1.73e-02	1.34
9:5	2.78e-03	2.09	6.23e-03	1.47

Table 6.1: Convergence of analytic vortex at the final time with and without the local temporal limiter.

6.2 Parasitic currents

In this second example, we consider the canonical example of parasitic or spurious currents [25, 62, 52]. Parasitic currents are purely artificial flows generated by capillary effects at the interface, as seen in the stress jump condition in Eq. (5.4). These manifest because our representation of the curvature is inherently inaccurate due its computation being through derivatives of the level-set function, Eq. (5.27). This inaccuracy generates a flow at the interface that can grow and lead to inaccurate velocity profiles near the interface. Since we are interested in studying applications where the most significant dynamics happen near the interface, we wish for the parasitic currents that arise in our numerical method to be bounded in time and ideally converging as the spatial resolution increases.

To study the magnitude and convergence of parasitic currents, we solve Eqs. (5.1)-(5.4) with $p = 0$, $\mathbf{f} = \mathbf{0}$, and $\mathbf{q} = \mathbf{0}$, with the interface given by the zero contour of the level-set function

$$\phi(x, y) = \sqrt{x^2 + y^2} - 0.003, \tag{6.2}$$

and parameters

$$\rho^+ = 1261, \rho^- = 1 \quad \mu^+ = 1.4746, \mu^- = 1, \quad \gamma = 0.05, \quad \Omega = [-0.005, 0.005]^2. \quad (6.3)$$

We enforce homogeneous Dirichlet boundary conditions for the velocity on the left and right computational walls, homogeneous Neumann boundary conditions for the velocity on the top and bottom walls, and the alternated boundary conditions for the pressure.

For the grid, we consider increasingly refined uniform grids of levels varying from 4 to 8, similar to that in [21]. The resulting L_∞-error for interface location and L_1-error for the velocity as functions of time are shown in figure 6.2. We observe second order convergence in the interface position, which is reasonable given that the error in the interface is largely driven by the advection of the reference map, which is done using a second order accurate method. The velocity also converges with values comparable to other studies [69, 25, 62, 21, 52]. Moreover, the capillary velocity $\mathbf{u}_{\text{cap}}$, which represents the relevant velocity scale for capillary effects, scales as

$$\mathbf{u}_{\text{cap}} \sim \sqrt{\frac{\gamma L}{\rho}}, \quad (6.4)$$

where L is the length of the computational domain. Here using $L = 0.01$ and ρ^+ for ρ, we find

$$\mathbf{u}_{\text{cap}} \sim \mathcal{O}(10^{-4}), \quad (6.5)$$

demonstrating that as we refine the spatial grid, the magnitude of the velocity induced by the parasitic currents converges to several orders of magnitude smaller than the scale of velocity relevant to any application featuring interface driven flow, further confirming that our method is not impacted by the presence of parasitic currents.

	Interface location		Velocity	
Level (max:min)	L^∞ error	Order	L^1 error	Order
4:4	1.47e-05	-	3.86e-07	-
5:5	2.96e-06	2.32	5.93e-08	2.70
6:6	6.92e-07	2.10	7.21e-08	-0.28
7:7	1.73e-07	1.99	1.74e-08	2.05
8:8	5.10e-08	1.76	1.06e-08	0.72

Table 6.2: Convergence of parasitic currents at the final time. The interface location error is computed within a band of Δx close to the interface.

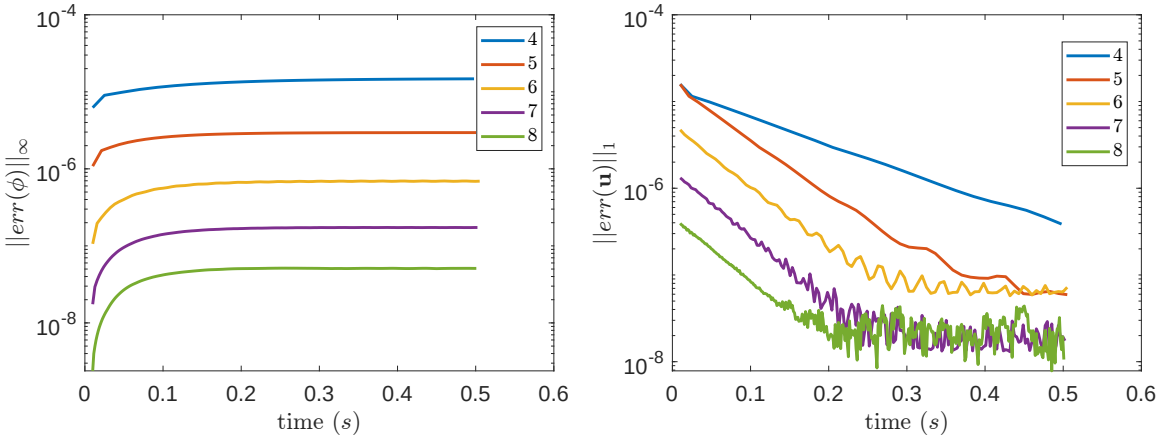

Figure 6.2: Convergence results for the parasitic currents example: interface location error (left) and velocity L_1 error (right) for increasing level of refinement.

6.3 Oscillating bubble

Next, we consider the oscillations of an initially spherical bubble around its equilibrium shape when a perturbation is introduced in its radius. This problem was originally theoretically studied by Lamb [36], where a perturbation of the radius of an initially spherical bubble generates capillary forces that produce oscillations which are eventually dampened by viscous effects. The initially circular bubble is defined in 2D by the level set function

$$\phi(r,\theta) = r - \left(R + \frac{\epsilon}{2}(3\cos^3\theta - 1)\right), \tag{6.6}$$

where r is the radial coordinate, θ is the polar angle, R is the initial bubble radius, and $\epsilon \ll 1$ is the perturbation to the radius. We set $\mathbf{f} = \mathbf{0}$, $\mathbf{q} = \mathbf{0}$, and use the parameters

$$\mu^+ = 0.02, \rho^+ = 1, \quad \frac{\mu^+}{\mu^-} = \frac{\rho^+}{\rho^-} = 10^3, \quad \gamma = 0.5, \quad R = 1, \quad \epsilon = 0.01, \quad \Omega = [-1.5, 1.5]^2. \tag{6.7}$$

We enforce homogeneous Neumann boundary conditions for the velocity on the left and right computational walls, homogeneous Dirichlet boundary conditions for the velocity on the top and bottom walls, and the alternated boundary conditions for the pressure.

A minimum level of 4 and an increasing maximum level of from 6 to 9 were used for the spatial resolution. Shown in figure 6.3 and table 6.3 is the convergence of the x radius of the bubble vs time. As the maximum level increases, we see that the period of oscillation converges to the predicted exponential decay in 3D of 3.629. In figure 6.4, the

velocity profile for the highest resolution case is shown at a time of expansion and a time of contraction.

Max level	Period of oscillation
6	3.834
7	3.774
8	3.706
9	3.693

Table 6.3: Convergence of period of oscillation. The 3D theoretical prediction is 3.629.

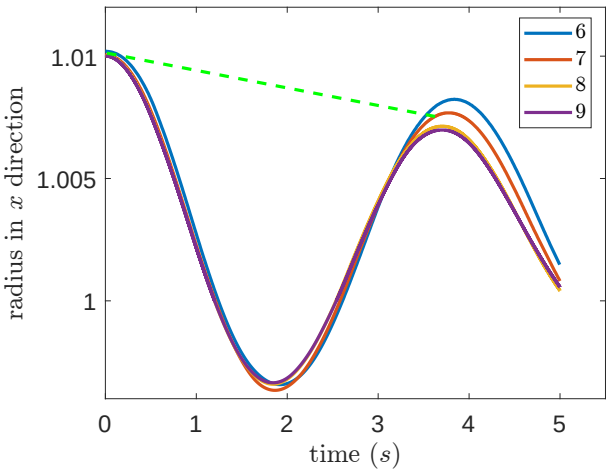

Figure 6.3: Oscillating bubble radius in x direction vs time with increasing max level. The green line represents the predicted exponential decay of the bubble in 3D.

6.4 Dynamics and deformations of rising bubbles

The next validation example we consider is the dynamics and deformations of rising bubbles subject to strong density and surface tension driven deformations. These examples are inspired by and validated against the experimental Cases a-h of [12]. We consider an initially spherical bubble suspended in another denser and more viscous fluid, where the density difference between the two fluids induces a buoyancy force causing the bubble to rise and deform.

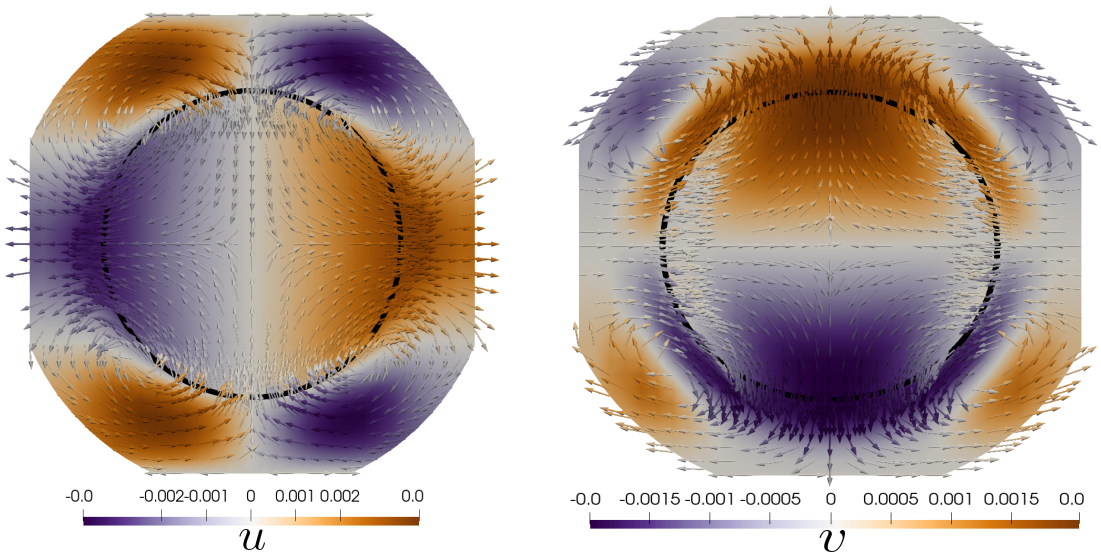

Figure 6.4: Oscillating bubble velocity profile at times of expansion and contraction. The colormap shows the horizontal velocity u at time of expansion (left) and vertical velocity v at time of contraction (right). The bubble interface is represented in black.

These dynamics are described by three non-dimensional numbers: the Morton, Eotvos and Reynolds numbers, respectively defined here as

$$Mo = \frac{g(\mu^-)^4}{\rho^-\gamma^3}, Eo = \frac{gd^2\rho^-}{\gamma}, Re = \frac{\rho^- Ud}{\mu^-}, \tag{6.8}$$

where U is the asymptotic rising velocity measured at the tip of the drop, g is the acceleration due to gravity, and d is the initial diameter of the undeformed bubble. The simulation parameters are constructed from these three numbers. We set the rising velocity U and undeformed diameter d to 1 and set the remaining parameters as

$$\rho^- = 1, \frac{\rho^-}{\rho^+} = 10^3, \mu^- = \frac{\rho^-}{Re}, \frac{\mu^-}{\mu^+} = 10^2, \gamma = \frac{(\mu^-)^2}{\rho^-}\sqrt{\frac{Eo}{Mo}}, g = \frac{(Mo)\rho^-\gamma^3}{(\mu^-)^4}. \tag{6.9}$$

The density and viscosity ratios are chosen to be close to those for air and water. By setting $U = 1$, we expect the asymptotic rising velocity of the rising bubbles to be close to 1. We perform these simulations in a domain of $[-16, 16]^2$ or $[-16, 16]^3$, much larger than the initial diameter of 1, and initialize the bubble far from the walls to minimize boundary effects. No-slip boundary conditions are imposed for the velocity on all walls except for the top wall, where a no-flux boundary condition is imposed, and the pressure

boundary conditions are imposed as the inverse of these. All of these simulations were performed with a minimum grid level of 4, giving a relatively coarse grid away from the bubble interface. We use the interface based refinement criterion Eq. (5.45) and a vorticity based refinement criterion Eq. (5.46) with the vorticity tolerance set to $T_V = 0.01$ and $\max_V = \max_{\text{level}} - 1$.

6.4.1 Single rising bubble in 2D

To begin, we consider case (d) from [12], where the non-dimensional parameters are

$$Mo = 266,\ Eo = 243,\ Re = 7.77. \tag{6.10}$$

We perform this investigation in a 2D domain of $[-16, 16]^2$ and consider the convergence of the bubble's asymptotic rising velocity and relative mass loss over time with increasing maximum grid level from 10 to 12. Figure 6.5 shows convergence in the rising velocity to an asymptotic value around 1, as predicted by the experiment. In figure 6.5, we additionally see convergence in relative mass loss as the refinement of the grid increases. Figure 6.6 shows the velocity magnitude at the final simulation time for each of these simulations, showing convergence in the interface shape as well. Though these are 2D simulations, the interface shape does qualitatively resemble the experiment conducted in this parameter regime.

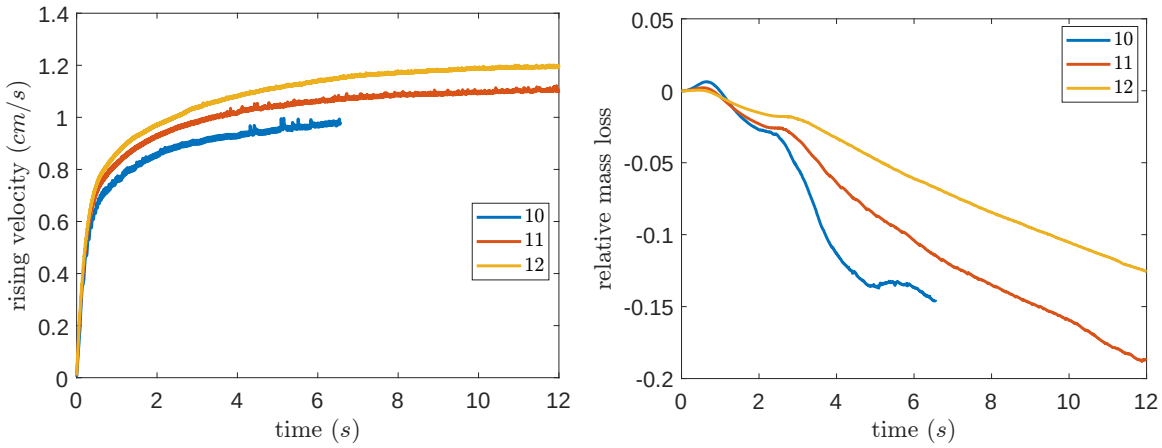

Figure 6.5: Rising velocity of 2D Bhaga Weber case (d) [12], with increasing max level and min level of 4 (left) and relative mass loss of 2D Bhaga Weber case (d) [12], with increasing max level and min level of 4 (right). Negative relative mass loss indicates a gain in mass.

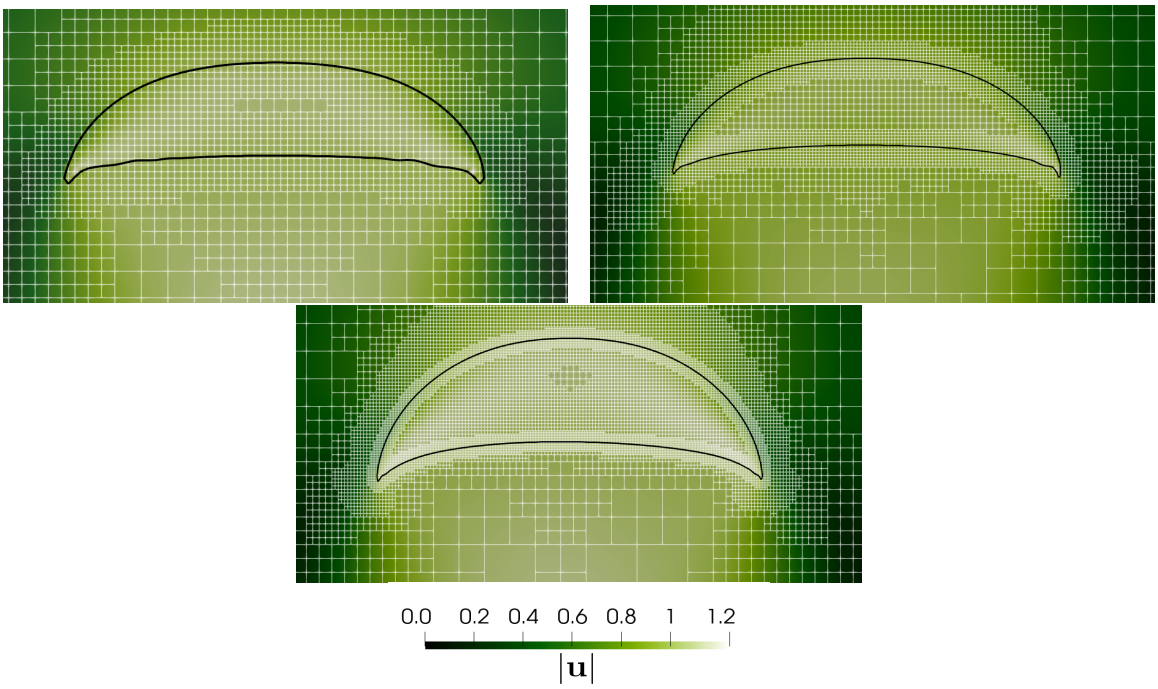

Figure 6.6: Velocity magnitude $|\mathbf{u}|$ and interface shape at final time of simulation for 2D rising bubble, Bhaga Weber case (d) [12] with increasing max level of refinement. Top left has a maximum level of 10, top right has a maximum level of 11, and bottom has a maximum level of 12. A subset around the bubble of the adaptive grid is also shown for each case. The initial bubble has a diameter of 1 cm.

6.4.2 Rising bubbles in 3D

Single bubble

To perform a more rigorous validation with the experiments of [12], we turn to 3D simulations of the 8 parameter regimes presented there. These 8 experimental cases are defined in terms of the dimensionless parameters: Eötvös number (Eo), Morton number (Mo), and Reynolds number (Re), defined as

$$Eo = \frac{gd^2\rho^-}{\gamma},\ Mo = \frac{g(\mu^-)^4}{\rho^-\gamma^3},\ Re = \frac{\rho^- dU}{\mu^-}, \tag{6.11}$$

where g is the acceleration due to gravity, d is the initial diameter of the bubble, and U is a characteristic velocity. These parameters are defined as follows for the 8 cases of Bhaga and Weber [12]. The Eötvös number represents the ratio of gravitational forces to capillary

Case:	a	b	c	d	e	f	g	h
Eo	8.67	17.7	32.2	243	115	237	339	641
Mo	711	711	8.2×10^{-4}	266	4.63×10^{-3}	8.2×10^{-4}	43.1	43.1
Re	0.078	0.232	55.3	7.77	94	259	18.3	30.3

Table 6.4: Rising bubbles in 3D: dimensionless parameters associated with the Bhaga Weber cases (a) - (h).

forces, the Morton number is used with the Eötvös number to characterize the shape of the bubbles, and the Reynolds number represents the ratio of inertial effects to viscous effects.

Figures 6.7 and 6.8 show the final interface shape of the bubbles along with streamlines of the magnitude of apparent velocity $|\mathbf{u}_a|$ (velocity in the reference frame of the bubble) for each experimental case. We observe qualitative agreement with the experiments over a range of parameter regimes, though we note that the simulated bubbles feature some roughness along their edges as a result of the limited spatial resolution. While we are unable to fully resolve the thin film skirts present in cases (e), (g), and (h) in the asymptotic regime, we still observe the formation of these thin regions as the bubbles rise before the thin regions disappear due to limitations in resolution. Figures 6.9 and 6.10 demonstrate the time evolution for cases (e) and (h), respectively, showing our ability to resolve the multiple configurations these bubbles take on over time. These results are qualitatively consistent with those presented in other numerical studies [69, 21].

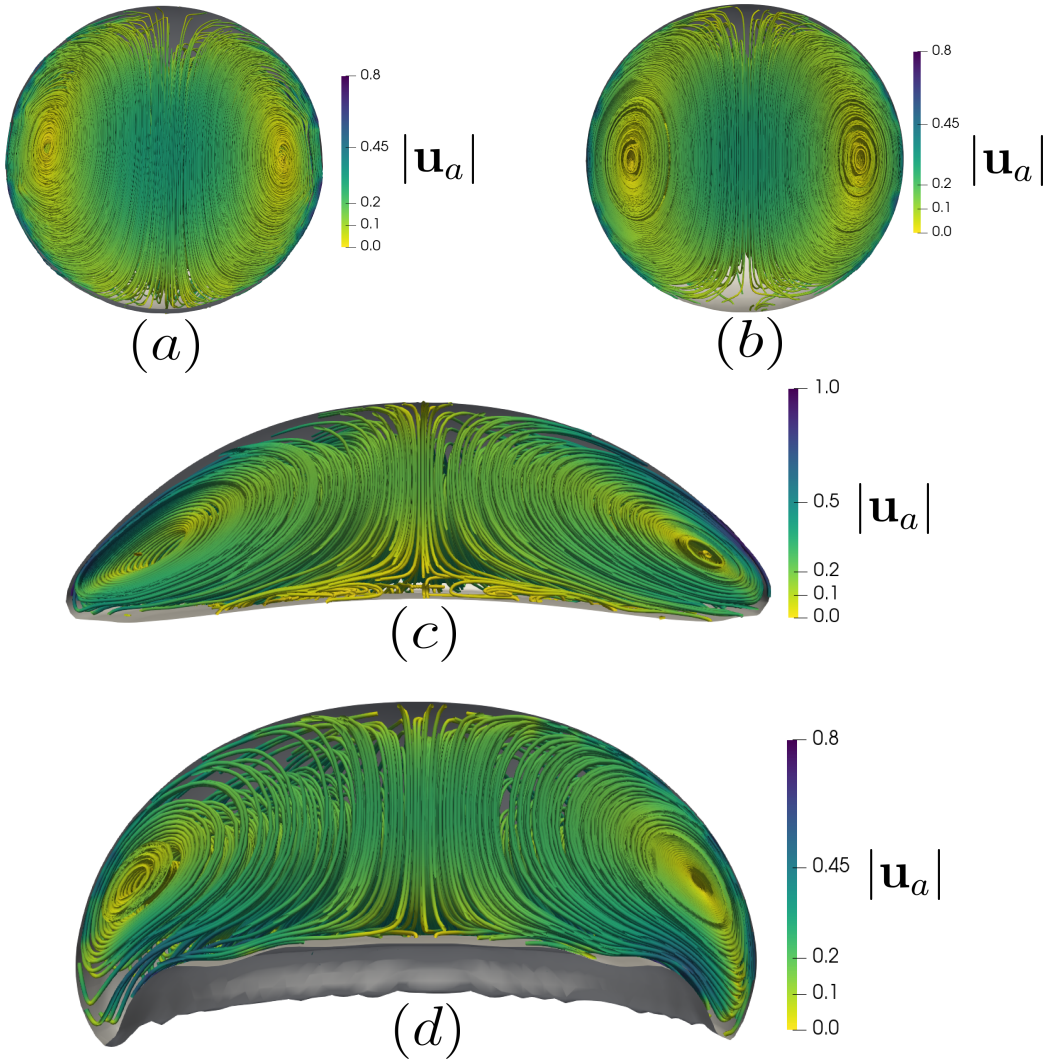

Figure 6.7: Final bubble shapes and interior flow fields corresponding to examples 2(a–d) from Bhaga and Weber [12]. The streamlines are colored according to the magnitude of the apparent velocity $|\mathbf{u}_a|$, or velocity of the fluid in the reference frame of the rising bubble. Each bubble was initialized with a diameter of 1 cm. For clarity, each bubble is displayed sliced in half.

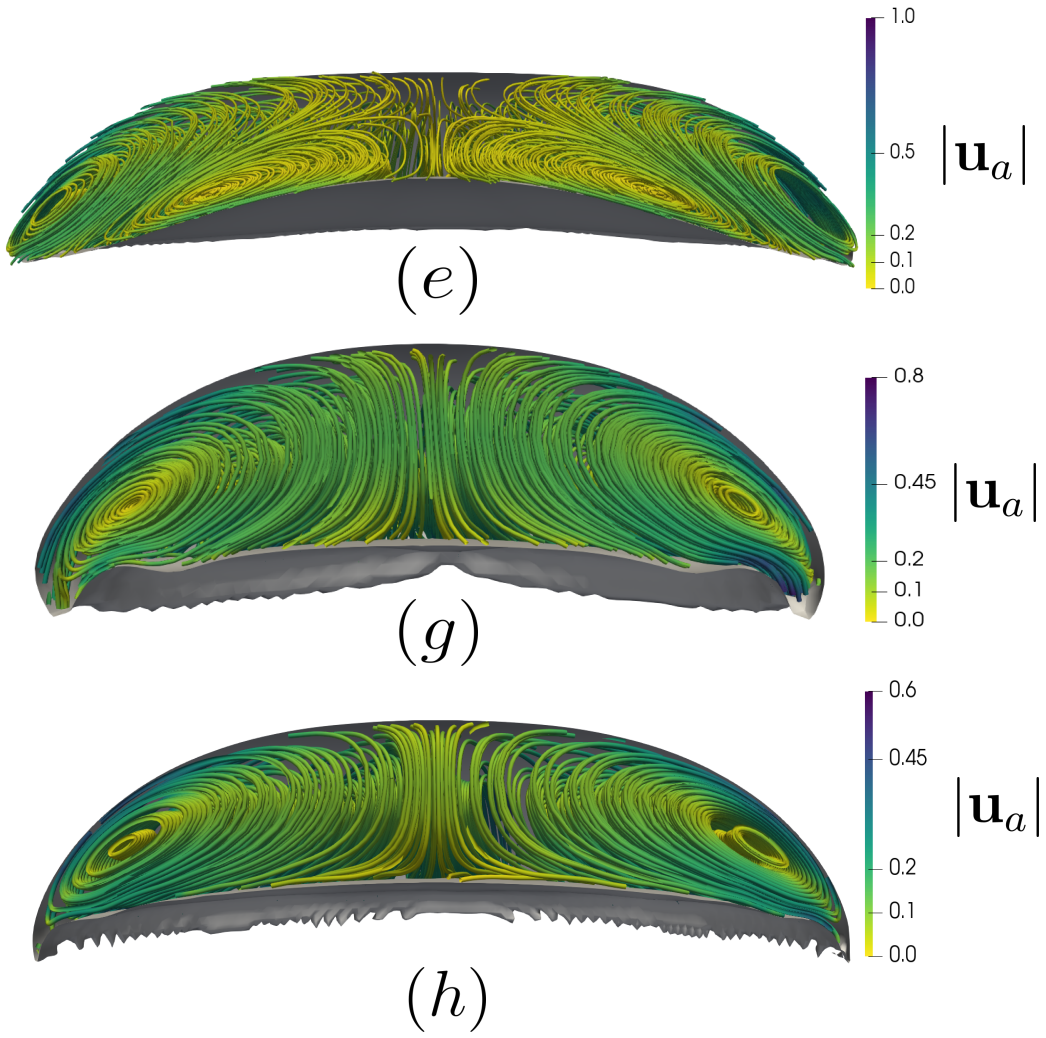

Figure 6.8: Final bubble shapes and interior flow fields corresponding to examples 2(e), 2(g), and 2(h) from Bhaga and Weber [12]. The streamlines are colored according to the magnitude of the apparent velocity $|\mathbf{u}_a|$, or velocity of the fluid in the reference frame of the rising bubble. Each bubble was initialized with a diameter of 1 cm. For clarity, each bubble is displayed sliced in half.

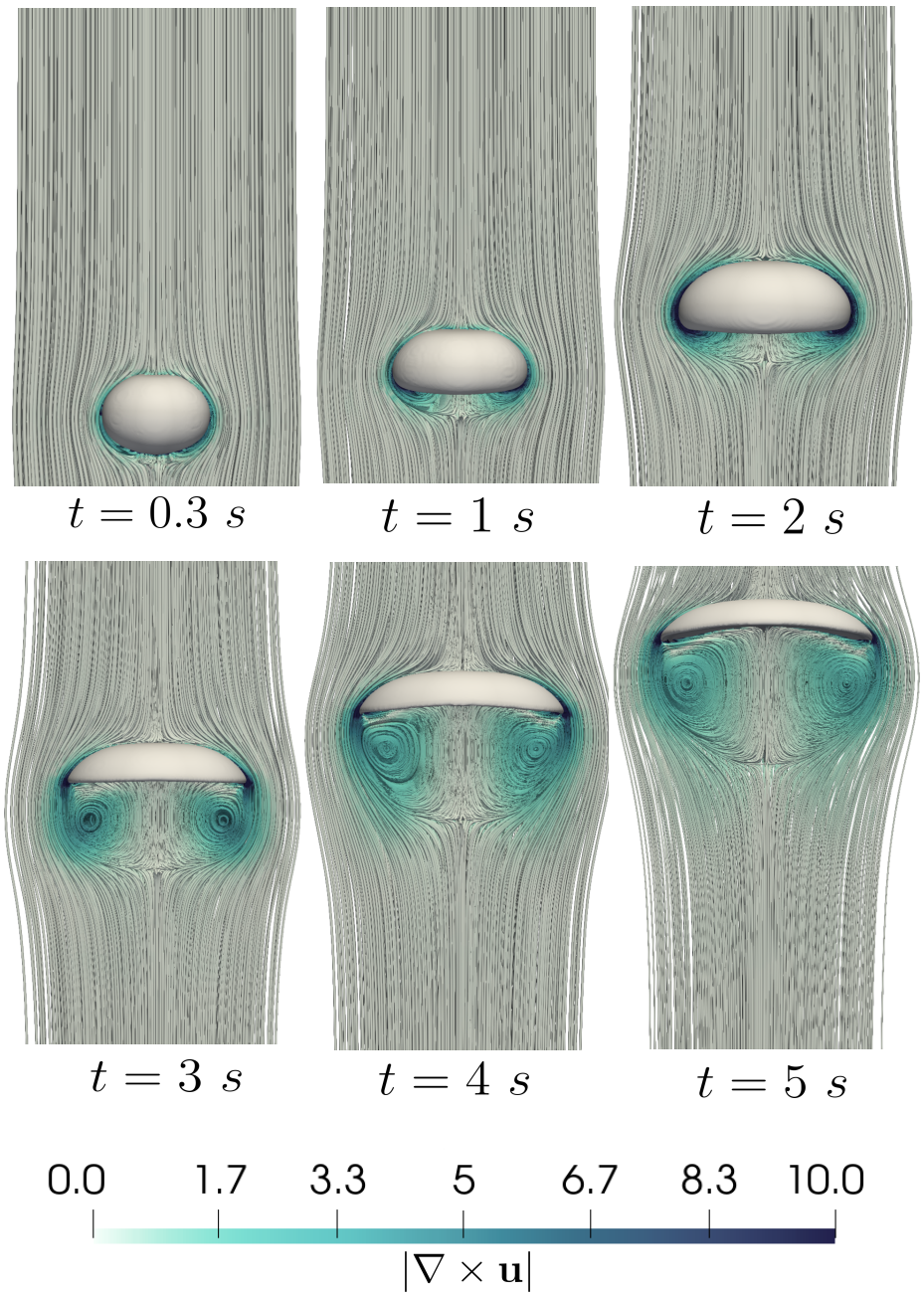

Figure 6.9: Time evolution of the instantaneous shape and apparent exterior flow for example 2(e) from Bhaga and Weber [12]. The flow streamlines in the suspending fluid are colored by the vorticity magnitude $|\nabla \times \mathbf{u}|$. The bubble was initialized with a diameter of 1 cm.

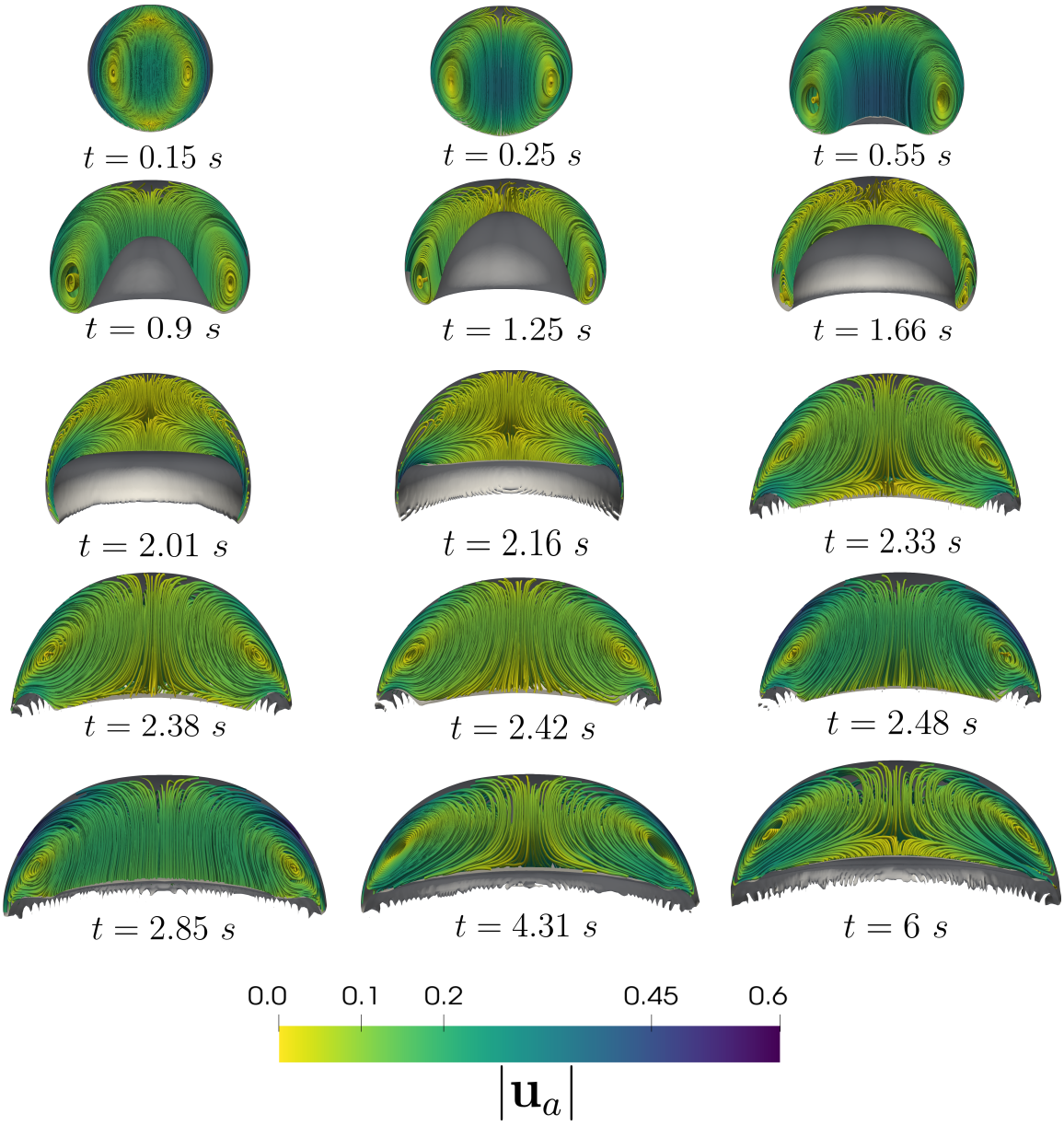

Figure 6.10: Time evolution of the instantaneous shape and apparent velocity for example 2(h) from Bhaga and Weber [12]. The streamlines are colored by the magnitude of the apparent velocity $|\mathbf{u}_a|$. The bubble was initialized with a diameter of 1 cm.

In table 6.5 we present a quantitative description of these 3D rising bubble simulations. The final simulation times were chosen as the times when the bubbles reached their asymptotic shape and rising velocity. For each case, we observe an asymptotic rising velocity close to 1, as expected. Moreover, each case demonstrated a small relative volume loss, which is measured at the final time as

$$\text{relative volume loss} = \frac{\frac{4}{3}\pi r_0^3 - \int_{\Omega^+} \mathrm{d}V}{\frac{4}{3}\pi r_0^3}, \tag{6.12}$$

where r_0 is the initial radius of the bubble. This behavior is expected given that the reference map used to track the interface is projected into the volume preserving space as part of the interface evolution algorithm. Finally, we present the CFL number and GV coefficient values used for determining the time step, and see that for most of the cases, larger time-steps were permissible. For cases (e), (g), and (h), we chose to use a smaller time-step restriction motivated by the choices made in [21] to ensure that numerical stability was maintained in these more complex cases. We note that the local temporal limiter discussed in section 5.7.2 was used for cases (e), (f), and (h) to ensure numerical stability. These results demonstrate the robustness and power of this numerical method to be used to simulate density and surface driven flows over a wide range of parameter regimes.

Case:	a	b	c	d	e	g	h
Final time (s)	3	3	6	5	6	4	6
Rising velocity (m/s)	1.16	1.11	1.04	1.08	0.83	0.99	0.9
Relative volume loss	0.2%	0.2%	0.2%	0.19%	0.05%	0.19%	0.05%
Max level	10	10	10	10	11	10	11
CFL number	1	1	1	1	0.25	0.45	0.45
GV coefficient	2	2	2	2	0.3	1	0.3

Table 6.5: Rising bubbles in 3D: parameters and measurements associated with the Bhaga Weber cases (a) - (h). The rising velocity is measured at the final time at the front tip of the bubble.

Case (f)

We choose to analyze case (f) separately from the other seven cases, as we observed considerably different behavior for the time evolution of case (f) compared to other numerical studies, such as [69]. Figure 6.11 shows the early time evolution of the instantaneous bubble shape. We observe that the middle section of the bubble becomes

incredibly thin as the bubble rises. At simulation time $t = 0.861\ s$, we observe that the tip of the bubble reaches such a thinning that we reach the limit of our resolution. At this point, the bubble's tip begins to break apart, similar to how the thin skirts of cases (e) and (h) break apart due to limits in resolution. Figure 6.12 shows this particular time step in 3D, and figure 6.13 shows a vertical slice in 2D of this time step, demonstrating the adaptive grid reaching a point where only a single grid node is resolving the bubble's tip. From this point, the tip of the bubble breaks apart and a hole forms in its place. This is shown in figure 6.14, which depicts the bubble's instantaneous shape at a later time once the tip has fully broken apart. Following this, the bubble is shown to undergo significant topological change as the hole at the tip expands to create three distinct segments at the top of the bubble, which all later start to break away, as shown in figure 6.15.

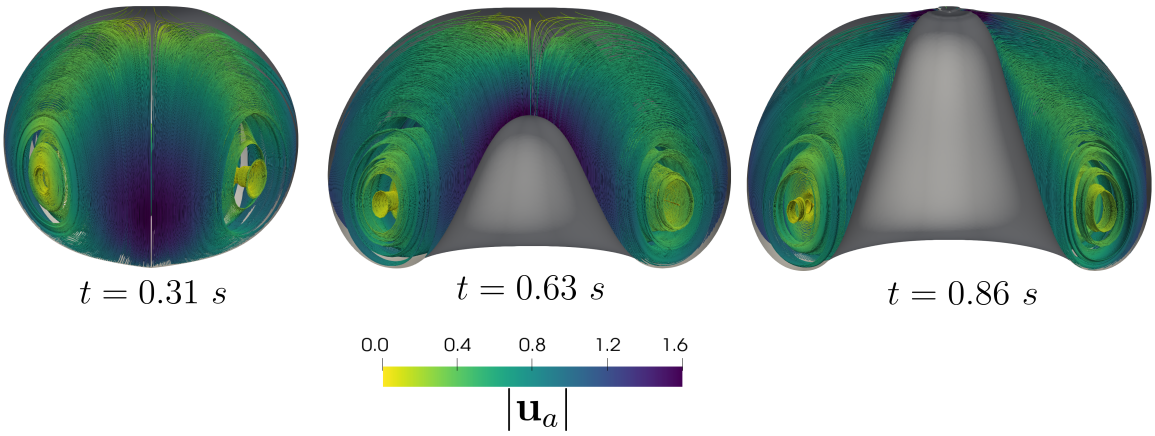

Figure 6.11: Time evolution of the instantaneous shape and apparent velocity for example 2(f) from Bhaga and Weber [12]. The streamlines are colored by the magnitude of the apparent velocity $|\mathbf{u}_a|$. The max level of this simulation is 12. The bubble was initialized with a diameter of 1 cm. For clarity, each bubble is displayed sliced in half.

There are several conclusions to draw from this case. The first is that our solver is capable of simulating significant topological changes and bubble breakup while maintaining numerical stability. This demonstrates our capability to resolve complex hydrodynamics involving considerable interfacial deformations. On the other hand, this case also highlights several limitations in our solver and computational approach. Case (f) as defined in [12] has the highest Reynolds number of Re $= 259$, and thus features the most momentum out of all of the cases. For each of these simulations, we choose the bubble's initial shape to be a sphere and we choose the initial velocity profile to be zero. In addition, other than gravity and interfacial tension, we do not incorporate any additional forces into our model. These choices may explain why the end shape of our simulation of

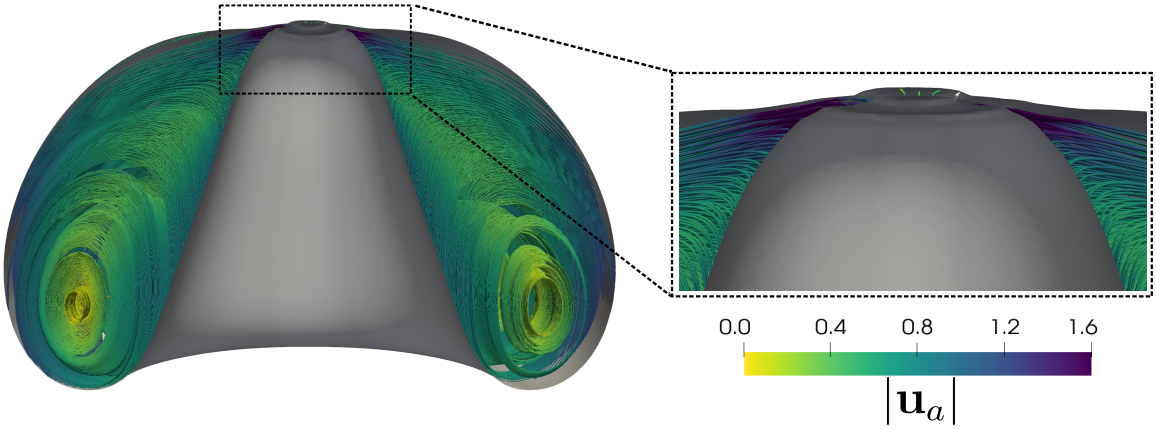

Figure 6.12: Instantaneous shape and apparent velocity for example 2(f) from Bhaga and Weber [12] at time $t = 0.861\ s$, the first time step at which the tip of the bubble begins to break apart. The full bubble is shown (left) along with a zoomed in view of the tip beginning to break apart (right). The streamlines are colored by the magnitude of the apparent velocity $|\mathbf{u}_a|$. The max level of this simulation is 12. For clarity, the full bubble is displayed sliced in half.

case (f) does not appear to resemble the experiment from [12] or past numerical studies such as [69] and why instead the bubble's tip becomes incredibly thin and reaches our resolution limit. Our future goal is to investigate different initial configurations that may lead to a more realistic terminal bubble shape.

Multiple bubbles

To further illustrate the capabilities of our numerical method, we simulate the flow of 20 rising bubbles with the same parameters as case (d) from [12], initialized in the domain with random initial positions and sizes. This simulation was performed on an octree grid with a minimum level of 4 and a maximum level of 10, where the grid was refined in regions near each bubble and in areas of high vorticity. We show the final configuration of this example in figure 6.16 and visualize the flow field through streamlines of velocity centered on each bubble, colored by the velocity magnitude. Here, in the zoomed region (a), we see an illustration of some asymmetries in the vertical rise of some bubbles due to the influence of nearby bubbles on the flow. We see that we are able to resolve complex multi-bubble interactions in regions where bubbles are close to one another, demonstrating the capability of our solver to effectively handle multi-scale hydrodynamics.

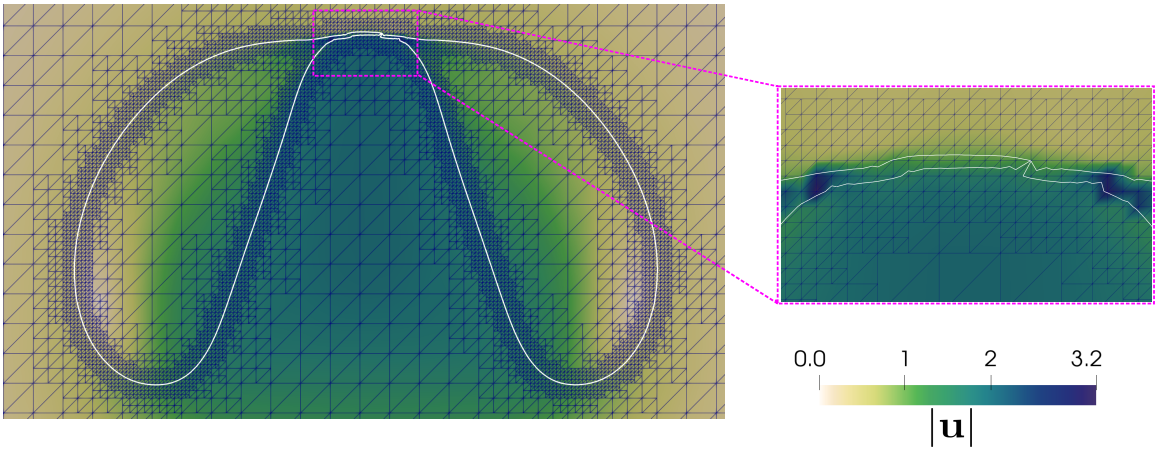

Figure 6.13: Vertical slice in 2D for example 2(f) from Bhaga and Weber [12] at time $t = 0.861\ s$, the first time step at which the tip of the bubble begins to break apart. The full slice is shown (left) along with a zoomed in view of the tip beginning to break apart (right). The adaptive grid is shown and the slice is colored by the velocity magnitude $|\mathbf{u}|$. The max level of this simulation is 12.

6.4.3 Rising bubbles past complex geometries

We conclude this section by simulating a rising bubble in a domain featuring solid flow obstructions: complex geometries that will remain stationary in the flow as the bubbles rise. We consider a rising bubble with the parameters of case (d) from Bhaga and Weber [12], and define solid obstructions using level set functions, as we do for the bubble interfaces. To treat the solid obstruction, we zero out all velocity fields everywhere in the domain where the obstruction level set functions are positive.

Shown in figure 6.17 is the time evolution of a single case (d) bubble placed in a flow with a solid flow obstruction featuring a converging conical section and a narrow pipe with a diameter half the size of the bubble's initial diameter. The bubble is shown to deform and change its shape to fit around the obstruction, and is shown to preserve its volume, as expected given our interface advection scheme and treatment of the solid obstruction. Five time steps from this example are highlighted in figure 6.18, demonstrating how the mesh adapts around the droplet as it changes its velocity and shape. We additionally highlight the flow field around the bubble in figure 6.20, where we show the velocity magnitude in a vertical slice at five times shown originally in figure 6.17. To quantify the accuracy of this example, in figure 6.19 we also show the relative mass loss of the bubble as it rises and deforms through its interactions with the flow obstruction. We observe excellent conservation of mass with deviations in mass at most

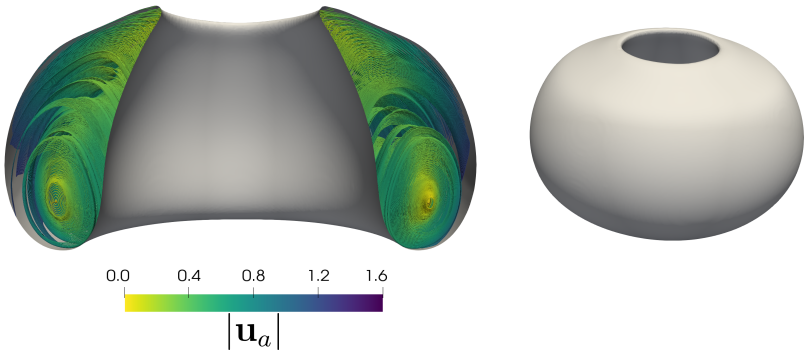

Figure 6.14: Instantaneous shape and apparent velocity for example 2(f) from Bhaga and Weber [12] at a later time of $t = 1.3\ s$, after the tip has fully broken apart (left). For clarity, the bubble is displayed sliced in half. The streamlines are colored by the magnitude of the apparent velocity $|\mathbf{u}_a|$. The instantaneous shape of the full bubble is also shown (right), depicting the full breakup of the bubble's tip. The max level of this simulation is 12.

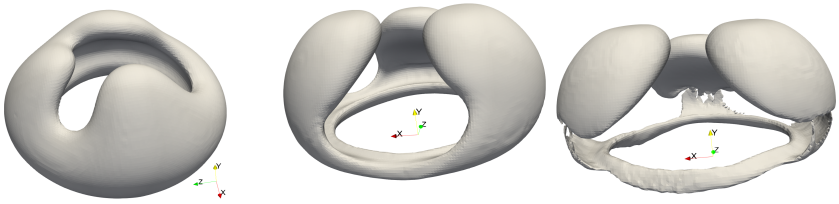

Figure 6.15: Time evolution of the instantaneous shape for example 2(f) from Bhaga and Weber [12] for later times after the initial breakup of the tip. The max level of this simulation is 11.

20% towards the end of the simulation and with most deviations close to zero. This example demonstrates the ability of our solver to simulate a two-phase flow impinging on an arbitrary irregular geometry, while maintaining numerical stability and thus illustrates that our approach can be used in the exploration of complex flow applications that feature both multi-phase flows and flow past solid obstructions.

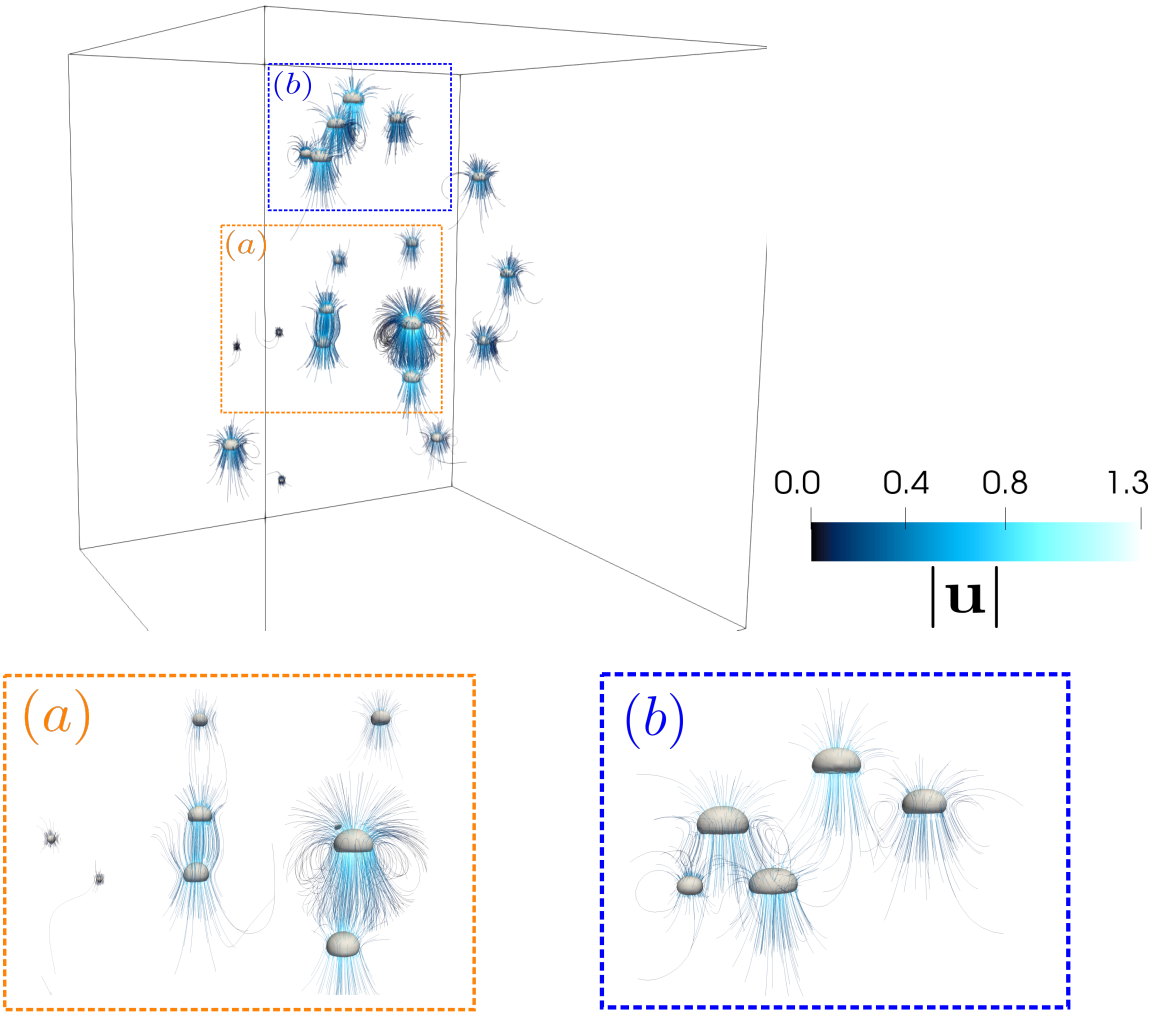

Figure 6.16: Final configuration in a 3D simulation of a collection of 20 rising bubbles with random initial positions and sizes. We visualize the flow field through streamlines of the velocity, colored by its magnitude $|\mathbf{u}|$. We show two regions, (a) and (b), zoomed in to further illustrate the complex flow structure in regions featuring multiple bubbles close to one another.

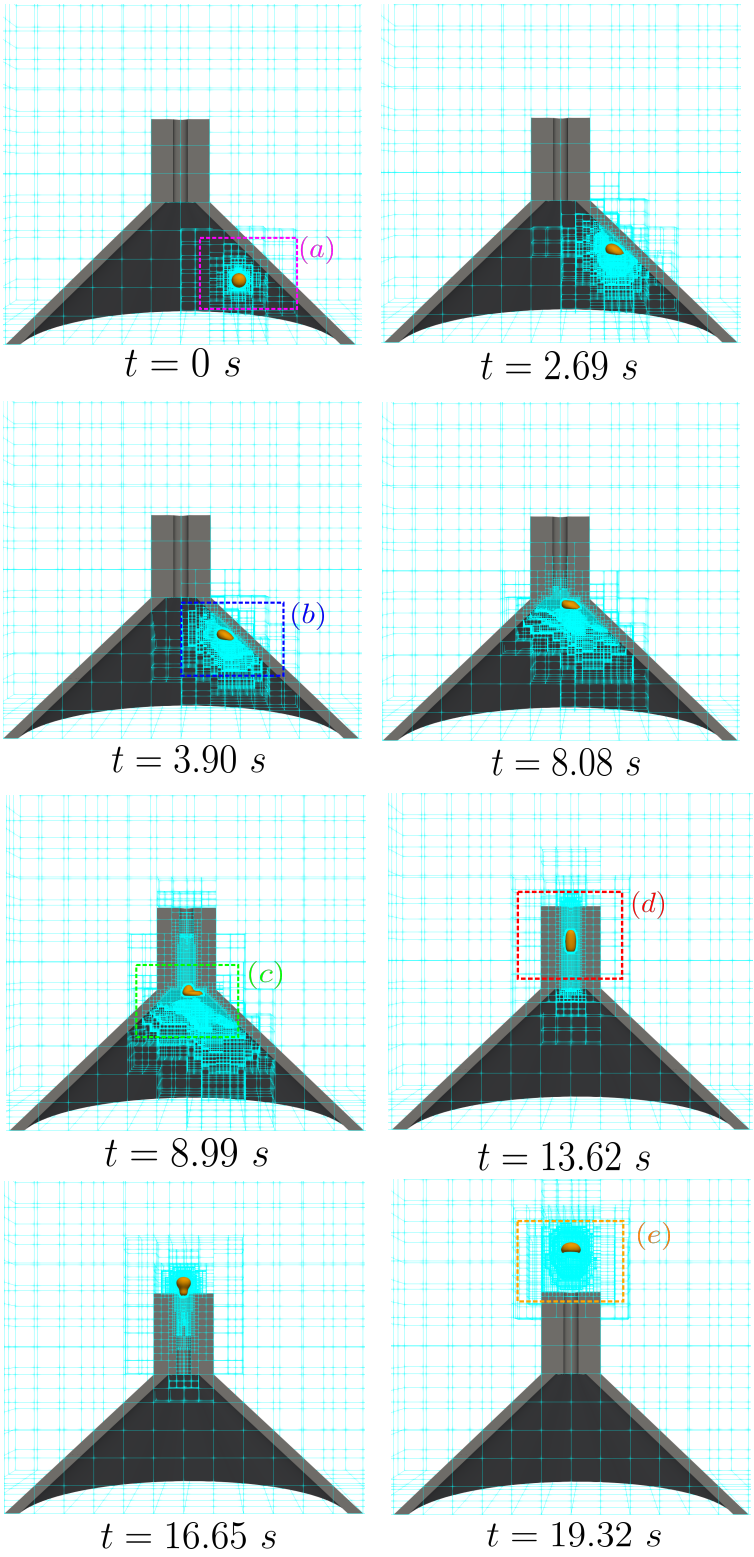

Figure 6.17: Time evolution of the shape and adaptive octree grid for a case (d) bubble from Bhaga and Weber [12] in a flow with a flow obstruction featuring a converging conical section and a narrow pipe. For clarity, the flow obstruction is displayed sliced in half. The droplet is shown to maintain its volume as it rises through the complex flow geometry. Five time steps labeled $(a) - (e)$ are identified with boxes and highlighted in figure 6.18.

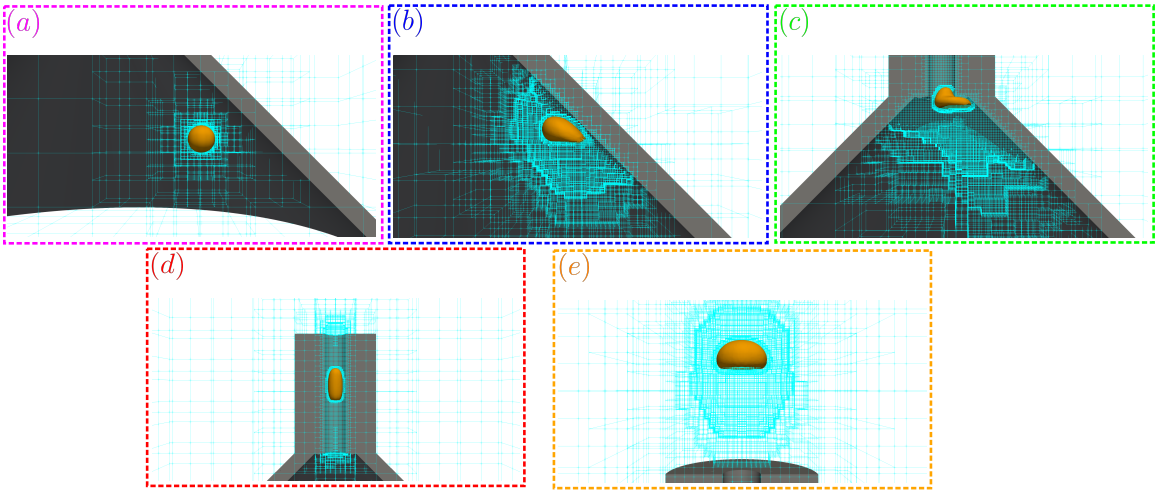

Figure 6.18: Zoomed in time steps from figure 6.17.

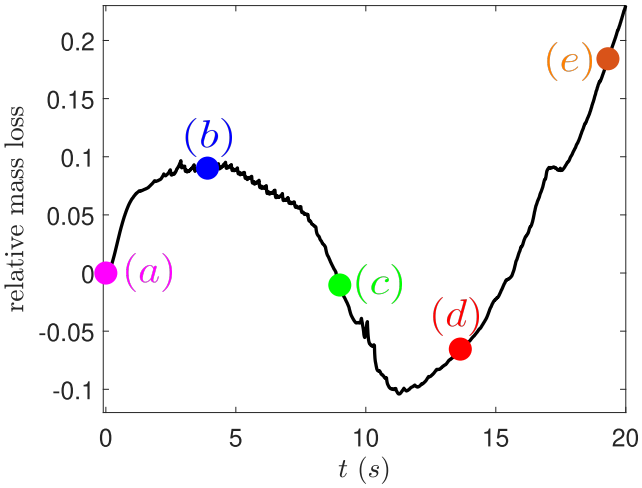

Figure 6.19: The relative mass loss of the rising bubble of the example shown in figure 6.17. The five time steps shown in more detail from figure 6.18 are identified with labeled plot markers.

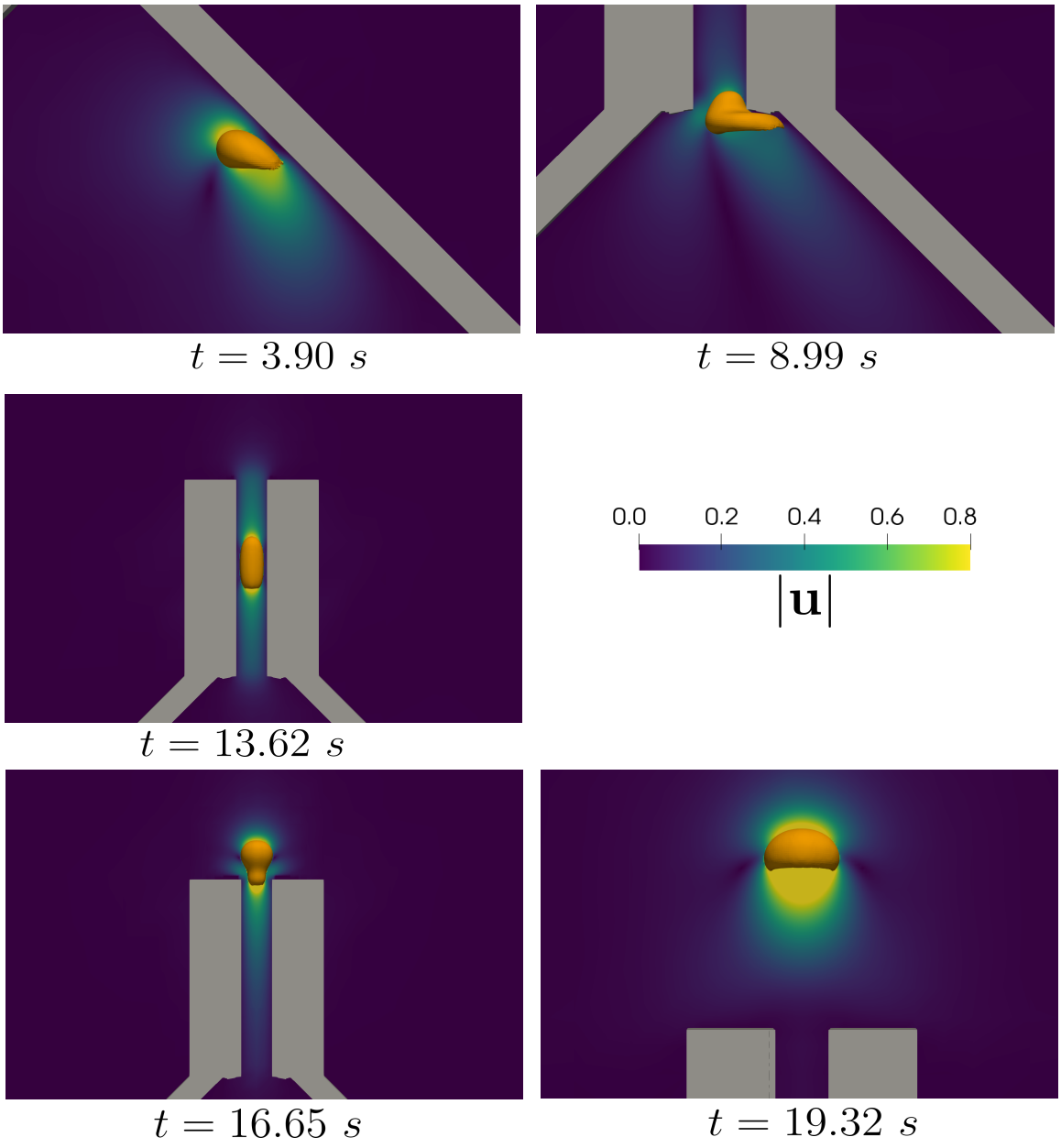

Figure 6.20: Time evolution of the velocity magnitude $|\mathbf{u}|$ shown in a slice in the vertical direction for a case (d) bubble from Bhaga and Weber [12] in a flow with a flow obstruction featuring a converging conical section and a narrow pipe.

6.5 Conclusions

We validated our two-phase nodal Navier-Stokes solver using several canonical two- and three-dimensional examples. We observe second order convergence, and we demonstrate the ability to accurately resolve surface tension and density driven flows, qualitatively and quantitatively matching experimental data on rising bubbles subject to strong and weak surface deformations. Finally, we demonstrated the capability of our solver to simulate complex flows of multiple rising bubbles and flow of bubbles rising past solid flow obstructions, displaying the potential of our solver to study a range of multi-physics applications.

There are several directions of future work for this project. To start, we wish to expand the capabilities of our solver to handle variable density flows. This would enable our solver to study atmospheric and oceanic applications featuring density stratified flows, such as rising oil droplets [20, 35, 59, 44], fine particle pollution in the atmosphere [71], and settling marine snow [53]. In addition, we would like to explore applications featuring non-uniform interfacial tension induced by effects such as the Marangoni effect. Also, we would like to explore additional approaches to treating the fluid-structure interaction, such as the approach using reference maps of [54], as this would expand the variety of applications that our method would be capable of resolving. Finally, we would like to parallelize our method on a distributed memory framework, given that at present we have only implemented parallelization on a shared-memory framework. This would enable our method to take better advantage of high performance computing resources and study a wider range of multi-physics applications.

For the work presented in this dissertation in particular, we believe that this incompressible two-phase flow Navier-Stokes solver is better suited to simulating the rising oil droplets experiment presented in chapters 2 and 3, due to its second order accuracy. However, at present, we have been unsuccessful in adapting this solver to simulate the rising oil droplet in a homogeneous ambient flow that was featured in section 3.4.1. When the density and viscosity values from this simulation are used instead of those featured in the examples of section 6.4.2, we observe far more significant surface deformations than what was observed in the simulations of chapter 2 and the experiments [43]. We believe that there is some sort of error in the implementation of our method or a potential instability that only manifests significantly when the density of the droplet is on the same order of magnitude as the ambient fluid, as we only validated our method against the experiments of [12], where the density of the rising bubbles is three orders of magnitude smaller than the surrounding fluid. Our immediate future work is to investigate this regime further and correct this issue to ensure that our solver can accurately simulate rising oil droplets.

Once our method is able to successfully simulate rising oil droplets in homogeneous ambient flows, the next direction for this project is to incorporate the ability to simulate a variable density ambient with the Boussinesq approximation, as was done for the method [69], and to re-run the validation simulations presented in chapter 2. Once these validation simulations are complete, we will then use our nodal method to solve the force decomposition presented in chapter 3 and perform an investigation into which surface forces primarily drive the entrainment of the oil droplet as it rises through the region of density stratification.

Chapter 7

Conclusion

Multi-phase flows are a significant component in many environmental systems, including atmospheric and oceanic systems featuring rising droplets and bubbles in stratified fluids. These applications are difficult and costly to fully study in an experimental manner, necessitating accurate and efficient numerical methods to simulate models that capture all the relevant multi-scale physics present. Motivated by modeling and simulating an experimental investigation into rising oil droplets in a density stratified flow, we developed a novel numerical method to simulate incompressible multi-phase fluid flow that is highly adaptable to studying a range of environmental applications.

The first project of this dissertation concerned the modeling and simulation of a rising oil droplet in a sharp two-layer density stratified flow. This problem serves as a means of isolating and better understanding the dynamics which drive droplet slowdown and entrainment when rising through regions of density stratification. We modeled the experiment of this problem with the incompressible multi-phase Navier-Stokes equations Eqs. (1.1)-(1.4) and simulated this model using the numerical method of [69]. These simulations showed good qualitative and quantitative agreement with the experiment of [43], demonstrating that we were able to successfully model this system and develop a computational framework for further study. To accompany this computational framework, we developed a model for surface forces impacting the rising droplet based on a decomposition of the pressure. This model was inspired by similar work for settling solid spheres in density stratified flows [78]. Despite the ability to capture the rising velocity trends found in the experiment with our simulations, we found the numerical method to be limited in its capability to solve our force decomposition model, given it only being a first order method. Nevertheless, we were able to validate the force decomposition model on a rising droplet in a homogeneous fluid, leaving future work in applying this model to a more accurate simulation of density stratified flow. We hope to use this model to quantify

the relative impact of each of the surface forces we identified in driving droplet slowdown and entrainment in the region of density stratification.

The second project of this dissertation concerned the development of a numerical method for solving the incompressible multi-phase Navier-Stokes equations, based on the method of [69], that achieves second order accuracy through collocated all computational variables at the nodes of the computational mesh. This method is a projection method that uses a collocated projection operator, that while only an approximate projection, converges to the canonical orthogonal projection when iterated. We initially developed this collocated projection operator for simulating incompressible single-phase flow in arbitrary domains [14], which we incorporated into a solver for the incompressible Navier-Stokes equations. This solver has second order accuracy and has a tremendously lower implementation cost compared to methods that employ staggered grids due to the collocation of all computational variables simplifying the discretization of the viscosity and projection operators and reducing the number of interpolations functions needed. Ultimately, our numerical method for incompressible multi-phase flow is the extension of our single-phase flow solver. We demonstrated that the nodal projection operator is stable in two-phase flow and validated our solver using several canonical two- and three-dimensional two-phase flows. Our solver achieves second order accuracy, is capable of accurately resolving density and surface tension driven flows, and also features the low implementation barrier present in our single-phase flow solver, making it an ideal tool for simulating environmental applications. We hope to continue our study of rising droplets and bubbles in environmental applications using this new solver, starting with returning to our preliminary numerical investigation of the rising oil droplet in a sharply stratified flow.

Bibliography

[1]N. Abaid, D. Adalsteinsson, A. Agyapong, and R. M. McLaughlin, "An internal splash: Levitation of falling spheres in stratified fluids", en, Physics of Fluids **16**, 1567–1580 (2004).

[2]D. Adalsteinsson and J. Sethian, "Transport and diffusion of material quantities on propagating interfaces via level set methods", en, Journal of Computational Physics **185**, 271–288 (2003).

[3]A. S. Almgren, J. B. Bell, and W. G. Szymczak, "A Numerical Method for the Incompressible Navier-Stokes Equations Based on an Approximate Projection", en, SIAM Journal on Scientific Computing **17**, 358–369 (1996).

[4]D. C. Assêncio and J. M. Teran, "A second order virtual node algorithm for Stokes flow problems with interfacial forces, discontinuous material properties and irregular domains", en, Journal of Computational Physics **250**, 77–105 (2013).

[5]F. Auguste and J. Magnaudet, "Path oscillations and enhanced drag of light rising spheres", en, Journal of Fluid Mechanics **841**, 228–266 (2018).

[6]K. Bäumler, M. Wegener, A. Paschedag, and E. Bänsch, "Drop rise velocities and fluid dynamic behavior in standard test systems for liquid/liquid extraction—experimental and numerical investigations", en, Chemical Engineering Science **66**, 426–439 (2011).

[7]M. Bayareh, A. Doostmohammadi, S. Dabiri, and A. M. Ardekani, "On the rising motion of a drop in stratified fluids", en, Physics of Fluids **25**, 103302 (2013).

[8]J. Bedrossian, J. H. Von Brecht, S. Zhu, E. Sifakis, and J. M. Teran, "A second order virtual node method for elliptic problems with interfaces and irregular domains", en, Journal of Computational Physics **229**, 6405–6426 (2010).

[9]T. Bellotti and M. Theillard, “A coupled level-set and reference map method for interface representation with applications to two-phase flows simulation”, en, Journal of Computational Physics **392**, 266–290 (2019).

[10]M. J. Berger and J. Oliger, “Adaptive mesh refinement for hyperbolic partial differential equations”, en, Journal of Computational Physics **53**, 484–512 (1984).

[11]E. Bertakis, S. Groß, J. Grande, O. Fortmeier, A. Reusken, and A. Pfennig, “Validated simulation of droplet sedimentation with finite-element and level-set methods”, en, Chemical Engineering Science **65**, 2037–2051 (2010).

[12]D. Bhaga and M. E. Weber, “Bubbles in viscous liquids: shapes, wakes and velocities”, en, Journal of Fluid Mechanics **105**, 61 (1981).

[13]F. Blanchette and A. M. Shapiro, “Drops settling in sharp stratification with and without Marangoni effects”, en, Physics of Fluids **24**, 042104 (2012).

[14]M. Blomquist, S. R. West, A. L. Binswanger, and M. Theillard, “Stable nodal projection method on octree grids”, en, Journal of Computational Physics **499**, 112695 (2024).

[15]K. Boukir, Y. Maday, B. Métivet, and E. Razafindrakoto, “A high-order characteristics/finite element method for the incompressible navier-stokes equations”, International Journal for Numerical Methods in Fluids **25**, 1421–1454 (1997).

[16]K. Boukir, Y. Maday, and B. Métivet, “A high order characteristics method for the incompressible navier—stokes equations”, Computer methods in applied mechanics and engineering **116**, 211–218 (1994).

[17]J. Brackbill, D. Kothe, and C. Zemach, “A continuum method for modeling surface tension”, en, Journal of Computational Physics **100**, 335–354 (1992).

[18]D. L. Brown, R. Cortez, and M. L. Minion, “Accurate Projection Methods for the Incompressible Navier–Stokes Equations”, en, Journal of Computational Physics **168**, 464–499 (2001).

[19]R. Camassa, C. Falcon, J. Lin, R. M. McLAUGHLIN, and N. Mykins, “A first-principle predictive theory for a sphere falling through sharply stratified fluid at low Reynolds number”, en, Journal of Fluid Mechanics **664**, 436–465 (2010).

[20]R. Camilli, C. M. Reddy, D. R. Yoerger, B. A. S. Van Mooy, M. V. Jakuba, J. C. Kinsey, C. P. McIntyre, S. P. Sylva, and J. V. Maloney, “Tracking Hydrocarbon Plume Transport and Biodegradation at Deepwater Horizon”, en, Science **330**, 201–204 (2010).

[21]H. Cho and M. Kang, “Fully implicit and accurate treatment of jump conditions for two-phase incompressible Navier–Stokes equations”, en, Journal of Computational Physics **445**, 110587 (2021).

[22]A. J. Chorin, “A numerical method for solving incompressible viscous flow problems”, en, Journal of Computational Physics **2**, 12–26 (1967).

[23]A. Doostmohammadi, S. Dabiri, and A. M. Ardekani, “A numerical study of the dynamics of a particle settling at moderate Reynolds numbers in a linearly stratified fluid”, en, Journal of Fluid Mechanics **750**, 5–32 (2014).

[24]R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, “A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method)”, en, Journal of Computational Physics **152**, 457–492 (1999).

[25]M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, and M. W. Williams, “A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework”, en, Journal of Computational Physics **213**, 141–173 (2006).

[26]C. Galusinski and P. Vigneaux, “On stability condition for bifluid flows with surface tension: Application to microfluidics”, en, Journal of Computational Physics **227**, 6140–6164 (2008).

[27]J. Glimm, O. McBryan, R. Menikoff, and D. H. Sharp, “Front Tracking Applied to Rayleigh–Taylor Instability”, en, SIAM Journal on Scientific and Statistical Computing **7**, 230–251 (1986).

[28]P. Gómez, C. Zanzi, J. López, and J. Hernández, “Simulation of high density ratio interfacial flows on cell vertex/edge-based staggered octree grids with second-order discretization at irregular nodes”, en, Journal of Computational Physics **376**, 478–507 (2019).

[29]A. Guittet, M. Theillard, and F. Gibou, “A stable projection method for the incompressible Navier–Stokes equations on arbitrary geometries and adaptive Quad/Octrees”, en, Journal of Computational Physics **292**, 215–238 (2015).

[30]F. H. Harlow and J. E. Welch, “Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface”, en, Physics of Fluids **8**, 2182 (1965).

[31]J. L. Hellrung, L. Wang, E. Sifakis, and J. M. Teran, “A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions”, en, Journal of Computational Physics **231**, 2015–2048 (2012).

[32]M. Horowitz and C. H. K. Williamson, “The effect of Reynolds number on the dynamics and wakes of freely rising and falling spheres”, en, Journal of Fluid Mechanics **651**, 251–294 (2010).

[33]K. Kamrin, C. H. Rycroft, and J.-C. Nave, "Reference map technique for finite-strain elasticity and fluid–solid interaction", en, Journal of the Mechanics and Physics of Solids **60**, 1952–1969 (2012).

[34]M. Kang, R. P. Fedkiw, and X.-D. Liu, "A Boundary Condition Capturing Method for Multiphase Incompressible Flow", Journal of Scientific Computing **15**, 323–360 (2000).

[35]J. D. Kessler, D. L. Valentine, M. C. Redmond, M. Du, E. W. Chan, S. D. Mendes, E. W. Quiroz, C. J. Villanueva, S. S. Shusta, L. M. Werra, S. A. Yvon-Lewis, and T. C. Weber, "A Persistent Oxygen Anomaly Reveals the Fate of Spilled Methane in the Deep Gulf of Mexico", en, Science **331**, 312–315 (2011).

[36]H. Lamb, *Hydrodynamics* (University Press, 1924).

[37]L. Lee and R. J. LeVeque, "An Immersed Interface Method for Incompressible Navier–Stokes Equations", en, SIAM Journal on Scientific Computing **25**, 832–856 (2003).

[38]R. J. LeVeque and Z. Li, "Immersed Interface Methods for Stokes Flow with Elastic Boundaries or Surface Tension", en, SIAM Journal on Scientific Computing **18**, 709–735 (1997).

[39]R. J. LeVeque and Z. Li, "The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources", en, SIAM Journal on Numerical Analysis **31**, 1019–1044 (1994).

[40]Z. Li and M.-C. Lai, "The Immersed Interface Method for the Navier–Stokes Equations with Singular Forces", en, Journal of Computational Physics **171**, 822–842 (2001).

[41]X. Long and C. Chen, "General Formulation of Second-Order Semi-Lagrangian Methods for Convection-Diffusion Problems", en, Abstract and Applied Analysis **2013**, 1–10 (2013).

[42]J. Magnaudet and M. J. Mercier, "Particles, Drops, and Bubbles Moving Across Sharp Interfaces and Stratified Layers", en, Annual Review of Fluid Mechanics **52**, 61–91 (2020).

[43]T. L. Mandel, D. Z. Zhou, L. Waldrop, M. Theillard, D. Kleckner, and S. Khatri, "Retention of rising droplets in density stratification", en, Physical Review Fluids **5**, 124803 (2020).

[44]M. K. McNutt, R. Camilli, T. J. Crone, G. D. Guthrie, P. A. Hsieh, T. B. Ryerson, O. Savas, and F. Shaffer, "Review of flow rate estimates of the *Deepwater Horizon* oil spill", en, Proceedings of the National Academy of Sciences **109**, 20260–20267 (2012).

[45]R. Mehaddi, F. Candelier, and B. Mehlig, "Inertial drag on a sphere settling in a stratified fluid", en, Journal of Fluid Mechanics **855**, 1074–1087 (2018).

[46]C. Min and F. Gibou, "A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids", en, Journal of Computational Physics **219**, 912–929 (2006).

[47]C. Min, F. Gibou, and H. D. Ceniceros, "A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids", en, Journal of Computational Physics **218**, 123–140 (2006).

[48]S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations", en, Journal of Computational Physics **79**, 12–49 (1988).

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

[50]J.-P. Pons, G. Hermosillo, R. Keriven, and O. Faugeras, "Maintaining the point correspondence in the level set framework", en, Journal of Computational Physics **220**, 339–354 (2006).

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

[52]S. Popinet and S. Zaleski, "A front-tracking algorithm for accurate representation of surface tension", International Journal for Numerical Methods in Fluids **30**, 775–793 (1999).

[53]J. Prairie, K. Ziervogel, C. Arnosti, R. Camassa, C. Falcon, S. Khatri, R. McLaughlin, B. White, and S. Yu, "Delayed settling of marine snow at sharp density transitions driven by fluid entrainment and diffusion-limited retention", en, Marine Ecology Progress Series **487**, 185–200 (2013).

[54]C. H. Rycroft and K. Kamrin, "Eulerian method for multiphase interactions of soft solid bodies in fluids", (2015).

[55]H. Samet, "An Overview of Quadtrees, Octrees, and Related Hierarchical Data Structures", en, in *Theoretical Foundations of Computer Graphics and CAD*, edited by R. A. Earnshaw (Springer Berlin Heidelberg, Berlin, Heidelberg, 1988), pp. 51–68.

[56]R. Saye, "Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: Part I", en, Journal of Computational Physics **344**, 647–682 (2017).

[57]C. Schroeder, A. Stomakhin, R. Howes, and J. M. Teran, "A second order virtual node algorithm for Navier–Stokes flow problems with interfacial forces and discontinuous material properties", en, Journal of Computational Physics **265**, 221–245 (2014).

[58]V. A. Shaik and A. M. Ardekani, "Drag, deformation, and drift volume associated with a drop rising in a density stratified fluid", en, Physical Review Fluids **5**, 013604 (2020).

[59]S. A. Socolofsky and E. E. Adams, "Multi-phase plumes in uniform and stratified crossflow", en, Journal of Hydraulic Research **40**, 661–672 (2002).

[60]E. A. Spiegel and G. Veronis, "On the Boussinesq Approximation for a Compressible Fluid.", en, The Astrophysical Journal **131**, 442 (1960).

[61]A. N. Srdić-Mitrović, N. A. Mohamed, and H. J. S. Fernando, "Gravitational settling of particles through density interfaces", en, Journal of Fluid Mechanics **381**, 175–198 (1999).

[62]M. Sussman, K. Smith, M. Hussaini, M. Ohta, and R. Zhi-Wei, "A sharp interface method for incompressible two-phase flows", en, Journal of Computational Physics **221**, 469–505 (2007).

[63]M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, "An Adaptive Level Set Approach for Incompressible Two-Phase Flows", en, Journal of Computational Physics **148**, 81–124 (1999).

[64]M. Sussman, E. Fatemi, P. Smereka, and S. Osher, "An improved level set method for incompressible two-phase flows", en, Computers & Fluids **27**, 663–680 (1998).

[65]M. Sussman, P. Smereka, and S. Osher, "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow", en, Journal of Computational Physics **114**, 146–159 (1994).

[66]Z. Tan, D. Le, Z. Li, K. Lim, and B. Khoo, "An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane", en, Journal of Computational Physics **227**, 9955–9983 (2008).

[67]Z. Tan, D. Le, K. Lim, and B. Khoo, "An Immersed Interface Method for the Incompressible Navier–Stokes Equations with Discontinuous Viscosity Across the Interface", en, SIAM Journal on Scientific Computing **31**, 1798–1819 (2009).

[68]M. Theillard, "A volume-preserving reference map method for the level set representation", en, Journal of Computational Physics **442**, 110478 (2021).

[69]M. Theillard, F. Gibou, and D. Saintillan, "Sharp numerical simulation of incompressible two-phase flows", en, Journal of Computational Physics **391**, 91–118 (2019).

[70]G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan, "A Front-Tracking Method for the Computations of Multiphase Flow", en, Journal of Computational Physics **169**, 708–759 (2001).

[71]R. P. Turco, O. B. Toon, T. P. Ackerman, J. B. Pollack, and C. Sagan, "Climate and Smoke: an Appraisal of Nuclear Winter", en, Science **247**, 166–176 (1990).

[72]S. O. Unverdi and G. Tryggvason, "A front-tracking method for viscous, incompressible, multi-fluid flows", en, Journal of Computational Physics **100**, 25–37 (1992).

[73]L. Verso, M. v. Reeuwijk, and A. Liberzon, "Transient stratification force on particles crossing a density interface", en, International Journal of Multiphase Flow **121**, 103109 (2019).

[74]M. Wegener, M. Kraume, and A. R. Paschedag, "Terminal and transient drop rise velocity of single toluene droplets in water", en, AIChE Journal, NA–NA (2009).

[75]D. Xiu and G. E. Karniadakis, "A Semi-Lagrangian High-Order Method for Navier–Stokes Equations", en, Journal of Computational Physics **172**, 658–684 (2001).

[76]J.-J. Xu and H.-K. Zhao, "An Eulerian Formulation for Solving Partial Differential Equations Along a Moving Interface", Journal of Scientific Computing **19**, 573–594 (2003).

[77]K. Y. Yick, C. R. Torres, T. Peacock, and R. Stocker, "Enhanced drag of a sphere settling in a stratified fluid at small Reynolds numbers", en, Journal of Fluid Mechanics **632**, 49–68 (2009).

[78]J. Zhang, M. J. Mercier, and J. Magnaudet, "Core mechanisms of drag enhancement on bodies settling in a stratified fluid", en, Journal of Fluid Mechanics **875**, 622–656 (2019).

Appendix A

Appendix for Chapter 4

A.1 Proof: divergence is negative transpose of gradient

For a scalar function $\psi(\mathbf{x})$ and a vector field $\mathbf{V}(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^{2,3}$, the following identity holds true

$$\nabla \cdot (\psi \mathbf{V}) = \psi \nabla \cdot \mathbf{V} + \mathbf{V} \cdot \nabla \psi, \tag{A.1}$$

due to the product rule. The adjoint of the linear operator $\mathbf{A}$ over an inner product is defined as the operator $\mathbf{A}^*$ such that

$$\langle \mathbf{A}x, y \rangle = \langle x, \mathbf{A}^* y \rangle, \tag{A.2}$$

where $\langle \cdot, \cdot \rangle$ is the inner product on that space. Then if we have the linear operators representing the divergence $\mathcal{D} = \nabla \cdot$ and gradient $\mathcal{G} = \nabla$, to show that $-\mathcal{G} = \mathcal{D}^{\mathrm{T}}$, we need to show that

$$\langle \mathcal{D}\mathbf{x}, y \rangle = \langle \mathbf{x}, -\mathcal{G}y \rangle, \tag{A.3}$$

where $\mathbf{x}$ is a vector field and y is a scalar field, both defined in $\mathbb{R}^{2,3}$, and the inner product is defined as

$$\langle x, y \rangle = \int_\Omega xy \mathrm{d}V, \tag{A.4}$$

for $\Omega \in \mathbb{R}^{2,3}$. We do this by using the identity (A.1) as

$$\langle \mathcal{D}\mathbf{x}, y \rangle = \int_\Omega (\nabla \cdot \mathbf{x}) y \mathrm{d}V = \int_\Omega \nabla \cdot (y\mathbf{x}) - \mathbf{x} \cdot \nabla y \mathrm{d}V = \int_\Omega \nabla \cdot (y\mathbf{x}) \mathrm{d}V - \int_\Omega \mathbf{x} \cdot \nabla y \mathrm{d}V \tag{A.5}$$

Now using the Divergence Theorem, this is rewritten as

$$\langle \mathcal{D}\mathbf{x}, y \rangle = \int_{\partial\Omega} (y\mathbf{x}) \cdot \mathbf{n} \mathrm{d}S - \int_\Omega \mathbf{x} \cdot \nabla y \mathrm{d}V, \tag{A.6}$$

where $\partial\Omega$ is the boundary of Ω and $\mathbf{n}$ is the outward facing normal on this boundary. Then if $\mathbf{x}$ and y are given such that the boundary term is zero, we obtain

$$\langle \mathcal{D}\mathbf{x}, y\rangle = \int_{\Omega} -\mathbf{x}\cdot\nabla y \mathrm{d}V = \langle \mathbf{x}, -\mathcal{G}y\rangle, \tag{A.7}$$

thus $\mathcal{D}^{\mathrm{T}} = -\mathcal{G}$. □