FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

AN ASYMPTOTICALLY PRESERVING METHOD FOR MULTIPHASE FLOW

By

MATTHEW JEMISON

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

Degree Awarded:
Summer Semester, 2014

Matthew Jemison defended this dissertation on June 3, 2014.
The members of the supervisory committee were:

Mark Sussman

Professor Directing Thesis

Doron Nof

University Representative

Nick Cogan

Committee Member

Kyle Gallivan

Committee Member

Xiaoming Wang

Committee Member

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

To my wife, for never failing to believe in me and all the other ways, beyond number, you helped me to get here. To my family, for all of your love and support.

ACKNOWLEDGMENTS

I would like to acknowledge my gratitude to my committee members, Nick Cogan, Kyle Gallivan, Doron Nof, and Xiaoming Wang. I would like to thank mentor Mikhail Shashkov, for convincing me to not let a good idea die. Finally, I would like to thank my thesis advisor, Mark Sussman. His inspirational work led me to pursue my interest in computational mathematics. His willingness and patience to answer any question made this work possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

A unified, asymptotically-preserving method for simulating multiphase flows using an exactly mass, momentum, and energy conserving Cell-Integrated Semi-Lagrangian advection algorithm is presented. The new algorithm uses a semi-implicit pressure update scheme that asymptotically preserves the standard incompressible pressure projection method in the limit of infinite sound speed. The asymptotically preserving attribute makes the new method applicable to compressible and incompressible flows, including stiff materials, which enables large time steps characteristic of incompressible flow algorithms rather than the small time steps required by explicit methods. Shocks are captured and material discontinuities are tracked, without the aid of any approximate or exact Riemann solvers. The new method enables one to simulate the flow of multiple materials, each possessing a potentially exotic equation of state. Simulations of multiphase flow in one and two dimensions are presented which illustrate the effectiveness of the new algorithm at efficiently computing multiphase flows containing shock waves and material discontinuities with large "impedance mismatch." Additionally, new techniques related to the Moment-of-Fluid interface reconstruction are presented, including a novel, asymptotically-preserving method for capturing "filaments," and an improved method for initializing the Moment-of-Fluid optimization problem on unstructured, triangular grids.

CHAPTER 1

INTRODUCTION

There are many applications in science and industry that necessitate the understanding of compressible, multiphase flows. Examples include underwater explosions and implosions [83, 82, 22, 41], bubble dynamics [68, 43, 15, 1], shock wave lithoptripsy [62, 40], atomization and spray in internal-combustion engines [6, 11, 71], and laser induced melting [88, 67].

Since the development of of the Implicit Continuous-Fluid Eulerian (ICE) method [32] in the late 1960's, there have been many new algorithms developed for simulating compressible multiphase flow [32, 23, 82, 68, 43, 87, 57, 41, 49, 27, 11, 22, 15, 1]. A summary of the key properties of algorithms that have been developed for simulating compressible multiphase flows are given chronologically in Table 1.1.

Asymptotic preservation is defined in the sense of Degond, *et. al.* [19]. If the method is expressed as a perturbation in terms of the sound speed, such that in the limit as squared sound speed $c^2 \to \infty$, the incompressible pressure projection method is recovered, then the method is said to be "asymptotically preserving."

ICE [32] (1968) is an implicit, asymptotically-preserving method for simulating multiphase flows with an arbitrary equation of state (EOS), using Lagrangian Particles to track the interface. The implicit discretization in [32] leads to strong coupling between equations. The Ghost Fluid Method (GFM) introduced in [23] (1999) is a general approach, in the context of level set methods, for asserting material jump conditions, without explicitly finding/reconstructing the interface. The Ghost Fluid Method removes spurious oscillations seen at contact discontinuities in many conservative methods. [83, 82] (1998) Developed an explicit, ALE+Remap algorithm for simulating underwater explosions. The ALE+Remap algorithm maintains the material discontinuity aligned with the computational grid, thereby preserving consistency of the jump conditions at the material interface.

As an alternative to a sharp interface representation, the material interface can be modeled as a "diffusion zone" [68] (1999) where density and energy undergo a rapid transition. The explicit

Table 1.1: Chronological listing of methods in compressible multiphase flows.

Authors	Interface Representation	Interface Treatment	Spatially Uniform Gas ρ/p	Mass of Each Material Conserved	Asymptotically Preserving
Harlow and Amsden [32]	Lagrangian Particles	FD	No	No	Yes
Fedkiw, *et. al.* [23]	Level-Set (LS)	Ghost Fluid Method (GFM)	No	No	No
Wardlaw and Mair [82]	ALE+Remap	FT	Constant/variable cases tested	Yes	No
Saurel and Abgrall [68]	Diffuse Interface (DI)	Riemann Solver/ Godunov	No	No	No
Koren, *et. al.* [43]	LS	FV-GFM	No	No	No
Yabe, *et. al.* [87]	DI	CIP	No	Total Mass Conserved	Yes
Nourgaliev, *et. al.* [57]	LS	FT	No	No	No
Kadioglu, *et. al.* [41]	CLSVOF	GFM	Yes	No	Yes
Galera, *et.al.* [27]	MOF	ALE	No	Yes	No
Bo, *et. al.* [11]	FT	GFM	No	No	No
Farhat, *et. al.* [22]	LS-Two Phase Riemann Solver	FV/Riemann Solver	No	No	No
Chang, *et. al.* [15]	LS	DNS-FT	No	Yes	No
Aanjaneya, *et. al.* [1]	LS	GFM	Yes	No	Yes

pressure relaxation technique developed in [68] enforces that uniform pressure and velocity are maintained from one time step to the next by loosening the requirement of exact conservation of mass, momentum, and energy. Koren *et. al.* [43] (2002) use a linearized, two-fluid Osher scheme along with a variant of the Ghost Fluid method to compute, explicitly, compressible, low-Mach number, water-air flows. The semi-implicit method of Yabe, *et. al.* [87] (2002) uses a diffuse interface-style method to represent a material interface, while applying Constrained Interpolation Profile (CIP) methods for sharp front capturing. The explicit Characteristic-Based Matching (CBM) method proposed in [57] (2004) uses a two-fluid Riemann solver to determine the material interface velocity and fluxes neighboring cut cells. The interface is represented by a level-set function in Nourgaliev and Dinh's work. [57] report that their characteristic-based method improve the performance of their method over the GFM method when there is a material discontinuity in the presence of strong shocks. Kadioglu, *et. al.* [41] developed a semi-implicit method to capture

sharp shock fronts and material discontinuities, while taking larger time steps than allowed by the standard explicit CFL restriction. Behavior of low Mach-number simulations in [41] was comparable to [43], while exhibiting more favorable mass conservation. The explicit method of Galera, *et. al.* [27] (2011) employs a cell-centered ALE method, using an approximate Riemann solver to compute the nodal velocity and the moment-of-fluid (MOF) method to represent the deforming material(s). The algorithm by Galera et al. exactly preserves the mass of all materials and exploits the accurate multimaterial reconstruction property of MOF making it well-suited to deformational, multi-material flows. The explicit method of Bo, *et. al.* [11] (2011) combines the GFM with front tracking, solving the two-fluid Riemann problem at the interface to find the ghost states, minimizing spurious pressure oscillation at the interface. Farhat, *et. al.* [22] (2012) develop an explicit, finite-volume approach that uses a two-fluid Riemann solver for more exotic equations of state (e.g. Jones-Wilkins-Lee), tabulating Riemann invariants offline to avoid expensive runtime calculations in simulating underwater explosions. Chang, *et. al.* [15] (2013) points out problems inherent with both diffuse interface and ghost fluid methods, and developed an all-speed, explicit DNS method for simulating multiphase flows. Emphasis is placed on exact conservation of mass, momentum, and energy. Aanjaneya, *et. al.* [1] (2013) simulate low-Mach number bubble dynamics using a monolithic, asymptotically preserving, coupled compressible gas/incompressible liquid algorithm.

Note from Table 1.1 that few compressible multiphase flow methods are asymptotically preserving. Those that are asymptotically preserving may not conserve mass of each material individually, may make simplifying assumptions (such as constant gas density and pressure), or may cast the problem in non-conservative form. The following are single-phase methods that exhibit asymptotic preservation in the limit of infinite sound speed. Wesseling [84] and Xiao [85, 86] developed unified methods for single-phase flow that handle both incompressible and compressible regimes, using an update for pressure that simplifies to the incompressible projection technique of Chorin [17], or the pressure Poisson equation, in the limit of zero Mach number, incompressible flow. Work by Martineau, *et. al.* [51] extends the ICE method [32] to use a semi-implicit pressure correction, allowing robust, single-phase simulation for both compressible and nearly incompressible flows. The semi-implicit pressure update introduced for single phase flow by Kwatra *et. al.* in [44] is utilized by Aanjaneya *et. al.* in [1] to simulate multiphase flow, asymptotically preserving the incompressible pressure projection method in the limit of zero Mach number as well. The method

of Lentine, *et. al.* [47] uses compressible Semi-Lagrangian advection for single-phase Euler and Navier-Stokes flows, in conjunction with an implicit pressure solve to allow for long time steps. Work by Gretarsson and Fedkiw [39] extends the methods of [47], which performed computation with embedded solids, to perform single-phase flow computation in the presence of thin, rigid, leak-proof structures. Additionally, [39] uses a high-order ENO, flux-based solver in the bulk region and a Semi-Lagrangian method near the structure to prevent penetration of the solid.

For deformational multi-material simulations, the accurate reconstruction of the material interface is important. In the field of interface reconstruction, the problem of a material configuration under-resolved by the mesh is often overcome by increasing mesh resolution. Regions of high-curvature can be detected and captured by using Adaptive Mesh Refinement [76, 3, 58], hybrid particle level set method[21], or by introducing oriented cell subzones[24, 25]. However, in the case of thin structures, it may be impractical to increase mesh resolution to fully resolve the structure.

Thin structures in material configuration can lead to reconstruction errors, resulting in erroneous material breakup [14, 29, 26]. Piecewise-linear interface reconstruction techniques that use non-local information to determine interface orientation, such as Volume of Fluid (VOF) and Young's method, exhibit "numerical surface tension," an artifact of the interface reconstruction method that can induce erroneous breakup of such thin structures.

The method of [29] examines the material occupying a shared node with respect to neighboring cells and detects discrepancies indicating interfacial breakup. In cells where a discrepancy is detected, interface smoothing is applied to repair topology after advection. Similarly, the method of [4] uses volume fraction data and material ID information at vertices to determine interface topology, generate a discretized material interface from a look-up table, and last perform a smooth, volume-accurate material interface reconstruction. The techniques in [29] and [4] do not ultimately remove the problems of filaments thinner than grid cells.

Increasing the sophistication of the volume-of-fluid reconstruction algorithm, either by piecewise parabolic reconstruction[66] or by the advecting normals technique[65], improve the performance of the volume-of-fluid method for simulating surface tension driven free surface flows, but do not address the problems when a filament is thinner than the mesh size.

High order level set methods[56, 59, 77] naturally represent filaments even when a filament is thinner than a grid cell. Unfortunately, large gradients in the level set function can develop[78]

in a flow with non-uniform velocity, thereby defeating the benefit of high order level set advection schemes. In order to prevent the build-up of large gradients in the level set function, the level set function is often reinitialized to be a distance function[78, 61]. Unfortunately, whereby reinitialization schemes prevent the buildup of large gradients in the level set function where the interface is smooth, the level set gradient in underresolved filamentary regions will be increased, defeating the purpose of a high order method.

A new method for capturing filaments using the existing functionality of the Multimaterial Moment-of-Fluid method in a unique way to reconstruct filamentary interfaces under-resolved by the mesh is developed in this work. Rather than introduce tracer particles, or otherwise increase the resolution in filamentary regions, a "twin fluid," an alternative labeling for a material of the same type, is introduced to capture under-resolved regions. A new "conglomeration algorithm" is introduced that determines whether a computational cell's advective preimage contains a filament, and therefore the single material cut by the filament needs to be partitioned into two materials. By introducing the "twin fluid," the reconstruction step is posed as three materials separated by two interfaces, rather than two materials separated by a single interface. This method automatically preserves volume exactly since the volume preserving property of the multimaterial Moment-Of-Fluid recontruction algorithm[20] carries over to the "filament capturing" algorithm. In contrast to adaptive mesh approaches, this method is asymptotically constant in spatial complexity as the width of the filament goes to zero.

Remarks:

- The polygonal area mapping method[89] has similarities with this approach in that all polygonal regions in the advective preimage are investigated in determining the complexity of the polygonal structure to be stored in a computational cell. This method is different from [89] in how the preimage information is compressed in order to prevent an exponential growth of polygonal information per cell, without the need for "ear-clipping" to eliminate surface nodes. Also this method has the capability to capture the creation of filaments for all advected materials instead of just one.

- If one explicitly tracked a deforming boundary using connected markers (see e.g. [81]), then filamentary regions are trivially preserved at the expense of one having to introduce extra logic in order to modify the connectivity of markers when opposite sides of a material become "close." Recent grid based front tracking methods[72, 73, 70] automatically reconnect markers when interfaces are about to merge, but the criteria for interface merging is implicitly

dependent on the underlying grid. The resolution of the underlying grid determines the thickness of resolvable filaments. In this approach, the adjacency condition (see Section 3.3.2) for analyzing material volumes in the advective preimage determines connectivity; in contrast to front tracking approaches, the parameters for determining whether interfaces touch depends on floating point precision, instead of a grid dependent or a user dependent parameter.

In this new work, new methods for "asymptotically-preserving" multiphase flow are presented. Aspects of the recent multimaterial, semi-Lagrangian developments presented by Galera *et. al.* [28, 27] are hybridized with the asymptotic preserving developments recently introduced by Kwatra *et. al.* [44] in order to produce a novel method that simultaneously has multimaterial/multiphase capability, accurately captures shocks, tracks material discontinuities without the aid of Riemann solvers or mixed material pressure formulations, and performs robustly in compressible and nearly incompressible flow regimes. Additionally, the procedure to identify and capture the formation of filaments, in an asymptotically-constant in cost manner, is is developed using the Moment-of-Fluid interface reconstruction technique. The dissertation is organized as follows: Chapter 2 defines the governing equations; Chapter 3 presents the Moment-of-Fluid interface reconstruction, Cell Integrated Semi-Lagrangian (CISL) advection scheme, and the novel Moment-of-Fluid based filament capturing method; Chapter 4 discusses the fluid algorithm and numerical discretization of Euler's equations; and Chapter 5 presents various benchmark tests and novel numerical results.

CHAPTER 2

FLUID EQUATIONS

The equations governing the behavior of inviscid fluid flow is derived in this chapter. This set of equations, known as Euler's equations, dictates the flow of mass, momentum, and energy in a system. Consider some arbitrary closed volume defining some finite region in a flow field. This volume is known as a *control volume*, with the volume denoted as $\mathcal{V}$ and the surface denoted as $\mathcal{S}$. We can consider two paradigms of flow: the volume is *fixed* in space, with fluid flowing in and out of the region; the volume is *moving* with the flow, tracking the motion of a given parcel of fluid, with no particles entering or exiting the control volume [5]. See Figure 2.1. Now, consider some

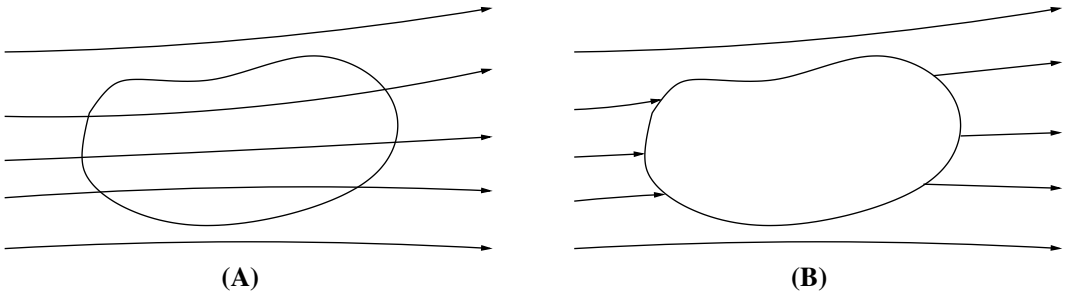

Figure 2.1: Two paradigms of fluid flow. (A) Fluid flows in and out of the control volume. (B) A parcel of fluid moves with the fluid flow.

infinitesimally small fluid element in the flow, with volume $d\mathcal{V}$. Let this element be infinitesimally small in the sense of differential calculus, but large enough that the continuum model for fluid dynamics still holds with in the element. As with the control volume, this infinitesimal element may be fixed in space or moving with the fluid flow, with instantaneous velocity $\boldsymbol{U}$ along a fluid streamline.

2.1 Continuity Equation

The continuity equation enforces the conservation of momentum, the principle that mass can be neither created nor destroyed. We will apply this principle on a fixed control volume in a flow

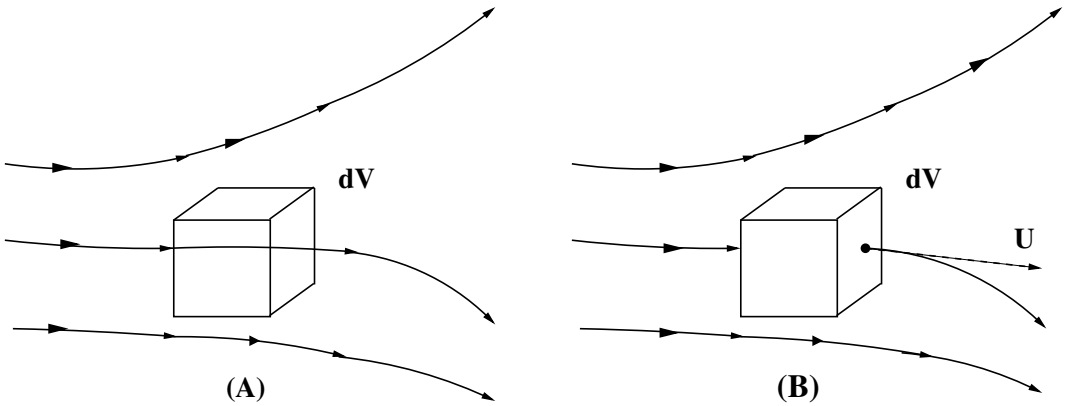

Figure 2.2: An infinitesimal element $d\mathcal{V}$ in a fluid flow, with streamlines shown as a solid line.

field (Figure 2.3). The volume $\mathcal{V}$ is the finite region contained within a closed surface S. Let $\boldsymbol{B}$ be some point on on the surface S. Define dS to be an elemental area around $\boldsymbol{B}$, with an outward pointing unit normal vector $\widehat{n}$. Define $\boldsymbol{U} = (u, v, w)^T$ and ρ to be the local velocity and density at point $\boldsymbol{B}$. Let θ be the angle between then normal vector to the surface and the velocity at the point $\boldsymbol{B}$. Then the mass flow $\dot{m}$ through the area dS can be calculated as

$$\dot{m} = \rho(|\boldsymbol{U}| \cos\theta) dS = \rho \boldsymbol{U} \cdot \widehat{n}\ dS. \tag{2.1}$$

The total mass flow into the control volume is the sum of elemental mass flows over the entire surface S. Note that the negative sign appearing in (2.2) is due to the fact that we are calculating flow *into* the control volume, and the vector $\widehat{n}$ is the *outward* pointing unit normal.

$$-\iint_S \rho \boldsymbol{U} \cdot \widehat{n}\ dS \tag{2.2}$$

Next, note that the mass in an infinitesimal volume region $d\mathcal{V}$ is equal to $\rho d\mathcal{V}$. Then, the total mass in the control region $\mathcal{V}$ is the sum over all infinitesimal regions in the control region,

$$\iiint_{\mathcal{V}} \rho\ d\mathcal{V}, \tag{2.3}$$

and the time rate of change of mass in the control volume is

$$\frac{\partial}{\partial t} \iiint_{\mathcal{V}} \rho\ d\mathcal{V}. \tag{2.4}$$

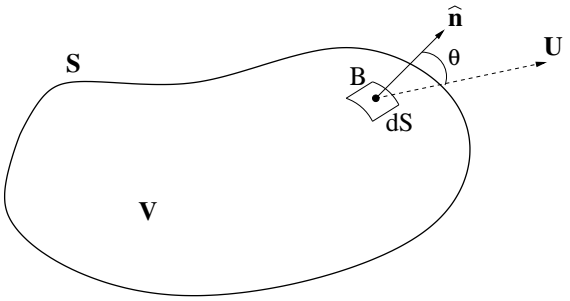

Figure 2.3: A control volume $\mathcal{V}$ and surface S. The outward normal $\widehat{n}$ and instantaneous velocity $\boldsymbol{U}$ are show at point $\boldsymbol{B}$, along with elemental area dS.

Let us now define some useful integral identities. Let $\boldsymbol{A}$ be some vector-valued function and Φ be some scalar function.

$$\iint_S \boldsymbol{A} \cdot \widehat{n}\, dS = \iiint_{\mathcal{V}} (\nabla \cdot \boldsymbol{A})\, d\mathcal{V} \tag{2.5}$$

$$\iint_S \Phi \widehat{n}\, dS = \iiint_{\mathcal{V}} (\nabla \Phi)\, d\mathcal{V} \tag{2.6}$$

Then, by the principle of conservation of mass, we have that the time rate of change of mass in the control volume must be equal to the total inflow of mass over the control surface. Thus,

$$\frac{\partial}{\partial t} \iiint_{\mathcal{V}} \rho\, d\mathcal{V} = - \iint_S \rho \boldsymbol{U} \cdot \widehat{n}\, dS. \tag{2.7}$$

The right-hand side of (2.7) can be rewritten, using (2.5), to obtain

$$\iint_S \rho \boldsymbol{U} \cdot \widehat{n}\, dS = - \iiint_{\mathcal{V}} \nabla \cdot (\rho \boldsymbol{U})\,. \tag{2.8}$$

The time derivative on the left-hand side of (2.7) can be interchanged with the integral, due to the fact that the control volume is fixed in time, i.e. the domain of the integral is *not* moving with the flow. So, we can write the expression for conservation of mass in terms of a volume integral,

$$\iiint_{\mathcal{V}} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}) \right)\, d\mathcal{V} = 0. \tag{2.9}$$

This expression is the integral form of the mass conservation equation. We will now note that the choice of the control volume $\mathcal{V}$ was arbitrary. For (2.9) to hold over *all* arbitrary control volumes,

it must be true that the integrand is identically zero everywhere. Let us first suppose that this is not true. This implies that there is some region $\mathcal{V}^+ \in \mathcal{V}$ such that the integrand, $\mathcal{I}(\boldsymbol{x})$, is strictly positive, and some region $\mathcal{V}^-$ such that the integrand is non-positive, where the integral over $\mathcal{V}^- \in \mathcal{V}$ exactly cancels the integral over $\mathcal{V}^+$. Here, the regions $\mathcal{V}^+, \mathcal{V}^-$ are disjoint.

$$\begin{aligned} \mathcal{V}^+ \cup \mathcal{V}^- &= \mathcal{V} \\ \mathcal{V}^+ \cap \mathcal{V}^- &= \emptyset \\ \iiint_{\mathcal{V}^-} \mathcal{I}(\boldsymbol{x})\, d\mathcal{V} &= -\iiint_{\mathcal{V}^+} \mathcal{I}(\boldsymbol{x})\, d\mathcal{V} \end{aligned} \tag{2.10}$$

But, the choice of the control volume was arbitrary. If we chose $\mathcal{V} = \mathcal{V}^+$, this would imply that the integral is non-zero. This leads to a contradiction, therefore the integrand must be identically zero.

Finally, we arrive at the differential expression for conservation of mass,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}) = 0. \tag{2.11}$$

2.2 Momentum Equation

The momentum equation is derived from the physical principle that the time rate of change of momentum in a body is equal to the net force on the body (2.12).

$$\frac{d}{dt}(m\boldsymbol{U}) = \boldsymbol{F} \tag{2.12}$$

Here, m, $\boldsymbol{U}$, and $\boldsymbol{F}$ represent the mass, velocity, and net force, respectively. For a constant mass, this expression takes the form of Newton's second law of motion, i.e. that force = mass $\times$ acceleration.

$$\boldsymbol{F} = m\frac{d\boldsymbol{U}}{dt} = m\boldsymbol{a} \tag{2.13}$$

It remains to express the forces that contribute to the net force:

1. **Body Forces:** Forces that perform "action at a distance" [5] on the fluid inside the control volume $\mathcal{V}$. This includes gravitational and electromagnetic forces. For this work, only gravitational force is considered. Let $\boldsymbol{f}$ be the force per unit mass on the fluid in the volume.

Consider some infinitesimal subvolume $d\mathcal{V} \in \mathcal{V}$. The body force on the infinitesimal volume is $\rho \boldsymbol{f}\ d\mathcal{V}$. Then, the total body force in $\mathcal{V}$ is calculated as

$$\boldsymbol{F}_b = \iiint_{\mathcal{V}} \rho \boldsymbol{f}\ d\mathcal{V}. \tag{2.14}$$

2. **Surface Forces:** Forces that act on the surface of a control volume include pressure and shear stress. However, because flow is assumed to be inviscid, we will consider only force due to pressure. Consider the surface element dS with outward unit normal $\widehat{n}$. Then, the force acting on the surface element area is

$$-p\widehat{n}\ dS. \tag{2.15}$$

The negative sign in (2.15) is due to the fact that we are computing force on the body from the surrounding fluid (i.e. the force int the inward direction $-\widehat{n}$). The total force on the surface of the body is then the integral of the pressure over the entire surface area,

$$\boldsymbol{F}_s = -\iint_S p\widehat{n}\ dS. \tag{2.16}$$

The net force is the sum of the body and surface forces, and it can then be expressed as

$$\boldsymbol{F} = \boldsymbol{F}_b + \boldsymbol{F}_s = \iiint_V \rho \boldsymbol{f}\ d\mathcal{V} - \iint_S p\widehat{n}\ dS. \tag{2.17}$$

We will now address the left-hand side of (2.12) by considering some fixed control volume $\mathcal{V}$, as in Figure 2.1 (A). Flow enters the control volume, bringing momentum, and exits, removing momentum. Let $\boldsymbol{A}_1$ denote the net rate of flow of momentum across S, the surface of the control volume $\mathcal{V}$. Across some elemental area dS with outward unit normal vector $\widehat{n}$, the mass flow is $\rho \boldsymbol{U} \cdot \widehat{n}\ dS$, so the momentum flow across the elemental area is then $(\rho \boldsymbol{U} \cdot \widehat{n}\ dS)\boldsymbol{U}$. Summing over all elemental volumes, we find the net rate of flow of momentum across the entire surface, S, to be

$$\boldsymbol{A}_1 = \iint_S (\rho \boldsymbol{U} \cdot \widehat{n}\ dS)\, \boldsymbol{U}. \tag{2.18}$$

In addition to the inflow/outflow of momentum through the control surface, the unsteady, transient effects of the flow field within the control volume must be considered as well. Let $\boldsymbol{A}_2$ denote these unsteady effects. For some infinitesimal volume within $\mathcal{V}$, the mass can be calculated

as $\rho\ d\mathcal{V}$. This elemental volume has momentum $(\rho\ d\mathcal{V})\boldsymbol{U}$. Summing over the entire control volume, we have the the total momentum in $\mathcal{V}$ is equal to

$$\iiint_{\mathcal{V}} \rho\boldsymbol{U}\ d\mathcal{V}. \tag{2.19}$$

So, the time rate of change of momentum in $\mathcal{V}$ due to the unsteady fluctuations in the local flow is

$$\boldsymbol{A}_2 = \frac{\partial}{\partial t}\iiint_{\mathcal{V}} \rho\boldsymbol{U}\ d\mathcal{V} = \iiint_{\mathcal{V}} \frac{\partial\left(\rho\boldsymbol{U}\right)}{\partial t}\ d\mathcal{V}. \tag{2.20}$$

Again, because the control volume $\mathcal{V}$ is assumed to be fixed, we are able to interchange the time derivative with the spatial integral in (2.20). Thus, we can describe the total instantaneous rate of change of momentum of the fluid as it flows through the control volume as

$$\frac{d}{dt}\left(m\boldsymbol{U}\right) = \boldsymbol{A}_1 + \boldsymbol{A}_2 = \iint_S (\rho\boldsymbol{U}\cdot\widehat{n}\ dS)\boldsymbol{U} + \iiint_{\mathcal{V}} \frac{\partial(\rho\boldsymbol{U})}{\partial t}. \tag{2.21}$$

Setting (2.17) equal to (2.21), (i.e. $\boldsymbol{F} = \frac{d}{dt}(m\boldsymbol{U})$), we obtain the integral form of the momentum equation

$$\iiint_{\mathcal{V}} \frac{\partial(\rho\boldsymbol{U})}{\partial t} + \iint_S (\rho\boldsymbol{U}\cdot\widehat{n}\ dS)\boldsymbol{U} = \iiint_{\mathcal{V}} \rho\boldsymbol{f}\ d\mathcal{V} - \iint_S p\widehat{n}\ dS. \tag{2.22}$$

To write (4.10) in conservation form, we will rewrite the surface integral on the right-hand side using the identity (2.6). Now, the vector equation is written in terms of its components to more easily apply Green's theorem, with $\boldsymbol{U} = \begin{pmatrix} u & v \end{pmatrix}^T$, $\boldsymbol{f} = \begin{pmatrix} f_x & f_y \end{pmatrix}^T$, and $\nabla p = \begin{pmatrix} \frac{\partial p}{\partial x} & \frac{\partial p}{\partial y} \end{pmatrix}^T$.

$$\iiint_{\mathcal{V}} \frac{\partial(\rho u)}{\partial t} + \iint_S (\rho\boldsymbol{U}\cdot\widehat{n}\ dS)u = \iiint_{\mathcal{V}} \left(\rho f_x\ -\frac{\partial p}{\partial x}\right)\ d\mathcal{V} \tag{2.23}$$

The surface integral is rewritten as a volume integral using the identity (2.5) to cast (2.23) as the integral over the control volume.

$$\iiint_{\mathcal{V}} \left(\frac{\partial(\rho u)}{\partial t} + \nabla\cdot(\rho u\boldsymbol{U}) - \rho f_x\ + \frac{\partial p}{\partial x}\right)\ d\mathcal{V} = 0 \tag{2.24}$$

Again, assert that because the control volume is arbitrary, the integrand must be identically zero. By treating the y-component of momentum in a similar fashion, we obtain the differential form of the expressions for conservation of momentum (2.25).

$$\begin{aligned} \frac{\partial(\rho u)}{\partial t} + \nabla\cdot(\rho u\boldsymbol{U}) &= -\frac{\partial p}{\partial x} + \rho f_x \\ \frac{\partial(\rho v)}{\partial t} + \nabla\cdot(\rho v\boldsymbol{U}) &= -\frac{\partial p}{\partial y} + \rho f_y \end{aligned} \tag{2.25}$$

2.3 Energy Equation

For the study of incompressible flow, it is necessary to model the conservation of mass and momentum to capture the mechanics of the flow. However, for the study of compressible flow, it is necessary to model the thermodynamics of the system. The energy equation is derived from the principle that energy is neither created nor destroyed, it simply changes form, as stated in the first law of thermodynamics. Consider some fixed control volume $\mathcal{V}$. Let

$$\begin{aligned} B_1 &= \text{rate of heat added to the fluid in } \mathcal{V} \text{ from the surroundings} \\ &\quad \text{(i.e. source terms)} \\ B_2 &= \text{rate of work done on the fluid inside the control volume} \\ B_3 &= \text{rate of change of energy of the fluid as it flows through } \mathcal{V}. \end{aligned}$$

The first law of thermodynamics then states

$$B_1 + B_2 = B_3. \tag{2.26}$$

Let q represent the heat added per unit mass, and $\dot{q}$ represent the time rate of change of the specific heat. Again, elemental mass for some infinitesimal region is calculated as $\rho\ \mathcal{V}$, so the B_1 term for the control volume is calculated as

$$B_1 = \iiint_{\mathcal{V}} \dot{q}\rho\ d\mathcal{V}. \tag{2.27}$$

Now, we will address the work done on the fluid. Let $\boldsymbol{F}$ denote some force exerted on the material over some displacement vector $d\boldsymbol{X}$. Let $\boldsymbol{X}^n$ denote a particle's position at time t^n and $\boldsymbol{X}^{n+1}$ denote the particle's position at time $t^n + \Delta t = t^{n+1}$. Then, the work done on the particle can be calculated as $\boldsymbol{F} \cdot \Delta \boldsymbol{x}$, where $\Delta \boldsymbol{X} = \boldsymbol{X}^{n+1} - \boldsymbol{X}^n$. Thus, the rate of work is

$$\boldsymbol{F} \cdot \frac{d\boldsymbol{X}}{dt} = \boldsymbol{F} \cdot \boldsymbol{U}. \tag{2.28}$$

Considering some body force per unit mass $\boldsymbol{f}$, the force exerted on the entire control volume can then be calculated.

$$\begin{Bmatrix} \text{Rate of work on fluid in } \mathcal{V} \\ \text{due to a body force on } S \end{Bmatrix} = \iiint_{\mathcal{V}} \rho \boldsymbol{f} \cdot \boldsymbol{U}\ d\mathcal{V} \tag{2.29}$$

Now, consider work done due to pressure forces on S, the bounding surface of $\mathcal{V}$. Let dS and $\widehat{n}$ be some elemental area on S and the outward unit normal vector. The force due to pressure is

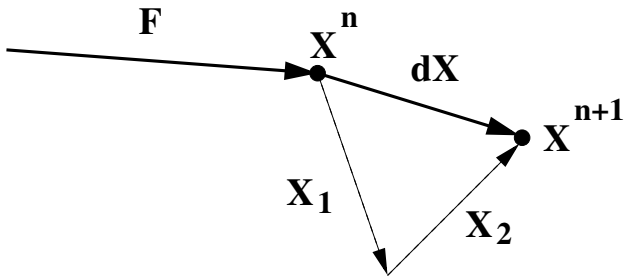

Figure 2.4: A force $\boldsymbol{F}$ moves a particle from $\boldsymbol{X}^n$ to $\boldsymbol{X}^{n+1}$, for a total displacement of $\Delta\boldsymbol{X}$, while traveling along intermediate paths X_1 and X_2 (dotted).

calculated as the force per unit area (pressure) multiplied by the area (dS) in the direction from the surface into the control volume ($-\widehat{n}$). By the same argument as before, the rate of work done by pressure on the control volume is then equal to the integral of the rate of work integrated over the surface.

$$\begin{cases}\text{Rate of work on fluid in } \mathcal{V} \\ \text{due to pressure on S}\end{cases} = -\iint_S p\boldsymbol{U}\cdot\widehat{n}\ dS \tag{2.30}$$

Thus, the total rate of work done on fluid in $\mathcal{V}$ by all forces on the control volume is

$$B_2 = -\iint_S p\boldsymbol{U}\cdot\widehat{n}\ dS + \iiint_{\mathcal{V}} \rho\boldsymbol{f}\cdot\boldsymbol{U}\ d\mathcal{V}. \tag{2.31}$$

Finally, we must compute the rate of change of energy in $\mathcal{V}$ as it flows through the control volume. Note that the total specific energy is the sum of the internal energy and the kinetic energy per unit mass.

$$E = e_{int} + \frac{||\boldsymbol{U}||^2}{2} \tag{2.32}$$

Integrating the elemental energy $\rho\left(e_{int} + \frac{||\boldsymbol{u}||^2}{2}\right)\ d\mathcal{V}$ over the entire control volume, we can calculate the total energy in $\mathcal{V}$ as

$$\iiint_{\mathcal{V}} \rho E\ d\mathcal{V} = \iiint_{\mathcal{V}} \rho\left(e_{int} + \frac{||\boldsymbol{U}||^2}{2}\right)\ d\mathcal{V}. \tag{2.33}$$

The rate of change of total energy in $\mathcal{V}$ is calculated by taking the time derivative of (2.33). By the typical argument, we can exchange the time derivative with the spatial integral, because the

control volume is fixed in time. This accounts for change of energy in $\mathcal{V}$ due to flow variations and fluctuations of fluid flow within the control volume.

$$\begin{cases}\text{Rate of change of energy in } \mathcal{V} \\ \text{due to transient variations of flow variables}\end{cases} = \iiint_{\mathcal{V}} \frac{\partial(\rho E)}{\partial t}\, d\mathcal{V}. \tag{2.34}$$

Finally, we must calculate the net rate of flow of energy across the surface of the control volume. The flow of total energy across some elemental area dS with outward unit normal $\widehat{n}$ is

$$\rho\left(e_{int} + \frac{||\boldsymbol{U}||^2}{2}\right)\boldsymbol{U}\cdot\widehat{n}\; dS = \rho E\; \boldsymbol{U}\cdot\widehat{n}\; dS. \tag{2.35}$$

Note that if (2.35) is positive, it denotes outflow, while a negative value denotes inflow, as $\widehat{n}$ points outward from the control volume. Integrating (2.35) over the entire surface S gives the rate of change of total energy in $\mathcal{V}$ due to fluid flow across the surface.

$$\begin{cases}\text{Net flow of total energy} \\ \text{across surface } S\end{cases} = \iint_{S} \rho E\; \boldsymbol{U}\cdot\widehat{n}\; dS \tag{2.36}$$

Thus, we can write the total change of energy in $\mathcal{V}$ as fluid flows through the control volume.

$$B_3 = \iiint_{\mathcal{V}} \frac{\partial(\rho E)}{\partial t}\, d\mathcal{V} + \iint_{S} \rho E\; \boldsymbol{U}\cdot\widehat{n}\; dS \tag{2.37}$$

Finally, we may apply the conservation of total energy in the control volume (2.26) by substituting in the expressions (2.27, 2.31, 2.37).

$$\iiint_{\mathcal{V}} \frac{\partial(\rho E)}{\partial t}\, d\mathcal{V} + \iint_{S} \rho E\; \boldsymbol{U}\cdot\widehat{n}\; dS = \iiint_{\mathcal{V}} \dot{q}\rho\; d\mathcal{V} - \iint_{S} p\boldsymbol{U}\cdot\widehat{n}\; dS + \iiint_{\mathcal{V}} \rho\boldsymbol{f}\cdot\boldsymbol{U}\; d\mathcal{V} \tag{2.38}$$

The surface integral terms on the left- and right-hand sides of (2.38) can be rewritten as volume integrals using the identity (2.5). This allows us to write the integral form of the expression for the conservation of energy as a single volume integral.

$$\iiint_{\mathcal{V}} \left(\frac{\partial(\rho E)}{\partial t} + \nabla\cdot(\rho E\boldsymbol{U}) - \dot{q}\rho - \rho\boldsymbol{f}\cdot\boldsymbol{U} + \nabla\cdot(p\boldsymbol{U})\right) d\mathcal{V} = 0 \tag{2.39}$$

By the typical argument, because the control volume $\mathcal{V}$ was chosen to be arbitrary, the only way for (2.39) at all points is for the integrand to be identically zero at all points. Thus, we finally arrive at the differential expression for conservation of energy.

$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot(\rho E\boldsymbol{U}) = \dot{q}\rho + \rho\boldsymbol{f}\cdot\boldsymbol{U} - \nabla\cdot(p\boldsymbol{U}) \tag{2.40}$$

2.4 Thermodynamic Relations

A gas is a collection of particles (molecules, atoms, ions, etc.) moving in more or less random motion. The electronic structure of these particles causes interaction due to intermolecular forces. By definition, a *perfect gas* is one in which these intermolecular forces are neglected. For this section, we will only consider a perfect gas (as opposed to a *real gas*, for which intermolecular forces are accounted) [5]. Ignoring intermolecular forces, the equation of state for a perfect gas, originally synthesized from experimental work, can be derived from the theory of modern statistical mechanics. The empirical relation derived from experiment was

$$p\nu = RT, \tag{2.41}$$

where p is pressure, ν is the specific volume, R is the universal gas constant, and T is temperature. Using the definition of specific volume $\nu = 1/\rho$, (2.41) can be rewritten as

$$p = \rho RT. \tag{2.42}$$

The specific energy of gas (as discussed in Section 2.3) is defined as

$$E = \left(e_{int} + \frac{||\boldsymbol{U}||^2}{2} \right),$$

the sum of the specific internal energy and the kinetic energy per unit of mass. The source of internal energy is motion on the microscopic level, including rotational and vibrational energy. Summed over all molecules in the material, these energies constitute the internal energy of a fluid. The kinetic energy per unit mass ($||\boldsymbol{U}||^2/2$) is due to bulk motion of the fluid at the macroscopic level. In simulations of compressible or high-speed flow, changes in local velocity equate to changes in local kinetic energy which may be non-negligible. So, capturing the thermodynamics of the system is necessary for the fidelity of the simulation [5, 7, 18]. For the sake of notation, let e denote the internal energy, omitting the subscript. Additionally, define enthalpy per unit mass as

$$h = e + p\nu. \tag{2.43}$$

Then we can write specific energy and enthalpy as functions of the other thermodynamic state variables,

$$\begin{aligned} e &= e(T, \nu) \\ h &= h(T, p). \end{aligned} \tag{2.44}$$

If intermolecular forces are ignored and the gas is not chemically reactive, then the material is deemed a *thermally perfect gas*, where internal energy and enthalpy, as well as specific heats at constant volume and pressure (c_ν and c_p respectively), are functions of temperature only:

$$\begin{aligned} e &= e(T) \\ h &= h(T) \\ de &= c_\nu \, dT \\ dh &= c_p \, dT \end{aligned} \tag{2.45}$$

If the specific heat indices c_ν and c_p are constant, then the system is said to be a calorically perfect gas, with

$$\begin{aligned} e &= c_v T \\ h &= c_p T. \end{aligned} \tag{2.46}$$

The assumption of a perfect gas for many compressible applications, in the regime of standard temperature and pressure, has been experimentally observed to differ from measured values by less than 1 percent [5]. It is worth noting that in equilibrium thermodynamics (i.e. in a system with no gradients in velocity, pressure, temperature, and material concentration), any thermodynamic state variable can be determined using any other two state variables.

The equation of state for pressure (as a function of density and internal energy) of an calorically perfect or thermally perfect gas will now be derived. Substituting (2.45) and (2.41) into (2.43) and differentiating with respect to temperature, we recover

$$c_p - c_\nu = R. \tag{2.47}$$

The expression for R is substituted into (2.42), and expressing energy in terms of energy in terms of temperature from (2.46), we recover

$$p = \frac{(c_p - c_\nu)}{c_\nu} \rho e. \tag{2.48}$$

Defining γ as the ratio of specific heat at constant pressure and specific volume, we arrive at the classical equation for the pressure of an ideal gas (2.49).

$$\begin{aligned} p &= (\gamma - 1)\rho e \\ \gamma &= \frac{c_p}{c_\nu} \end{aligned} \tag{2.49}$$

CHAPTER 3

MATERIAL INTERFACE RECONSTRUCTION AND TRANSPORT

3.1 Piecewise Linear Interface Calculation

A method that utilizes a piecewise-linear representation of an interface or material boundary is known as Piecewise Linear Interface Calculation (PLIC) method. Methods such as the Level Set, Volume-of-Fluid, and Moment-of-Fluid Methods all fall under the category of PLIC methods. The Level Set method uses a signed distance function from the material interface to represent a material sub-region, where level set function may be positive in the material sub-region, negative in the other material sub-region, and zero on the interface. This method has seen wide use in practice [16, 43, 59, 56, 13], but has the drawback that it does not necessarily conserve volume in deformational flows, leading to mass loss.

Volume-of-Fluid methods correct this problem by posing the interface reconstruction problem in each cell such that volume of the material in the cell is exactly captured, while the orientation of the interface is determined such that volume error in surrounding cells is minimized. This has the drawback that the reconstruction of an interface depends on data that is not local to the cell, making it more expensive and less readily applicable to meshes in 3D or unstructured meshes. Additionally, in the case that more than two materials occupy a cell, the VOF method relies on the "onion skin" model, which assumes that all previously captured material lies behind the next reconstructed interface (i.e. in the negative outward normal direction). VOF does not capture "triple point" or "T-junction" material configurations. For a comprehensive review of VOF methods, see [10]. The Volume-of-Fluid Method and its extension, the Coupled Level Set/Volume-of-Fluid (CLSVOF) Method, enjoy wide use in the community for numerically modeling evolving material interfaces [14, 63, 79, 60, 46].

The Moment-of-Fluid method [20], used in this work, is also a volume-conserving interface reconstruction technique, though it has several advantages over the Volume-of-Fluid method. It uses only data local to a cell to perform interface reconstruction, which allows it to capture sharp

corners better than VOF. Adaptive Mesh Refinement (AMR) on structured grids can also be applied in the context of the Moment-of-Fluid method to apply refinement in regions of high interface curvature or irregularity [3]. Use of local data makes MOF applicable on unstructured or chimera grids. Additionally, it is capable of capturing complex material configurations, such as triple-points and filaments [2].

3.1.1 Moment-of-Fluid Interface Reconstruction

The Moment-of-Fluid (MOF) method can be seen as an extension of the Volume-of-Fluid (VOF) method. Rather than finding the optimal interface reconstruction using only information about volume, or the zeroth "moment," MOF finds the optimal interface reconstruction using information about volume and centroid information, the first "moment." For each cell Ω_i, given the volume and first moment of each material subregion Ω_i^m, the Moment-of-Fluid interface reconstruction method prescribes the optimal interface as the linear segment that exactly captures the material volume and minimizes error in the material centroid.

Define a material interface $\Gamma(\hat{n}, b)$ as the zero level set of a function φ, as in (3.1), where $\hat{n}$ is the outward unit normal vector to the interface, relative to the reconstructed material.

$$\Gamma(\hat{n}, b) = \{\boldsymbol{x} | \varphi = \hat{n} \cdot \boldsymbol{x} + b = 0\} \tag{3.1}$$

This interface reconstruction technique can be posed as a constrained optimization problem, in that one must simultaneously solve for $\hat{n}$ and b such that (3.2) is satisfied. Here, let F_{ref}^m be the input, or reference, volume fraction of the given material in the cell, and let F_{act}^m be the actual volume of the material in the cell given the reconstruction $\Gamma(\hat{n}, b)$.

$$\begin{cases} |F_{act}(\hat{n}, b) - F_{ref}| = 0 \\ \underset{(\hat{n},b)}{\arg\min} \, |\boldsymbol{x}_{act}(\hat{n}, b) - \boldsymbol{x}_{ref}| \end{cases} \tag{3.2}$$

For a subregion of material m, $\Omega_i^m \subseteq \Omega_i$, let $H_i^m(\boldsymbol{x})$ be a Heaviside function, such that

$$H_i^m = \begin{cases} 1, & \boldsymbol{x} \in \Omega_i^m \\ 0, & \text{otherwise} \end{cases}.$$

Then, reference volume fraction F_{ref}^m and reference moment $\boldsymbol{x}_{ref}^m$ can be calculated as (3.3 - 3.4)

$$F_{ref}^m = \frac{\int_{\Omega_i} H_i^m(\boldsymbol{x}) \, d\boldsymbol{x}}{\int_{\Omega_i} d\boldsymbol{x}} \tag{3.3}$$

$$\boldsymbol{x}_{ref}^{m} = \frac{\int_{\Omega_i} \boldsymbol{x} H_i^m(\boldsymbol{x})\ d\boldsymbol{x}}{\int_{\Omega_i} d\boldsymbol{x}} \tag{3.4}$$

Let V_i be the volume of cell Ω_i. To limit numerical errors resulting from prohibitively small volume fractions, volume fractions below a cutoff tolerance ϵ are truncated to zero. Similarly, volume fractions greater than $1-\epsilon$ are truncated to one (3.5). For the 2D test problems presented, the truncation value $\epsilon = 10^{-10}$ is used. After truncation, volume fractions are normalized to sum to one (3.6).

$$F^m \leftarrow \begin{cases} 0, & F^m < \epsilon \\ F^m, & \epsilon \leq F^m \leq 1-\epsilon \\ 1, & F^m > 1-\epsilon \end{cases} \qquad m = 1, \ldots, M \tag{3.5}$$

$$\begin{aligned} F &= \sum_{m=1}^{M} F^m \\ F^m &\leftarrow \frac{F^m}{F} \end{aligned} \tag{3.6}$$

The normal vector is taken to be a function of θ, the angle of rotation between the vector $\hat{n}$ and the standard basis vector $\hat{e}_x = (1,0)^T$. This parametrization allows us to define $\hat{n}(\theta)$ as

$$\hat{n}(\theta) = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}. \tag{3.7}$$

In the case of $M = 2$, the reference material (i.e. the material for which the volume fraction and centroid are used in the MOF reconstruction) is chosen as the material with the largest volume fraction. The case for $M \geq 3$ is discussed in Section 3.1.2.

Define $\boldsymbol{x}_{comp}$ as the centroid of the complementary region to Ω_i^{ref} in cell Ω_i and $F_{comp} = 1 - F_{ref}$. Then, if we define the centroid of cell Ω_i as $\boldsymbol{x}_i$,

$$\boldsymbol{x}_i = \frac{\int_{\Omega_i} \boldsymbol{x}\ d\boldsymbol{x}}{\int_{\Omega_i} d\boldsymbol{x}} \tag{3.8}$$

then $\boldsymbol{x}_{comp}$ has the property

$$\boldsymbol{x}_i = F_{ref}\boldsymbol{x}_{ref} + F_{comp}\boldsymbol{x}_{comp}. \tag{3.9}$$

Note that the choice of angle θ for normal vector $\hat{n}$ implies a specific value for parameter b to satisfy exactness for the volume fraction. This defines the reconstructed centroid $\boldsymbol{x}_A$ and the

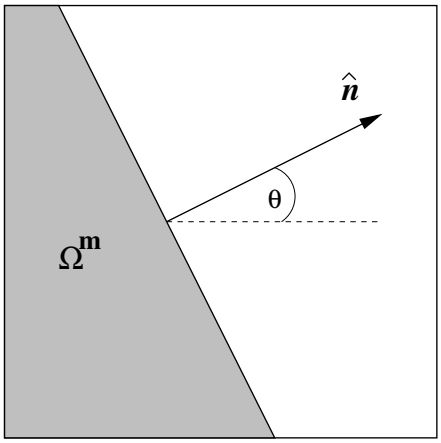

Figure 3.1: Given a region Ω^m of material m, the outward unit normal vector is defined by angle θ between $\hat{n}$ and $\hat{e}_x = (1,0)^T$.

reconstructed centroid of the complementary region $\widehat{\boldsymbol{x}}_A$. So, the total Moment-of-Fluid error can be written as a non-linear least-squares problem for the angle, as a function of $f(\theta)$. The minimization is thus performed for the interface that simultaneously minimizes error in the reference centroid and complementary centroid for the objective function (3.11), through a procedure known as the Symmetric Moment-of-Fluid [33]. This has been demonstrated to produce more accurate results and perform more robustly on unstructured grids than posing the minimization problem solely in terms of error of the reference centroid.

$$f(\theta) : \mathbb{R} \to \mathbb{R}^4, \quad f(\theta) = (\boldsymbol{x}_{ref} - \boldsymbol{x}_A(\theta), \boldsymbol{x}_{comp} - \widehat{\boldsymbol{x}}_A(\theta))^T \tag{3.10}$$

$$S(\theta) = ||\boldsymbol{x}_{ref} - \boldsymbol{x}_A(\theta)||_2^2 + ||\widehat{\boldsymbol{x}}_{ref} - \widehat{\boldsymbol{x}}_A(\theta)||_2^2 \tag{3.11}$$

The minimization problem is solved for θ by means of the Gauss-Newton method, which is given by the following algorithm.

1. Choose an initial angle θ_0, initialize $k = 0$, and initialize E_{MOF} to some arbitrarily large value. Set $tol = 10^{-8}\sqrt{V_i}$.
2. Find $b_k(\theta)$ such that (3.2) holds.
3. Compute the Moment-of-Fluid error E_{MOF}^{new}.

(a) If $E_{MOF}^{new} < E_{MOF}$ then set $E_{MOF} \leftarrow E_{MOF}^{new}$

4. Find reconstructed centroid $\boldsymbol{x}_{A,k}(\theta_k, b_k)$.

5. Find the Jacobian matrix J_k of $S(\theta)$ evaluated at (θ_k) and $S(\theta_k)$.

6. Stop if one of the following conditions is satisfied:

   - $||J_k^T \cdot f_k|| \leq tol \cdot 10^{-2}\sqrt{V_i}$
   - $||f_k|| < tol$
   - $k = 1$

   else continue

7. Solve the linear least squares problem: find $s_k \in \mathbb{R}$ such that the minimization problem

   $$\min_{s \in \mathbb{R}} ||J_k s_k + f_k||_2^2$$

   is satisfied.

8. Update the optimal angle $\theta_{k+1} = \theta_k + \widehat{s}_k$, with maximum angular change $d\theta^{MAX} = 0.3\ rad.$

   $$\widehat{s}_k = \min(s_k, d\theta^{MAX})$$

9. Set $k \leftarrow k + 1$.

Upon exit, compute the MOF reconstruction error (3.12) as the sum over all materials m of the norm of the difference between the material's reference (input) centroid and actual (reconstructed) centroid, multiplied by the material volume fraction.

$$E_{MOF} = \sum_{m=1}^{M} F^m ||\boldsymbol{x}_{m,ref} - \boldsymbol{x}_{m,act}|| \tag{3.12}$$

The Jacobian (3.13) of the objective function $S(\theta)$ is the change of the components of the reconstructed centroids with respect to the angle θ.

$$J(\theta) = \left(\frac{dx_A}{d\theta} \ \frac{dy_A}{d\theta} \ \frac{d\widehat{x}_A}{d\theta} \ \frac{d\widehat{y}_A}{d\theta} \right)^T \tag{3.13}$$

The quantities that appear in the Jacobian are discretized using central differencing, as in (3.14), for some small perturbation $\delta\theta$. For this work, I take $\delta\theta = 10^{-8}$.

$$\frac{df_i}{d\theta} \approx \frac{f_i(\theta + \delta\theta) - f_i(\theta - \delta\theta)}{2\delta\theta} \tag{3.14}$$

The optimization procedure described is not guaranteed to converge to the global minimum, as it is a gradient descent-based technique. A maximum angular change for each individual iteration $d\theta^{MAX} = 0.3\ rad$ is prescribed. This prevents overshoot if $||J(\theta)||^2$ large, but for a sufficiently poor initial guess, it is possible for the method to become caught in a local minimum. The minimization procedure can be made more aggressive by making the following change. If upon exit, the error E_{MOF} exceeds the tolerance, the initial angle θ_0 is rotated by a factor $d\theta^\star$ and iteration counter k is reset to zero. This aggressive restarting procedure can be repeated as desired, at the cost of making the interface reconstruction procedure more expensive. It should be noted that for a highly irregular interface, it is possible to locate the global minimum of (3.11) and still have a centroid error that exceeds the tolerance. For this work, a factor $d\theta^\star = \frac{\pi}{2}$ with a maximum of 3 restarts was used. A study of the number of iterations to convergence and optimality of the initial angle θ_0, for two methods of choosing an initial interface orientation, can be found in Section 5.4. Once the Moment-of-Fluid optimization procedure has completed, accept the interface reconstruction that yielded the *lowest* error, E_{MOF}, over all material moments

3.1.2 Multimaterial Moment-of-Fluid Method

The Multimaterial Moment-of-Fluid Method is the generalization of the two material Moment-of-Fluid algorithm, described in the previous section, to an arbitrary number of materials. Consider an ordered set of material centroids and volume fractions $\{(\boldsymbol{x}^m, F^m)\}_{m=1}^M$, where material $m = 1$ is reconstructed first, material $m = 2$ is reconstructed second, etc. After material m has been reconstructed, a region Ω^m in the cell Ω will be defined. After the reconstruction is performed, material m is denoted as "captured." The region Ω^m is then removed from the domain (here, the cell), and all further reconstructions are performed in the "uncaptured" region. Define the captured and uncaptured regions after the reconstruction of material m as

$$\Omega^{cap,m} = \bigcup_{\mu=1}^{m} \Omega^\mu \tag{3.15}$$

and

$$\Omega^{uncap,m} = \Omega \bigcap \left(\bigcup_{\mu=1}^{m} \Omega^\mu \right)^C \tag{3.16}$$

respectively. The super-script C in (3.16) denotes the complementary region of a set. Each step m of the Multimaterial MOF algorithm can then be posed as a reconstruction for two materials in

the reduced domain $\Omega^{uncap,m-1}$. Define the reference moment data as

$$(\boldsymbol{x}_{ref}, F_{ref}) = (\boldsymbol{x}^m, F^m)\,. \tag{3.17}$$

The complementary moment data can then be computed as

$$\begin{aligned} F_{comp} &= \frac{\sum_{\mu=m+1}^{M} V^\mu}{\sum_{\mu=m}^{M} V^\mu} \\ \boldsymbol{x}_{comp} &= \frac{\sum_{\mu=m+1}^{M} \boldsymbol{x}^\mu V^\mu}{\sum_{\mu=m}^{M} V^\mu}. \end{aligned} \tag{3.18}$$

By defining the reference and complementary moment data on the reduced, uncaptured region at each step, the MOF reconstruction algorithm for two materials can be applied at each step. After all materials are reconstructed, the Moment-of-Fluid error is calculated. If this error exceeds the tolerance, then the material reconstruction ordering may be non-optimal. Another material ordering may be then chosen. This process can be repeated until the error is acceptably small, or all material orderings are exhausted, at which point the reconstruction with the lowest error is accepted. As with the two material MOF procedure, for a sufficiently irregular interface, it is possible for the optimization routine to reach a global minimum and still have large error. Note that the Multimaterial MOF reconstruction is sensitive to material ordering. See Figure 3.2. A general rule for choosing the reconstruction order is to choose the first reconstructed material as the one with reference centroid farthest from the cell centroid [2]. Let us now write the Multimaterial Moment-of-Fluid reconstruction algorithm.

1. Choose a material ordering.
2. Mark all materials as uncaptured. Set $\Omega^{uncap,0} = \emptyset$. Initialize E_{MOF} to be some arbitrary large number. Set $m = 1$.
3. For $m = 1 \ldots M$, do the following.
   (a) Set $(\boldsymbol{x}_{ref}, F_{ref})$ as in (3.17) and $(\boldsymbol{x}_{comp}, F_{comp})$ as in (3.18).
   (b) Solve for the Moment-of-Fluid reconstruction on $\Omega^{uncap,m-1}$ with data $(\boldsymbol{x}_{ref}, F_{ref})$ and $(\boldsymbol{x}_{comp}, F_{comp})$.

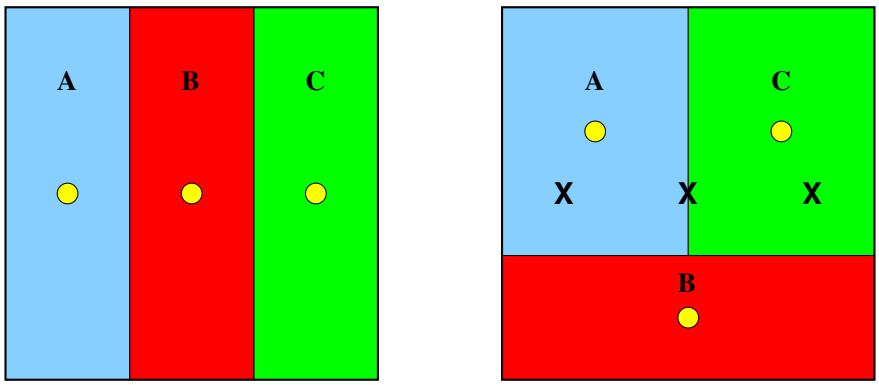

Figure 3.2: Materials A, B, C (blue, red, green respectively) with centroids as circles. If Material A or C is chosen to reconstruct first, this can be captured exactly (left). If Material B is chosen to reconstruct first, the reconstruction will be incorrect (shown right, with reference centroids as an 'X' and actual centroids circles).

(c) Mark material m as "captured." Update the upcaptured region.

(d) Go to Step (3).

4. Calculate error E_{MOF}. If error is less than the tolerance, exit. Otherwise, go to Step (1) with a new material ordering.

3.1.3 Calculating Initial Moment Data

To use the Moment-of-Fluid interface reconstruction procedure, it is necessary to calculate the volume and centroid of each material sub-region in each cell. For moving interface problems, volumes and centroids are calculated from the advected piecewise-linear interface computed at the previous time step. However, interface data must be initialized at the start of the simulation. The procedure for calculating this data on a triangular mesh will be discussed is this section.

Assume that the initial material interface Γ_m is defined as the zero contour of some level-set function, $\phi_m(\boldsymbol{x})$. For a cell Ω_i that is not cut by the interface, or a "pure" cell, the sign of $\phi_m(\boldsymbol{x})$ will be the same at all vertices of the cell.

$$\phi_m(\boldsymbol{x}) \begin{cases} < 0, & \boldsymbol{x} \in \text{Material } m \\ = 0, & \boldsymbol{x} \in \Gamma^m \\ > 0, & otherwise \end{cases}$$

For a cell that is cut by the interface, there exist vertices $\boldsymbol{P}_{r1}$ and $\boldsymbol{P}_{r2}$ such that

$$\phi_m(\boldsymbol{P}_{r1}) \cdot \phi_m(\boldsymbol{P}_{r2}) < 0. \tag{3.19}$$

A cut cell can be triangulated based on the value of the level-set function at the vertices. Assuming that ϕ_m varies linearly in the cell, the intersection of the interface with the cell can be found. There exist 6 cases for a cut cell.

1. $\phi_m(\boldsymbol{P}_r) < 0, \quad \phi_m(\boldsymbol{P}_{r+1}) \geq 0, \quad \phi_m(\boldsymbol{P}_{r+2}) \geq 0$
2. $\phi_m(\boldsymbol{P}_{r+1}) < 0, \quad \phi_m(\boldsymbol{P}_{r+2}) \geq 0, \quad \phi_m(\boldsymbol{P}_r) \geq 0$
3. $\phi_m(\boldsymbol{P}_{r+2}) < 0, \quad \phi_m(\boldsymbol{P}_r) \geq 0, \quad \phi_m(\boldsymbol{P}_{r+1}) \geq 0$
4. $\phi_m(\boldsymbol{P}_r) \geq 0, \quad \phi_m(\boldsymbol{P}_{r+1}) < 0, \quad \phi_m(\boldsymbol{P}_{r+2}) < 0$
5. $\phi_m(\boldsymbol{P}_{r+1}) \geq 0, \quad \phi_m(\boldsymbol{P}_{r+2}) < 0, \quad \phi_m(\boldsymbol{P}_r) < 0$
6. $\phi_m(\boldsymbol{P}_{r+2}) \geq 0, \quad \phi_m(\boldsymbol{P}_r) < 0, \quad \phi_m(\boldsymbol{P}_{r+1}) < 0$

It is clear that two distinct cases exist: there is one vertex such that the level-set function is negative and two vertices such that the level-set function is non-negative; or there is one vertex such that the level-set function is non-negative and two vertices such that the level-set function is negative. Denote s_1 and s_2

$$s_1 = \begin{cases} +1, & \text{only 1 non-negative signed vertex} \\ -1, & \text{otherwise} \end{cases} \tag{3.20}$$
$$s_2 = -s_1$$

and $\boldsymbol{P}_{r\star}$ as the vertex such that the level-set function evaluated at the other vertices has opposite sign (arbitrarily, consider $\phi = 0$ to be "positive"). Calculate the linear reconstruction $\widehat{\phi}$ of ϕ in the cell such that

$$\widehat{\phi}(\boldsymbol{P}_r) = \phi(\boldsymbol{P}_r), \quad r = 1, \ldots, 3$$

Let $\boldsymbol{v}_1$ denote the intersection of $\widehat{\phi} = 0$ with the edge spanning $\{\boldsymbol{P}_{r\star}, \boldsymbol{P}_{r\star+1}\}$, and let $\boldsymbol{v}_2$ denote the intersection of $\widehat{\phi} = 0$ with the edge spanning $\{\boldsymbol{P}_{r\star}, \boldsymbol{P}_{r\star+2}\}$. The sub-region with sign s_1 can be triangulated by forming a triangle with vertices $\boldsymbol{P}_{r\star}$, $\boldsymbol{v}_1$, and $\boldsymbol{v}_2$ The remaining sub-region with sign s_2 can be triangulated by forming two triangles: one with vertices $\boldsymbol{v}_1$, $\boldsymbol{P}_{r\star+1}$, and $\boldsymbol{P}_{r\star+2}$; and the other with vertices $\boldsymbol{v}_1$, $\boldsymbol{P}_{r\star+2}$, and $\boldsymbol{v}_2$ (see Figure 3.3). If the level-set function is zero at one or more of the vertices, at least one degenerate triangle will be generated by this procedure. For the pathological case such that the level-set function is zero at all vertices, the cell is considered to be pure.

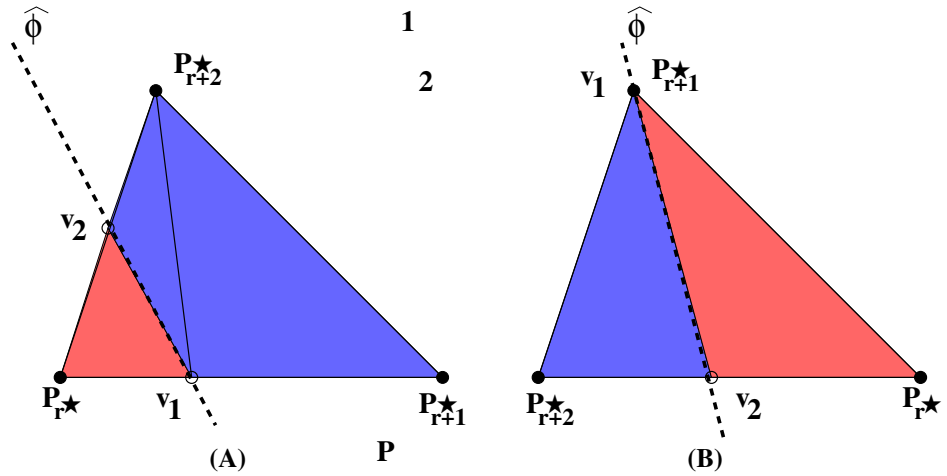

Figure 3.3: Region with sign s_1 is shown in red, s_2 region is shown in blue. (A) Triangular cell cut by linear approximation to ϕ (dotted). Level-set function at all vertices is non-zero. Vertex $\boldsymbol{P}_{r\star}$ is shown. Both regions are triangulated. (B) Triangular cell cut to produce two triangular sub-regions, i.e. level-set function is zero at exactly one vertex.

The volume of the material regions in each cell can be calculated. However, for accurate initial data, we wish to be able to calculate the volume and moment data to an arbitrary level of accuracy. The procedure to use a recursive triangulation procedure to achieve arbitrary accuracy in the calculation of moment data will be described. Consider a cut cell Ω_i, as defined in (3.19).

1. Define a maximum level of refinement $REFMAX$. Set the refinement counter $k = 0$. Set up a list of triangles T^m for each material.

2. Refine triangular cell Ω_i by forming four triangular sub-cells $\{\Omega_{i,j}\}_{j=1}^4$, generated by adding new vertices at the midpoint of each edge (see Fig. 3.4).

3. For each sub-cell $\Omega_{i,j}$, $j = 1, \ldots, 4$

   (a) Evaluate $\phi(\boldsymbol{x})$ at the vertices of $\Omega_{i,j}$.

   (b) If the sub-cell is cut and $k < REFMAX$, then set $k \leftarrow k+1$ and $\Omega_i \leftarrow \Omega_{i,j}$. Go to Step 1.

   (c) Otherwise, perform the triangulation procedure described and append triangulated material sub-regions to the appropriate list T^m.

4. Calculate the volume and centroid data for each set of material triangles T_m.

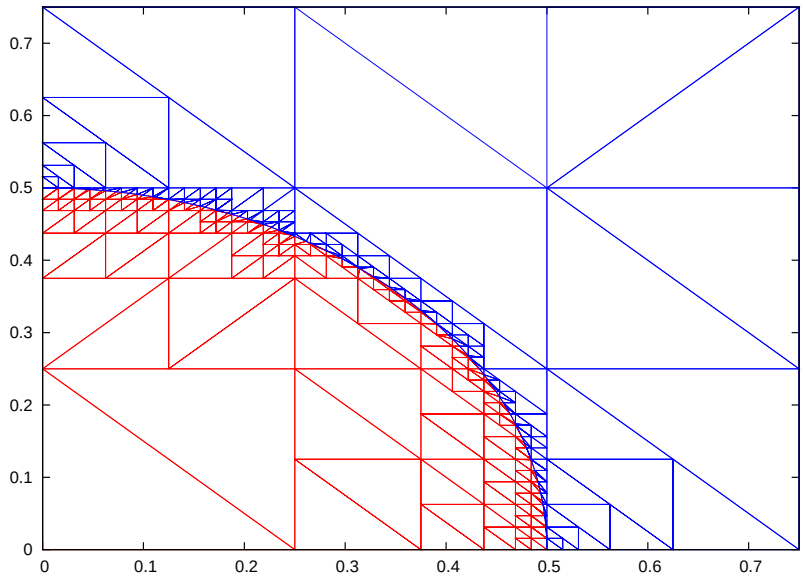

Figure 3.4: Triangulated mesh cut by the interface $\phi(x, y) = 1/2 - \sqrt{x^2 + y^2}$, with the recursive triangulation initialization procedure. Initial mesh resolution is $\Delta x = \Delta y = 0.25$, with 4 levels of refinement to compute initial moment data. Material such that $\phi \geq 0$ is shown in red, and $\phi < 0$ is shown in blue.

3.1.3.1 Moment Calculation in Cartesian Coordinates. Moment calculation in Cartesian coordinates is a straightforward task. Let vertices $\{\boldsymbol{x}_i\}_{i=1}^3$ define a triangle $\mathcal{T}$. Then triangle $\mathcal{T}$ has volume $V_{\mathcal{T}}$ and centroid $\boldsymbol{x}_{\mathcal{T}}$,

$$\begin{aligned} V_{\mathcal{T}} &= \frac{1}{2}|(\boldsymbol{x}_2 - \boldsymbol{x}_1) \times (\boldsymbol{x}_3 - \boldsymbol{x}_1)| \\ \boldsymbol{x}_{\mathcal{T}} &= \frac{\int_{\mathcal{T}} \boldsymbol{x}\ d\boldsymbol{x}}{\int_{\mathcal{T}} d\boldsymbol{x}} = \frac{1}{3}\sum_{i=1}^{3} \boldsymbol{x}_i. \end{aligned} \tag{3.21}$$

Now, define a polygonal region $\mathcal{P}$ as the union of N triangles $\mathcal{T}_i$, each with volume V_i and centroid $\boldsymbol{x}_{\mathcal{T}_i}$,

$$\mathcal{P} = \bigcup_{i=1}^{N} \mathcal{T}_i.$$

Then, the volume $V_{\mathcal{P}}$ and centroid $\boldsymbol{x}_{\mathcal{P}}$ of the triangulated polygonal region can be calculated as

$$V_{\mathcal{P}} = \sum_{i=1}^{N} V_i \tag{3.22}$$

$$\boldsymbol{x}_{\mathcal{P}} = \frac{\int_{\mathcal{P}} \boldsymbol{x}\ d\boldsymbol{x}}{\int_{\mathcal{P}} d\boldsymbol{x}} = \frac{\sum_{i=1}^{N} V_i \boldsymbol{x}_{\mathcal{T}_i}}{V_{\mathcal{P}}}, \tag{3.23}$$

using the fact from (3.21) that

$$\int_{\mathcal{T}} \boldsymbol{x}\ d\boldsymbol{x} = \boldsymbol{x}_{\mathcal{T}} V_{\mathcal{T}}. \tag{3.24}$$

3.1.4 Moment Calculation in Axisymmetric Coordinates

Volume calculation of a triangle in axisymmetric coordinates is equivalent to calculating the volume of revolution of a triangle about the axis of rotation. Again, let vertices $\{\boldsymbol{x}_j\}_{j=1}^{3}$ define a triangle $\mathcal{T}$. Compute the volume $V_{\mathcal{T}}$ as in (3.21). Calculate the radius $\widehat{r} = \frac{1}{3}\sum_{j=1}^{3} x_j$ of the centroid in Cartesian coordinates from the axis of revolution (here assumed to be the y-axis). Then, the volume $V_{\mathcal{T}}^{A}$ of the triangle in axisymmetric coordinates is

$$V_{\mathcal{T}}^{A} = \widehat{r} V_{\mathcal{T}}.$$

Here, the common factor of 2π in all moment calculations, resulting from integrating about the axis of revolution, has been omitted.

It is also necessary to calculate moment data in cylindrical coordinates for an arbitrary triangular element. This is accomplished by performing a change of variables, transforming the arbitrary triangular element Ω_i to some "unit" triangle Ω' (Fig. 3.5) and calculating the new integral on the domain of the unit triangle in terms of the new variables. The change of variables is computed with respect to a certain vertex $\boldsymbol{x}_i$ of the original triangular element (3.25).

$$\int_{\Omega} f(\boldsymbol{x})\ d\boldsymbol{x} = \in_{\Omega'} f(\boldsymbol{x}')\ d\boldsymbol{x}'$$

1. Compute the linear mapping from the original variable $\boldsymbol{x}$ to the variable on the unit triangle.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = A \begin{pmatrix} x - x_i \\ y - y_i \end{pmatrix} \tag{3.25}$$

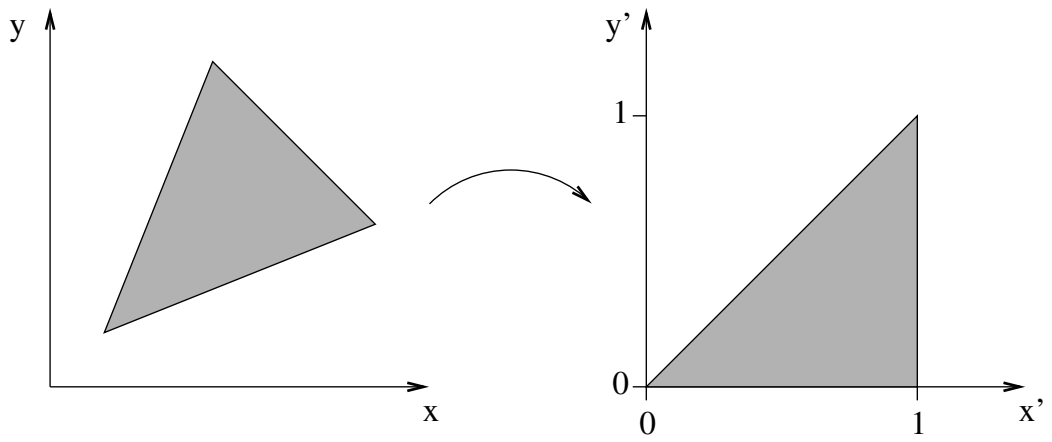

Figure 3.5: An arbitrary triangular element (left) is mapped to a unit triangle (right), to make moment calculation simpler in axisymmetric coordinates.

2. Compute the inverse mapping as a function of variables on the unit triangle.

$$\begin{aligned} x &= ex' + fy' + g \\ y &= qx' + ry' + s \end{aligned} \tag{3.26}$$

3. Compute the integral for the centroid on the unit triangle.

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \begin{pmatrix} \int_0^1 \int_{y'}^1 x(x',y')^2 \ J \ dx'dy' \\ \int_0^1 \int_{y'}^1 x(x',y')y(x',y') \ J \ dx'dy' \end{pmatrix} \tag{3.27}$$

Moment computation for some triangulated polygonal region proceeds as before. Given some polyon $\mathcal{P}^A$ in axisymmetric coordinates, where $\mathcal{P}^A = \bigcup_{i=1}^{N} \mathcal{T}_i^A$, each with volumes V_i^A and centroid $\boldsymbol{x}_{\mathcal{T}_i}$, then moments are calculated as

$$\begin{aligned} V_{\mathcal{P}}^A &= \sum_{i=1}^{M} V_i^A \\ \boldsymbol{x}_{\mathcal{P}} &= \frac{\int_{V_{\mathcal{P}}} \boldsymbol{x} \ x \ d\boldsymbol{x}}{\int_{V_{\mathcal{P}}} x \ d\boldsymbol{x}} = \frac{\sum_{i=1}^{N} \boldsymbol{x}_{\mathcal{T}_i} V_i^A}{V_{\mathcal{P}}^A}. \end{aligned} \tag{3.28}$$

3.1.5 A New Initial Guess for the Moment-of-Fluid Reconstruction Procedure

While the Moment-of-Fluid method has been used in a variety of contexts on structured and unstructured grids, little has been done in the way of determining an "optimal" initial angle. The

initial interface normal direction proposed in the original work [20] on MOF is

$$\widehat{n} = \frac{\boldsymbol{x}_i - \boldsymbol{x}_{ref}}{||\boldsymbol{x}_i - \boldsymbol{x}_{ref}||_2}. \tag{3.29}$$

A survey of the literature shows that this is the standard initial guess used. On rectangular meshes in Cartesian coordinates, this choice of an initial orientation has the favorable property that it is exact for linear interfaces. This property is not preserved for triangular meshes. As seen in Figure 3.6, even a linear interface, aligned with the mesh, is not accurately captured with this initial guess.

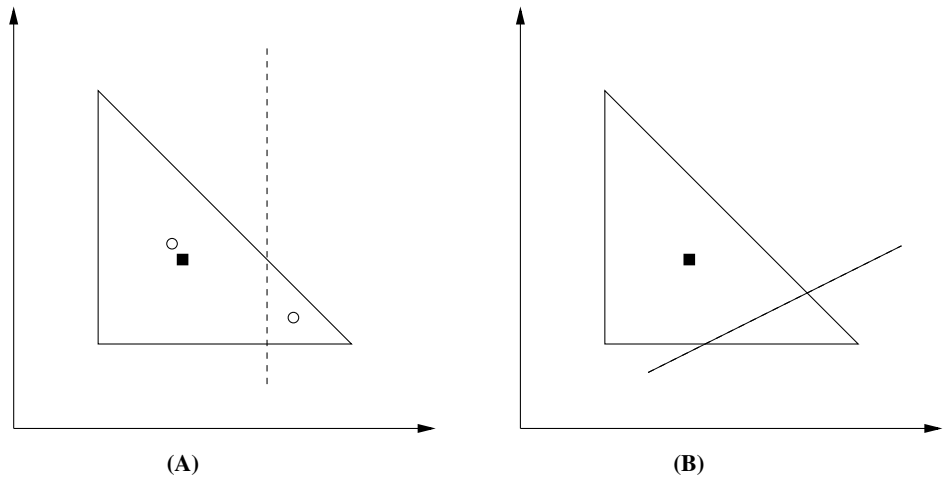

Figure 3.6: (A) Triangular cell cut by an interface (dotted). Cell centroid shown as a filled square. Reference and complementary centroids shown as circles. (B) Initial guess for interface orientation given by (3.29) for centroid configuration in (A). The method (3.29) is seen to fail for this straightforward case.

In this section, a new "optimal" initial angle is proposed on 2D simplicial meshes in Cartesian coordinates. This choice of initial angle is deemed "optimal" in that it shares the property that the initial guess is exact for a linear interface, while the computational cost of computing the initial guess is less than a single iteration of the Moment-of-Fluid optimization problem. As stated in Section 3.1.1, an accurate initial guess can eliminate the need for a more aggressive and costly optimization procedure.

First, note that there are only two ways to cut a triangular cell Ω_i with a linear interface to yield two sub-regions with non-zero volume (see Fig. 3.7).

1. The interface cuts the triangle to yield a triangular and trapezoidal sub-region.

2. The interface cuts the triangle to yield two triangular sub-regions.

Either method of cutting a triangle with a linear interface produces *at least* one triangular sub-region. All other ways of cutting a triangular cell with a linear interface yield a sub-region of zero volume for one of the materials. If the linear interface is viewed as the zero-contour of some level-set function $\Gamma(\boldsymbol{x})$, then the first method corresponds to the case when $|\Gamma(\boldsymbol{P}_r)| > 0$ for all vertices P_r of the cell. The second method corresponds to the case when $\Gamma(\boldsymbol{P}_r) = 0$ for *exactly one* vertex $\boldsymbol{P}_r$ of the cell. Define a "critical vertex" $\boldsymbol{P}_{r\star}$ of the cell cut by a linear interface $\Gamma(\boldsymbol{x})$ as a vertex such that $|\Gamma(\boldsymbol{P}_{r\star})| > 0$ and that $\boldsymbol{P}_{r\star}$ is a vertex of a triangular sub-region (see Fig. 3.7). Additionally,

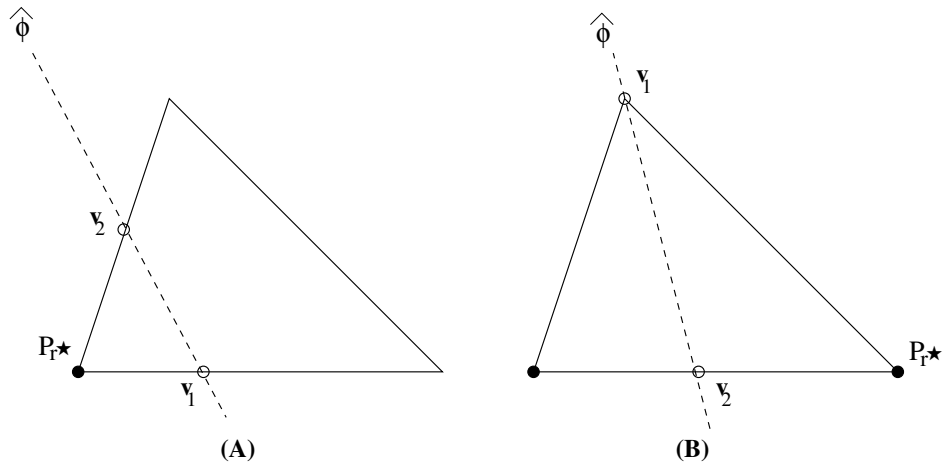

Figure 3.7: (A) Triangular cell cut by an interface $\widehat{\phi}$ (dotted). Cell is cut to produce a triangular and trapezoidal sub-region. (B) Triangular cell cut to produce two triangular sub-regions. Critical vertices marked with a filled circle. Intersection points relative to critical vertex $\boldsymbol{P}_{r\star}$ are marked $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.

note that in Cartesian coordinates, the centroid of a triangle with vertices $\{\boldsymbol{v}_i\}_{i=1}^3$ is

$$\boldsymbol{c} = \begin{pmatrix} c_x \\ c_y \end{pmatrix} = \frac{1}{3}\sum_{i=1}^{3} \boldsymbol{v}_i.$$

Next, note that any arbitrary interface $\widehat{\Gamma}$ that cuts a cell yields reference moment $\boldsymbol{x}_{ref}$ and complementary moment $\boldsymbol{x}_{comp}$. From [20], the polygonal partition of the cell is *uniquely* identified by the set of the material centroids. So, there exists some *unique* linear interface $\Gamma^\star(\hat{n}^\star, b^\star)$ that cuts cell Ω_i into sub-regions with centroids $\boldsymbol{x}_{ref}$ and $\boldsymbol{x}_{comp}$. Thus, if $\widehat{\Gamma}$ is a linear interface, then by uniqueness, $\widehat{\Gamma} = \Gamma^\star(\hat{n}, b)$. Either $\boldsymbol{x}_{ref}$ or $\boldsymbol{x}_{comp}$ will correspond to the centroid of the triangular

sub-region. So, we seek to find the interface $\widehat{\Gamma}$ such that the centroid of the triangular region, with critical vertex $\boldsymbol{P}_{r\star}$ and the two points of intersection of $\widehat{\Gamma}$ with the face of the cell, is either equal to $\boldsymbol{x}_{ref}$ or $\boldsymbol{x}_{comp}$ If the initial angle is chosen as $\theta^0 = \theta^0(\hat{n}^\star)$, the initial guess will be exact for a linear interface. To compute the interface $\Gamma^\star(\hat{n}^\star, b^\star)$, the following procedure is used.

1. Assume that vertex $\boldsymbol{P}_r$ is a critical vertex for either the reference or complementary region.
2. Define the vertices $\boldsymbol{v}_1 = \boldsymbol{P}_r + t_1(\boldsymbol{P}_{r+1} - \boldsymbol{P}_r)$ and $\boldsymbol{v}_2 = \boldsymbol{P}_r + t_2(\boldsymbol{P}_{r+2} - \boldsymbol{P}_r)$.
3. Compute t_1, t_2 such that

   $$\boldsymbol{x}_{ref} = \frac{1}{3}\left(\boldsymbol{P}_r + \boldsymbol{v}_1 + \boldsymbol{v}_2\right).$$

   If $(t_1, t_2) \in (0, 1]^2$, then $\boldsymbol{P}_r$ is a critical vertex.

   - Calculate $\hat{n}^\star$, $b^\star$ such that $\Gamma^\star(\boldsymbol{P}_r) = \hat{n}^\star \cdot \boldsymbol{P}_r + b^\star < 0$.
   - Set $\theta^0 = \theta(\hat{n}^\star)$.

   Otherwise, continue.
4. Compute t_1, t_2 such that

   $$\boldsymbol{x}_{comp} = \frac{1}{3}\left(\boldsymbol{P}_r + \boldsymbol{v}_1 + \boldsymbol{v}_2\right).$$

   If $(t_1, t_2) \in (0, 1]^2$, then $\boldsymbol{P}_r$ is a critical vertex.

   - Calculate $\hat{n}^\star$, $b^\star$ such that $\Gamma^\star(\boldsymbol{P}_r) = \hat{n}^\star \cdot \boldsymbol{P}_r + b^\star > 0$.
   - Set $\theta^0 = \theta(\hat{n}^\star)$.

   Otherwise, continue.
5. If $(t_1, t_2) \notin (0, 1]$ for both Steps 3 - 4, set $r \leftarrow r + 1$.

Note that uniqueness of the polygonal partition implies that for some vertex $\boldsymbol{P}_{r\star}$, parameters $(t_1, t_2) \in (0, 1]$. If either $t_0, t_1 = 0$, this would imply that $\Gamma^\star(\boldsymbol{P}_r) = 0$. Geometrically, the solutions such that $t_i \notin [0, 1]$ correspond to the case that $\widehat{\Gamma}$ intersects the ray $\overrightarrow{\boldsymbol{P}_r\boldsymbol{P}_{r+i}}$ at a point outside of the cell. The vector $\hat{n}^\star$ is the outward unit normal relative to the reference region, thus the sign in Steps 3 - 4 is chosen such that $\Gamma^\star(\boldsymbol{x}_{ref}) < 0$.

The computation of the parameters t_1, t_2 involves solving a linear system (3.30). Let

$$\boldsymbol{P}_r = \begin{pmatrix} P_r^x \\ P_r^y \end{pmatrix}$$

$$\begin{pmatrix} 3c_x - 3P_r^x \\ 3c_y - 3P_r^y \end{pmatrix} = \begin{pmatrix} P_{r+1}^x - P_r^x & P_{r+2}^x - P_r^x \\ P_{r+1}^y - P_r^y & P_{r+2}^y - P_r^y \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \tag{3.30}$$

$$Det = \frac{1}{(P_{r+1}^x - P_r^x)(P_{r+2}^y - P_r^y) - (P_{r+2}^x - P_r^x)(P_{r+1}^y - P_r^y)}.$$

Then,

$$\begin{aligned} t_1 &= 3 \cdot Det \cdot \left[(c_x - P_r^x)(P_{r+2}^y - P_r^y) - (c_y - P_r^y)(P_{r+2}^x - P_r^x) \right] \\ t_2 &= 3 \cdot Det \cdot \left[-(c_x - P_r^x)(P_{r+1}^y - P_r^y) + (c_y - P_r^y)(P_{r+1}^x - P_r^x) \right]. \end{aligned} \tag{3.31}$$

Through a simple geometric argument, the centroids $\boldsymbol{x}_{ref}$ and $\boldsymbol{x}_{comp}$ can be recovered exactly. However, unless the true interface is linear in cell Ω_i, the volume fractions recovered by the initial interface $\Gamma^\star$ will not be exact. It is worth noting that this initialization procedure is similar to the Moment-of-Fluid optimization problem, which seeks to exactly recover the volume fractions and minimize error in the centroid, while this initialization procedure exactly recovers the material centroids and allows for volume fraction discrepancy. In Cartesian coordinates, this choice of initial guess can lead to convergence of the MOF optimization problem in fewer iterations and the avoidance of local minima. While this method can produce an initial guess in curvilinear coordinates that converges to the global minimizer without aggressive optimization, it is no longer exact for linear interfaces or guaranteed to have a solution. This is due to the non-linear terms in centroid calculation in curvilinear coordinates (Sec. 3.1.4). Expressing the centroid in terms of parameters (t_1, t_2) that define the intersection of $\widehat{\phi}$ with the cell leads to a set of equations that is much more difficult to solve. In the case that the algorithm fails to find an acceptable solution (3.31) that satisfies $(t_1, t_2) \in (0, 1]$, the default guess of (3.29) is used.

3.2 Cell-Integrated Semi-Lagrangian Multiphase Advection

Semi-Lagrangian methods were first developed in the meteorological community as a way to circumvent the small time steps necessitated by Eulerian methods for stability [74]. Advection of quantities is performed by tracing particle paths on a fixed grid and updating the value based on the reconstructed value of the quantity at the foot of the particles's characteristic path. This avoids the strict CFL time step restriction because, unlike Eulerian methods, the stencil for updating a quantity at a computational node is not fixed; the stencil is determined dynamically as the points

required to form the variable reconstruction at the foot of the particle's characteristic path. For stability, the time step is limited only by the requirement that characteristic paths not intersect [8, 52].

Cell-Integrated Semi-Lagrangian (CISL) advection methods use a similar philosophy to Semi-Lagrangian methods. Rather than follow an individual particle path, the path of a region is traced over a set of characteristic paths. In general, some "departure region" region is mapped to a "target region" using characteristic tracing. State variables in a cell are updated using information from all sub-regions of the grid that map into that cell at the next time step. Various methods exist for updating state variables using CISL advection [50, 53, 55, 54]. Note that the CISL method is similar to a step of a pure Lagrangian advection method followed immediately by a remap step, however it does not incur as large of a computational cost for remapping, as the grid is deformed only slightly by the single step of the advection procedure. The time step restriction is such that the characteristic paths of cell vertices not cross [45], which has the physical interpretation that material is neither spontaneously created nor destroyed.

CISL advection methods are natural for multiphase simulations because exact conservation of mass, momentum, and energy is trivial. Additionally, multiphase advection is not bound by the (often strict) acoustic time step restriction. For methods based on Riemann solvers, the maximum time step is based on the speed of the fastest wave in the system, which is a function of the material velocity and the speed of sound of the material. In the presence of a material in which sound travels quickly (e.g. water), this may necessitate a small time step, despite the possibility that the material velocity is small, relative to mesh size [36, 35]. Again, because the CISL advection method updates cell variables based on the dynamic intersection of the departure region or target region with the mesh, this advection strategy is not subject to the strict CFL or acoustic time step restriction for stability. The algorithm for split CISL advection on structured grids and 2D unsplit CISL advection on unstructured grids will be discussed in the context of multiphase flow.

3.2.1 1D Time Step Computation

Define a 1D cell as some interval $\Omega_i = [x_{i-1/2}, x_{i+1/2}]$ with centroid x_i. Cell-Integrated Semi-Lagrangian methods define velocity at cell faces, with $u(x_{i\pm1/2}) = u_{i\pm1/2}$. Define the "departure region" Ω_i^D as the region which will be mapped into cell Ω_i at the next time step. In this framework, known as back-tracing or a backward sweep, the cell Ω_i is the "target region," i.e. the region into

which material is mapped. Alternatively, advection can be performed in the framework of forward tracing, or a forward sweep, in which Ω_i^T is the target region into which material from cell Ω_i, the departure region is mapped. Consider a time step Δt. Then the departure and target regions are defined as

$$\Omega_i^D = [x_{i-1/2} - \Delta t u_{i-1/2}, x_{i+1/2} - \Delta t u_{i+1/2}] \tag{3.32}$$

and

$$\Omega_i^T = [x_{i-1/2} + \Delta t u_{i-1/2}, x_{i+1/2} + \Delta t u_{i+1/2}]. \tag{3.33}$$

can be computed as

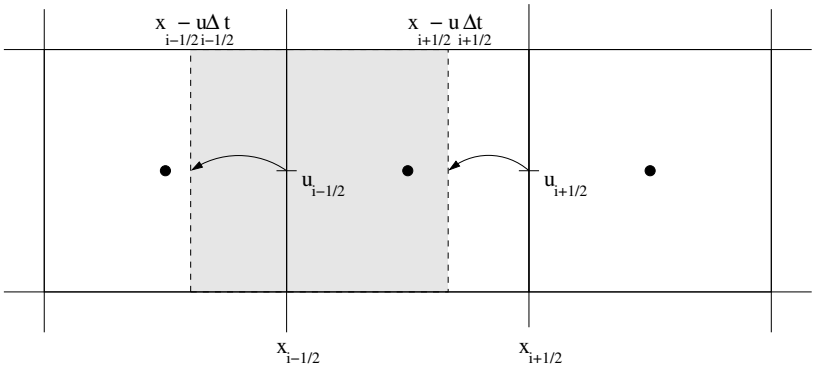

Figure 3.8: The departure region for cell $\Omega = [x_{i-1/2}, x_{i+1/2}]$ is shaded, for a time step Δt and face velocities $u_{i\pm 1/2}$, along with backtraced characteristic paths. Cell centroids are shown as filled circles.

$$(u_{i+1/2} - u_{i-1/2})\Delta t_i^{D,collapse} = (x_{i+1/2} - x_{i-1/2})$$
$$(u_{i-1/2} - u_{i+1/2})\Delta t_i^{T,collapse} = (x_{i+1/2} - x_{i-1/2})$$

for the backward sweep and forward sweep advection respectively. Once the time to cell collapse has been computed, a maximum time allowable step for the cell is determined. Additionally, a time step based on the CFL condition is computed.

$$\Delta t_i^{CFL} = CFL \frac{x_{i+1/2} - x_{x-1/2}}{\max(|u_{i+1/2}|, |u_{i-1/2}|)} \tag{3.34}$$

The final time step for a cell is then computed as (3.35), where $\alpha < 1$ is some maximum fraction of the time to cell collapse that we are willing to allow the characteristics to travel.

$$\Delta t_i = \begin{cases} \alpha \Delta t_i^{collapse}, & \text{if } 0 < \alpha \Delta t_i^{collapse} < \Delta t_i^{CFL} \\ \Delta t_i^{CFL}, & \text{otherwise} \end{cases} \tag{3.35}$$

Note that there exist cases (e.g. $u_{i+1/2} = u_{i-1/2}$) such that there is no restriction on the time step (i.e. $\Delta t^{collapse}$ is negative or infinite). For this reason, the cell time step is taken as the minimum of the CFL time step and some fraction of the time to cell collapse.

3.2.2 2D Time Step Computation

The method for computing the time step on 2D unstructured grids is similar in spirit to the 1D method. The time to cell collapse is computed, along with a CFL based time step. The cell time step is then taken as the minimum of the CFL time step and some factor of the time to cell collapse. Assume that the vertices of cell Ω_i are labeled in a counter-clockwise fashion, $\{\boldsymbol{v}_i\}_{i=1}^3$. Let $\begin{pmatrix} X_i \\ Y_i \end{pmatrix}$ be the components of $\boldsymbol{v}_i$. Then, the volume of Ω_i in Cartesian coordinates is calculated as

$$|\Omega_i| = \frac{1}{2} \left| (\boldsymbol{v}_2 - \boldsymbol{v}_1) \times (\boldsymbol{v}_3 - \boldsymbol{v}_1) \right|.$$

We wish to find the smallest positive time such that the departure cell Ω_i^D has zero volume. If the set of vectors $\{\boldsymbol{U}_i\}_{i=1}^3$ represents the velocity at the corresponding cell vertices, then the time of cell collapse is the the time step Δt that solves

$$|\Omega_i^D| = \frac{1}{2} \left| (\Delta \boldsymbol{v}_2 - \Delta t \Delta \boldsymbol{U}_2) \times (\Delta \boldsymbol{v}_3 - \Delta t \Delta \boldsymbol{U}_3) \right| = 0. \tag{3.36}$$

Here, let $\Delta \boldsymbol{v}_i = \boldsymbol{v}_i - \boldsymbol{v}_1$ and $\Delta \boldsymbol{U}_i = \boldsymbol{U}_i - \boldsymbol{U}_1$. Let $\begin{pmatrix} U_i \\ V_i \end{pmatrix}$ be the components of $\boldsymbol{U}_i$. This yields an expression (3.37) for time to cell collapse $\Delta t^{collapse}$ that may be quadratic, linear, or constant, depending on the relationship between the vertex locations and nodal velocities. Recognizing the physical motion associated with the expression for cell collapse time is not as intuitive as in 1D, but it may be noted that the constant case corresponds to translation of the cell, whereas the linear case corresponds to compression or expansion.

$$\begin{aligned} &a\Delta t^2 + b\Delta t + c = 0 \\ &a = \Delta U_2 \Delta V_3 - \Delta V_2 \Delta U_3 \\ &b = \Delta Y_2 \Delta U_3 + \Delta X_3 \Delta V_2 - \Delta X_2 \Delta V_3 - \Delta Y_3 \Delta U_2 \\ &c = \Delta X_2 \Delta Y_3 - \Delta Y_2 \Delta X_3 \end{aligned} \tag{3.37}$$

Labeling the vertices $\boldsymbol{v}_i$ in periodic fashion, define the CFL time step as

$$\Delta t_i^{CFL} = CFL \frac{\min\limits_{i=1,\dots,3} (||\boldsymbol{v}_{i+1} - \boldsymbol{v}_i||_2)}{\max\limits_{j=1,\dots,3} (||\boldsymbol{u}_j||_2)}.$$

$$\Delta t_i = \begin{cases} \alpha \Delta t_i^{collapse}, & \text{if } 0 < \alpha \Delta t_i^{collapse} < \Delta t_i^{CFL} \\ \Delta t_i^{CFL}, & \text{otherwise} \end{cases} \tag{3.38}$$

Because large deformation is possible in unsplit advection, and the task of finding all cells Ω_j on the unstructured grid such that $\Omega_j \cap \Omega_i^D \neq \emptyset$ can be difficult, parameter α and the CFL number are chosen fairly conservatively. For all 2D unsplit cases presented, I take $\alpha = 1/4$ and $CFL = 1/2$. Note that (3.37) is posed for backward tracing, but the equation for the time to cell collapse for forward tracing characteristics is similar, also resulting in an equation that is quadratic, linear, or constant, and must be solved for the smallest positive time such that Ω_i^T has zero volume.

3.2.3 Characteristic Mapping

The Cell-Integrated Semi-Lagrangian procedure of characteristic mapping from the departure region to the target region will be discussed in this section. Again, define cells Ω_i as the triangle with vertices $\{\boldsymbol{v}_i\}_{i=1}^3$ with components $\boldsymbol{v}_i = \begin{pmatrix} X_i \\ Y_i \end{pmatrix}$. The changes necessary for implementation of forward characteristic tracing will be mentioned.

Mapping data from the departure region to the target region requires that the velocity is known at each point. A linear velocity profile is assumed in the cell. Velocity at faces $\boldsymbol{x}_{i\pm1/2}$ in the departure region are equal to $\boldsymbol{u}_{i\pm1/2}$, as in Section 3.2.1 for calculations in 1D, and velocities at vertices $\boldsymbol{v}_j^D$ of the departure cell are equal to $\boldsymbol{U}_j$, as defined in Section 3.2.2 for calculations in 2D. In 1D, we may find the linear mapping from the departure region to the target region for some arbitrary point $x_d \in [x_{i-1/2}^D, x_{i+1/2}^D] = \Omega_i^D$ as

$$\mathcal{L}x_d = \frac{x_d - x_{i-1/2}^D}{x_{i+1/2}^D - x_{i-1/2}^D} + x_{i-1/2}^T.$$

Here $\Omega_i^T = [x_{i-1/2}^T, x_{i+1/2}^T]$. In 2D, we wish to compute the mapping from the departure region to the target region. Define the linear operator $\mathcal{L}$ that satisfies

$$\mathcal{L} : \Omega_i^D \to \Omega_i^T. \tag{3.39}$$

Arbitrarily writing the mapping in terms of vertex $\boldsymbol{v}_1^T$ and $\boldsymbol{v}_1^D$, its pre-image in the departure region,

$$\mathcal{L}\boldsymbol{x}_d = A(\boldsymbol{x}_d - \boldsymbol{v}_1^D) + \boldsymbol{v}_1^T. \tag{3.40}$$

The first row of A satisfies

$$\begin{pmatrix} X_2^D - X_1^D & Y_2^D - Y_1^D \\ X_3^D - X_1^D & Y_3^D - Y_1^D \end{pmatrix} \begin{pmatrix} A_{11} \\ A_{12} \end{pmatrix} = \begin{pmatrix} X_2 - X_1 \\ X_3 - X_1 \end{pmatrix}, \tag{3.41}$$

and the second row of A satisfies

$$\begin{pmatrix} X_2^D - X_1^D & Y_2^D - Y_1^D \\ X_3^D - X_1^D & Y_3^D - Y_1^D \end{pmatrix} \begin{pmatrix} A_{21} \\ A_{22} \end{pmatrix} = \begin{pmatrix} Y_2 - Y_1 \\ Y_3 - Y_1 \end{pmatrix}. \tag{3.42}$$

Let Γ be the restriction of a linear interface to some cell (3.43). A linear interface Γ can also be mapped from a cell in the departure region into the cell of interest. Consider Γ in cell Ω_j, where

$$\Gamma = \{\boldsymbol{x} | \widehat{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_j) + b = 0, \boldsymbol{x} \in \Omega_j\}. \tag{3.43}$$

Then, the mapped interface $\mathcal{L}\Gamma \subset \Omega_i$ is defined as

$$\mathcal{L}\Gamma = \{\mathcal{L}\boldsymbol{x} | \widehat{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_j) + b = 0, \boldsymbol{x} \in \Omega_j\} = \{\boldsymbol{x} | \widehat{n}_\star \cdot (\boldsymbol{x} - \mathcal{L}\boldsymbol{x}_j) + b = 0, \boldsymbol{x} \in \Omega_i\}. \tag{3.44}$$

Note that the mapped interface $\mathcal{L}\Gamma$ will have a different orientation from the original interface. The orientation of the mapped interface is denoted as $\widehat{n}_\star$. By enforcing the condition (3.44) that the image under $\mathcal{L}$ of the set of point on Γ in Ω_j is equal to the points on the new interface in cell Ω_i, we get

$$\widehat{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_j) + b = \widehat{n}_\star \cdot (\mathcal{L}\boldsymbol{x} - \mathcal{L}\boldsymbol{x}_j) + b. \tag{3.45}$$

Substituting in the expression for $\mathcal{L}\boldsymbol{x}$, we recover

$$\widehat{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_j) + b = \widehat{n}_\star \cdot ([\, A(\boldsymbol{x} - \boldsymbol{v}_1^D) + \boldsymbol{v}_1 \,] - [\, A(\boldsymbol{x}_j - \boldsymbol{v}_1^D) + \boldsymbol{v}_1 \,]). \tag{3.46}$$

Simplifying the right-hand side, we get

$$\widehat{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_j) = \widehat{n}_\star A \cdot (\boldsymbol{x} - \boldsymbol{x}_j), \tag{3.47}$$

which implies that

$$\widehat{n}_\star = (A^{-1})^T \widehat{n}.$$

This definition of the mapping of material from the departure region to the target cell is mathematically precise, but it is not algorithmically useful. A practical algorithm for calculating mapped

material volumes in the region of interest is now presented. It will be done in the framework of backward characteristic tracing, i.e. calculating the departure region Ω_i^D that is mapped to the cell of interest Ω_i at the next time step. Recall that for backward tracing of characteristics, the target region Ω_i^T is simply the cell of interest, Ω_i. A cut cell, which contains more than one material, can be written as the union of material subregions within that cell (3.48). In 1D, these material sub-regions can be written as intervals, while in 2D, these material sub-regions can be represented as convex polygonal figures. These polygonal figures can be triangulated using a standard ear-clipping algorithm, to represent the material sub-region as a set of triangles.

$$\Omega_j = \bigcup_{m=1}^{M} \Omega_j^m \tag{3.48}$$

Once the intersection of the departure region with the mesh has been found, the departure region is intersected with all material sub-regions in each cell that it intersects. Represent the triangular departure region, with vertices $\left\{\boldsymbol{v}_i^D\right\}_{i=1}^3$ and centroid $\boldsymbol{x}^D$, as a level set function ϕ^D that is positive in the triangle and negative outside of the triangle. Denote the components of $\boldsymbol{v}_i^D$ as $\begin{pmatrix} X_i^D \\ Y_i^D \end{pmatrix}$. This level set function can be constructed as the minimum of three linear level set functions, each of which is positive in the direction of the centroid $\boldsymbol{x}^D$ and zero at two of the vertices.

$$\begin{aligned} &\phi^D(\boldsymbol{x}) = \min_j \left\{\phi_j^D(\boldsymbol{x})\right\} \\ &\begin{cases} \phi_j^D(\boldsymbol{v}_j) = \phi_j^D(\boldsymbol{v}_{j+1}) = 0 \\ \phi_j^D(\boldsymbol{x}^D) > 0 \end{cases} \end{aligned} \tag{3.49}$$

The triangulated material sub-regions in each cell can then be cut by the level set function (3.49) by sequentially cutting each material triangle with the level set functions that make up the edges of the triangle. The result of cutting each triangle with an individual level set function ϕ_j^D is then triangulated using the algorithm in Section 3.1.3.

1. Suppose cell $\Omega_j \cup \Omega_i^D \neq \emptyset$ for cells Ω_j, $j = 1, \ldots, N$.
2. For $j = 1, \ldots, N$
   (a) If Ω_j is a cut cell with a linear MOF interface, triangulate each material sub-region as in Section 3.1.3, yielding $\Omega_j = \bigcup_{l=1}^{L} T_l$, for triangle T_l, with material identities $\{m_l\}_{l=1}^L$.
   (b) Append $\{T_l\}_{l=1}^L$ to the list $\mathcal{T}$ of triangles to be cut by the departure region. Append $\{m_l\}_{l=1}^L$ to the list $\mathcal{M}$ of triangle material identities.

3. For each level set function $\phi_j^D(\boldsymbol{x})$, $j = 1, \ldots, 3$

   (a) Cut each triangle in $\mathcal{T}$ with $\phi_j^D(\boldsymbol{x})$ and triangulate the result, using the algorithm in Section 3.1.3.

   (b) Append the newly generated triangle and material information to $\mathcal{T}$ and $\mathcal{M}$.

The result of this procedure is that the departure region can be expressed as a set of triangles with material identities. This is used to map the material sub-regions from the departure region to the cell of interest. In 1D, the analogue of this procedure is trivial, as the MOF interface is represented as a single point. After the backward tracing procedure is completed for a given cell, it will have all necessary data to perform the Moment-of-Fluid interface reconstruction.

The procedure for forward tracing of characteristics is similar. In this case, the departure region is the cell of interest, Ω_i. Rather than "gathering" data from surrounding cells, data is "scattered" from the target region to all cells that intersect the target region. All material sub-regions in the cell of interest are triangulated and mapped into the target region. The set of mapped material triangles $\{\mathcal{LT}_k\}$ is cut with the level set function defining each cell Ω_j such that $\Omega^T \cap \Omega_j \neq \emptyset$. Volume and moment data for each material sub-region in $\Omega^T \cap \Omega_j$ are distributed to cell Ω_j for the purpose of the Moment-of-Fluid interface reconstruction. Note that for forward tracing advection, a cell Ω_i does not have all necessary data for the MOF interface reconstruction until all cells Ω_j, such that $\Omega_j \cap \Omega_i \neq \emptyset$, have been advanced in time.

3.2.4 CISL State Variable Update

The Cell-Integrated Semi-Lagrangian advection method is natural for multiphase flow problems, as the "advected" state variables are updated such that mass, momentum, and energy are conserved exactly from the departure region to the target region. The method will be presented in terms of backward characteristic tracing, and the differences for forward tracing will be addressed.

Volume fractions are advanced first, to determine which materials occupy a cell. Consider a cell occupied by a material m, with the interface defined by some 2D plane $\Gamma^m = \{\boldsymbol{x} | \phi^m(\boldsymbol{x}) = 0\}$, where the normal vector points away from the region occupied by material m. Let $\left\{\Omega_{j'}\right\}$ denote the set of all cells such that $\Omega_{j'} \cap \Omega_i^D \neq \emptyset$.

$$\phi_j^m(\boldsymbol{x}) = \widehat{n} \cdot (\boldsymbol{x} - \boldsymbol{x}_j) + b \tag{3.50}$$

Note that because the normal vector for ϕ_j^m points outward from the region Ω_j^m we have

$$\phi_j^m(\boldsymbol{x}) = \begin{cases} < 0, & \boldsymbol{x} \in \Omega_j^m \\ = 0, & \boldsymbol{x} \in \Gamma_j^m \\ > 0, & \text{otherwise.} \end{cases}$$

This can be used to define a Heaviside function for material m,

$$H_j^m(-\phi^m(\boldsymbol{x})) = \begin{cases} 1, & \boldsymbol{x} \in \Omega_j^m \\ 0, & \text{otherwise,} \end{cases} \tag{3.51}$$

where $H(z)$ is the standard Heaviside function that is one for positive z, and zero otherwise. Define $\mathcal{L}$ and $\widehat{n}_\star$ as in Section 3.2.3, as the mapping from the departure region to the cell of interest and the mapped normal vector. Additionally, $\mathcal{L}\Omega_{j'}$ denotes the image of $\Omega_{j'}^m$ in the cell of interest Ω_i under the mapping $\mathcal{L}$. Then,

$$H_{j,\star}^m(\boldsymbol{x}) = H_j^m\left(-\left(\widehat{n}_\star \cdot (\boldsymbol{x} - \mathcal{L}\boldsymbol{x}_j) + b\right)\right). \tag{3.52}$$

The volume fraction F_i^m in the cell of interest Ω_i is calculated as

$$F_i^m = \frac{\sum_{j,} \int_{\mathcal{L}\Omega_{j'}^m \cap \Omega_i} H_{j,\star}(\boldsymbol{x}) d\Omega}{\int_{\Omega_i} d\Omega}. \tag{3.53}$$

Note that volume fractions should *always* be calculated in the target region, as this will be the region occupied by the material at the next time step. While calculation of volume fractions will be equal in the departure region and the target region in Cartesian coordinates, the same is not true in curvilinear coordinates.

Density $\rho^{n+1,m}$ for each material m in a cell of interest Ω_i is updated as the total mass of that material in the departure region, divided by the volume of the material sub-region mapped to the cell of interest. This density is the time advanced solution to the continuity equation,

$$\rho_t + \nabla \cdot (\rho \boldsymbol{u}) = 0. \tag{3.54}$$

If F_i^m is less than the cutoff value, then the fraction is truncated to zero (as described in Section 3.1.1) and the material density is set to zero. Let the function $\varrho_{j'}^m(\boldsymbol{x})$ denote some reconstruction of density for material m in cell $\Omega_{j'}$.

$$\rho_i^{a,m} = \frac{\sum_{j'} \int_{\Omega_{j'}^m \cap \Omega_i^D} \varrho_{j'}^{n,m}(\boldsymbol{x}) d\Omega}{\sum_{j'} \int_{\Omega_{j'}^m \cap \Omega_i^D} d\Omega} \tag{3.55}$$

$$\rho_i^{n+1,m} = \frac{\sum_{j'} \int_{\Omega_{j'}^m \cap \Omega_i^D} \varrho_{j'}^{n,m}(\boldsymbol{x}) d\Omega}{F_i^m \int_{\Omega_i} d\Omega} \tag{3.56}$$

The advected cell-centered velocity $\boldsymbol{u}_i^a$ is updated as the average velocity in the cell of interest that exactly conserves momentum in the departure region. Let $\varrho\mathcal{U}(x)_j^{n,m}$ denote the linear reconstruction of momentum of material "m" at time level t^n in cell Ω_j.

$$\boldsymbol{u}_i^a = \frac{\sum_{j'} \sum_{m=1}^{M} \int_{\Omega_{j'}^m \cap \Omega_i^D} \varrho\mathcal{U}_{j'}^{n,m}(\boldsymbol{x}) d\Omega}{\sum_{m=1}^{M} (F_i^m \rho_i^{n+1} \int_{\Omega_i} d\Omega)} \tag{3.57}$$

Similarly, the specific energy $E_i^{a,m}$ for material m is updated as the average energy such that $\rho^m E^m$ is conserved from the departure region to the cell of interest. Let $\varrho\mathcal{E}_j^{n,m}$ denote the energy of material m.

$$E_i^{a,m} = \frac{\sum_{j'} \int_{\Omega_{j'}^m \cap \Omega_i^D} \varrho\mathcal{E}_{j'}^{n,m}(\boldsymbol{x}) d\Omega}{F_i^m \rho_i^{n+1,m} \int_{\Omega_i} d\Omega} \tag{3.58}$$

Note that while it is crucial to calculate material volume in the target region, mass, momentum, and energy can be calculated in the departure region because these quantities are conserved. The method for evolving velocity and energy from $\boldsymbol{u}^a$ and E^a to $\boldsymbol{u}^{n+1}$ and E^{n+1} differs subtly for each fluid algorithm. As such, this is addressed in Chapter 4.

The method for updating state variables differs slightly for forward tracing. Rather than sum contributions over all cells that intersect the departure region, contributions are summed over all cells such that the target region intersects the cell of interest. Letting $\left\{\Omega_{j'}\right\}$ be the set of all cells such that $\Omega_{j'}^T \cap \Omega_i \neq \emptyset$, (3.53) remains unchaged. Density in the target region is computed as

$$\rho_{j'}^T = \rho_{j'} \frac{|\Omega_{j'}|}{|\Omega_{j'}^T|},$$

i.e. the mass of the material in the original cell divided by the volume of the target region. So for a constant reconstruction in density, (3.56) becomes

$$\rho_i^{n+1,m} = \frac{\sum_{j'} \int_{\Omega_{j'}^{m,T} \cap \Omega_i} \rho_{j'}^{n,m,T} d\Omega}{F_i^m \int_{\Omega_i} d\Omega}. \tag{3.59}$$

If a slope reconstruction is used in density, the pre-image of $\Omega_{j'}^T \cap \Omega_i$ in $\Omega_{j'}$ is found. Let $\Omega_{j',i}^D$ denote the pre-image of the target region of $\Omega_{j'}$ with Ω_i Then, density in the intersection of the

target region with the cell of interest is calculated as the mass in the pre-image $\Omega^D_{j',i}$ divided by the volume of $\Omega^D_{j',i}$. Transport of momentum and energy proceeds as with transport of mass.

3.2.4.1 Slope Reconstruction. A number of methods for slope reconstruction were implemented in 1D, including the constant reconstruction, as well as MINMOD, van Leer, and Superbee limiting. The most consistently accurate method used a MINMOD slope for each conserved quantity, ρ, $\rho\boldsymbol{U}$, and ρE. As a note, the slope in any cell cut by the interface is take to be zero.

Results for 1D MINMOD slope versus constant reconstruction in density are presented in Section 5.8.2. It is seen that the sharpness of shock fronts and contact discontinuities are better captured with the MINMOD reconstruction (3.60 - 3.62). Additionally, it is found that for the Shock-Turbulence Interaction (Sec. 5.1), use of MINMOD slope limiting for all conserved quantities produced results equivalent to constant reconstruction, or MINMOD for density alone, with an additional level of mesh refinement. The reconstructed slope is taken to be zero when computing face-centered density and momentum quantities. This was found to produce the most regular results. Because slope reconstruction on unstructured grids is a non-trivial task, all 2D results presented assume a constant reconstruction in density. Consider $\varphi(x)$, the linear reconstruction of some arbitrary conserved quantity ϕ.

$$\varphi(x) = \phi_i + \phi'_{i,MM}(x - x_i) \tag{3.60}$$

$$\phi'_{i,MM} = \begin{cases} SGN \cdot \min\left(|\frac{\phi_{i+1}-\phi_i}{\Delta x}|, |\frac{\phi_i-\phi_{i-1}}{\Delta x}|\right), & \text{if } \left(\frac{\phi_{i+1}-\phi_i}{\Delta x}\right)\left(\frac{\phi_i-\phi_{i-1}}{\Delta x}\right) > 0, \Omega_i \text{ not cut} \\ 0, & \text{otherwise} \end{cases} \tag{3.61}$$

$$SGN = \begin{cases} 1, & \left(\frac{\phi_{i+1}-\phi_i}{\Delta x}\right) > 0 \\ -1, & \left(\frac{\phi_{i+1}-\phi_i}{\Delta x}\right) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.62}$$

3.2.5 Alternate Advection Strategies

A number of alternate methods for performing advection, based on the same techniques, exist. Second order Semi-Lagrangian Runge-Kutta methods [53] perform time integration by calculating the advective velocity at the midpoint of the characteristic path in space and time of the specified for standard RK2 methods. Alternating forward/backward tracing [37, 46] for multi-material

simulations has the property of decreasing volume fluctuations caused by errors in calculating the characteristic paths. In this work, this forward/backward tracing was found to be the most effective method on 2D unstructured grids. In 1D, a backward tracing fixed-point method is utilized. This method uses multiple iterations of backward tracing to obtain a more accurate prediction of velocity at time t^{n+1}. Error results for these methods was comparable in the tests presented, as time error is dominated by the first order time error made in the discretization of the asymptotically-preserving all-speed pressure method and the dissipative error in state variable reconstructions [45].

3.2.5.1 Runge-Kutta Advection. The spatially second-order Runge-Kutta method is implemented for forward tracing advection. As noted in [53, 54], backward tracing methods require multiple iterations to produce a second order method, and do not lend themselves to higher order methods, due to the nature of backward characteristic tracing material transport. Let $\Omega_j^{T,1/2}$ be the image of cell $\Omega_j = \{\boldsymbol{x}_i\}_{i=1}^3$ under forward characteristic tracing for a time step $\Delta t/2$.

$$\Omega_j^{T,1/2} = \{\widehat{\boldsymbol{x}}_i\}_{i=1}^3 = \left\{\boldsymbol{x}_i + \frac{\Delta t}{2}\boldsymbol{u}_i\right\}_{i=1}^3 \tag{3.63}$$

Let $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^3$ denote the vertices of the triangular region $\Omega_j^{T,1/2}$. Given velocity as cell nodes, a bi-linear or linear interpolation of velocity at the point $\widehat{\boldsymbol{x}}_j$ can be computed for rectangular or triangular grids, respectively. Denote the linear interpolation of velocity to a point $\widehat{\boldsymbol{x}}_j$ as

$$\widehat{\boldsymbol{u}}_j = \boldsymbol{u}_{interp}(\widehat{\boldsymbol{x}}_j, t^n). \tag{3.64}$$

Then, the spatially second-order Runge-Kutta target region is computed as

$$\Omega_j^T = \{\boldsymbol{x}_j + \Delta t\widehat{\boldsymbol{u}}_j\}_{j=1}^3 . \tag{3.65}$$

The Runge-Kutta method then proceeds as a standard single iteration of the forward tracing CISL advection method, with the target region Ω_i^T defined as in (3.65). Note that this method computes the velocity at the characteristic path "midpoint" at time level t^n rather than time level $t^{n+1/2}$. Computation of velocity at an arbitrary point at time level $t^{n+1/2}$ would require two interface reconstructions and two linear solves to update pressure per time step. For passively advected deforming material interface tests, velocity is prescribed. This makes it straightforward to compute velocity at the midpoint in space and time,

$$\boldsymbol{u}(\boldsymbol{x} + \frac{\Delta t}{2}\boldsymbol{u}, t + \frac{\Delta t}{2}).$$

As such, this is used in structured grid computations for filament capturing, where maintaining accuracy is necessary to capture the formation of thin filament structures.

3.2.5.2 Alternating Forward-Backward Advection. An alternating forward-backward advection method is used for unsplit transport, here applied to unstructured grids. This method was shown in [46] to produce results comparable to RK2 advection for material volume conservation and interface reconstruction in incompressible flow tests on 3D rectangular grids.

Given a time step Δt and an advective velocity $\boldsymbol{U}$ for each node over the entire mesh, forward tracing advection is performed with a time step of $\Delta t/2$. State variables are reconstructed at time level $t^{n+1/2}$ to yield $\rho^{n+1/2}$, $\boldsymbol{u}^{n+1/2}$, and $E^{n+1/2}$. The advective velocity $\boldsymbol{U}$ is again used, now to perform backward tracing advection with a time step of $\Delta t/2$ on the state variables at time level $t^{n+1/2}$ to obtain the advected state variable quantities at time level t^{n+1}.

3.2.5.3 Backward Tracing Velocity Fixed Point Method. While forward tracing of characteristics tracks material from its current location along its path of motion in a Lagrangian fashion, backward tracing defines a region which is mapped to the cell of interest at the next time step [54]. In the continuous sense, as material arrives in the cell of interest, it will be traveling with a speed equal to the cell's advective velocity at the next time step.

To better mimic this property, a fixed-point iteration method is used for backward characteristic tracing to generate a more accurate prediction of velocity at the next time step. Advection is performed, the advective velocity at the next time step is calculated, and advection is performed at time t^n again, iterating until convergence in the advective velocity. Error analysis in 1D (Section 5.1) demonstrates that this fixed-point procedure for calculating the departure volume converges in typically 2 - 3 iterations. While this procedure leads to a slight increase in error, associated with diffusion at a shock front, it reduces overshoots at discontinuities. The algorithm for the velocity fixed point iteration is presented.

1. Set iteration counter $k = 0$. Given state variables $\rho^{n,0}$, $\boldsymbol{u}^{n,0}$, $E^{n,0}$.
2. Calculate advective node/face velocity $\boldsymbol{u}^{adv,k}$
3. For each cell Ω_i...
   (a) Calculate departure region Ω_i^D.
   (b) Map material $\rho^{n,0}$, $\boldsymbol{u}^{n,0}$, and $E^{n,0}$ to cell Ω_i.

(c) If Ω_i is cut after advection, perform the MOF reconstruction.

(d) Calculate advected state variables $\rho^{a,k+1}$, $\rho^{n+1,k+1}$, $\boldsymbol{u}^{a,k+1}$, $E^{a,k+1}$, and $p^{a,k+1}$.

4. If the iterations have converged or the maximum iteration is reached, exit. Otherwise, set $k \leftarrow k+1$ and go to Step (2).

Note that calculating the advective velocity involves evolving $p^{a,k} \rightarrow p^{n+1,k}$ for each iteration k. This procedure requires solving a linear equation for pressure, which takes a significant amount of computational time, relative to the time for advection and interface reconstruction. A computational study in Section 5.1 shows that the L2 norm of the difference between density after 2 iterations and after 5 iterations is less than 1%.

3.2.6 Unsplit CISL Advection Algorithm

All components of the Cell-Integrated Semi-Lagrangian advection method have been described. The full algorithm for performing CISL over the entire grid is now presented.

1. Calculate a global time-step Δt_{global} to advance from time level $t^n \rightarrow t^{n+1}$.

   (a) Initialize Δt_{global} to some arbitrary large value.

   (b) For each cell $\Omega_i,\ i = 1, \ldots, N$, compute Δt_i using (3.35) or (3.38).

   (c) If $\Delta t_i < \Delta t_{global}$, then set $\Delta t_{global} \leftarrow \Delta t_i$.

2. Compute advective velocity at cell faces/vertices.

3. For each cell Ω_i, calculate Ω_i^D and Ω_i^T.

   (a) Compute the linear mapping $\mathcal{L} : \Omega_i^D \rightarrow \Omega_i^T$.

4. Map all material sub-regions $\Omega_i^{m,D}$ to the target region.

5. Calculate volume and centroid data on the grid at time level t^{n+1}, using the methods described in Sections 3.1.3.1 and 3.1.4.

6. For each cut cell, perform the MOF interface reconstruction, given volume fraction and centroid data.

7. Update density ($\rho^n \rightarrow \rho^{n+1}$), cell-centered velocity ($\boldsymbol{u}^n \rightarrow \boldsymbol{u}^a$), and material energy ($E^n \rightarrow E^a$) as described in Chapter 4.

8. If the backward tracing fixed-point method is used and the maximum iteration is not reached, increment the iteration counter and return to Step (2).

3.3 Filament Capturing with Multimaterial MOF

Let us define a "filament" as some thin material structure, under-resolved by grid structure, that separates two regions of the same material in a cell. Let us also define the material in the cell cut by the filament as the "bulk" material. Rather than methods which depend on user-defined thresholds to identify and capture filaments, this method uses a purely geometric approach. If a two "disconnected" regions of the same material in a cell are detected, this may indicate the presence of a filament. In this case, a conglomeration procedure is applied to distinguish between the disconnected regions of material. Finally, the reconstruction problem is posed for three materials separated by two interfaces, rather than trying to capture a filament using a single interface separating two materials.

In this section, the procedure for detecting and resolving filaments is presented. First, we discuss the conglomeration procedure applied to a triangulated cell occupied by one or more materials. This detects disconnected regions of material in a single cell, which may indicate the presence of a filament. The method for classifying triangles as "adjacent" for the purpose of conglomeration is defined. Then, the logic which determines whether or not a filament exists, given a set of material conglomerates in a cell, is presented. The introduction of the "twin fluid," an alternative label introduced when two conglomerates of the same material exists within a cell, is discussed. Next, the logic for dealing with pathological cases, including flotsam, is presented. Finally, the fully generalized algorithm for conglomeration and filament capturing is presented.

3.3.1 Basic Algorithm

In general, the Cell-Integrated Semi-Lagrangian advection procedure maps portions of material from multiple cells to a single cell (the target cell). Each of these cells is potentially cut by an interface and intersected with the departure region that will be mapped to the target cell. Once all portions of the intersected cells have been mapped to the target cell, the target cell is occupied by a set of triangulated polygons. A conglomeration procedure is used to identify which triangular regions should be grouped together when reconstructing the material interface. Each triangle carries a material ID, and adjacent triangular figures with matching material ID's should be grouped into the same region, or conglomerate. The simplest algorithm for forming these material conglomerates is as follows.

1. Choose a starting triangle T_i.
2. Identify all ungrouped triangles T_j that are adjacent to T_i and have the same material ID.
3. If new triangles have been identified, repeat from Step 1 for each T_j. If no new triangles are identified, then continue.
4. Denote grouped triangles a completed conglomerate C_i.
5. Compute volume fraction F_i and centroid $\boldsymbol{x}_i$ of conglomerate C_i.
6. Identify any ungrouped T_k. If no such triangle exists, all conglomerates have been found; exit. If there does exist such a triangle, return to Step 1.

Once all conglomerates have been found, several situations may exist. A single conglomerate indicates that the cell is pure (occupied by a single material). Two conglomerates indicates two materials separated by a single interface, a configuration which may be resolved by the Moment-of-Fluid interface reconstruction or other PLIC strategy. The presence of three or more conglomerates may indicate the existence of a filament, the presence of flotsam, or some other material configuration. The logic for resolving these cases is discussed in Sections 3.3.3 - 3.3.5.

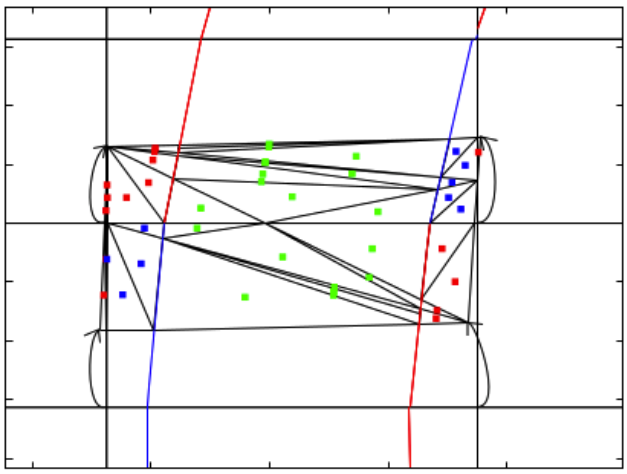

Figure 3.9: The departure region is found by backtracing characteristics (arrows). The departure region is intersected with the mesh, including any interfaces, and the region is triangulated. Colored lines represent a material interface. Colored squares in the departure region indicate material labels of the triangulated regions. The red and blue labels indicate "twin" materials; the green label indicates the filamentary material.

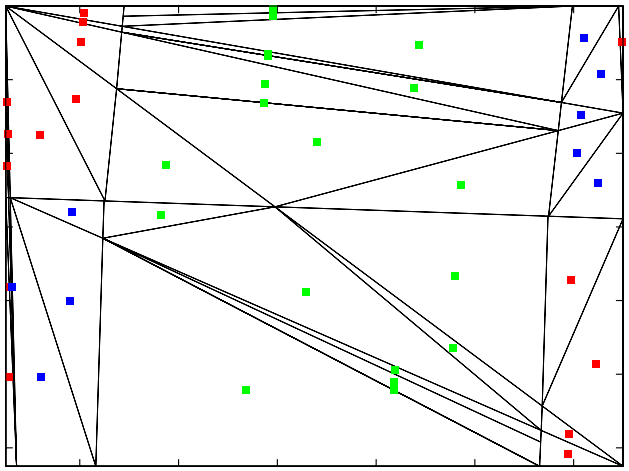

Figure 3.10: The triangulated departure region from Figure 3.9 is mapped into the cell of interest. Material ID's from the departure region are maintained. Note that red and blue material ID's represent the same material.

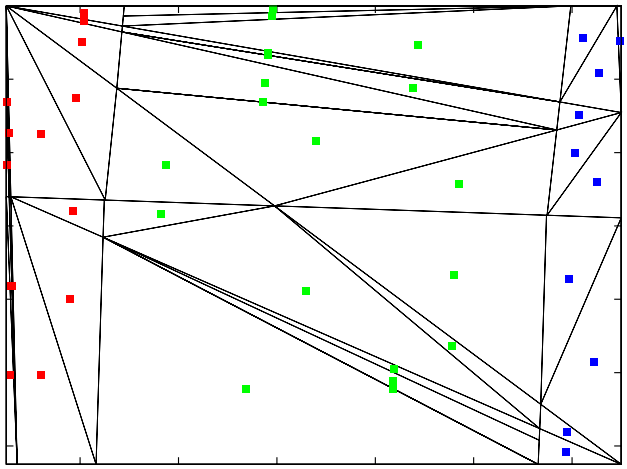

Figure 3.11: The conglomeration procedure is performed in the cell of interest, given the material configuration in Figure 3.10. Adjacent regions of "twin" materials are grouped together under conglomeration. Disconnected regions of the same material are relabeled as the primary material ID (red) and its "twin" ID (blue). The filamentary region is seen in green. Volume fractions and moments are calculated for the three materials.

3.3.2 Adjacency

Consider two triangles T_i and T_j with sides represented by line segments $S_{1,i}$, $S_{1,j}$, $S_{2,i}$, etc. For the purpose of conglomeration, triangles T_i and T_j are considered "adjacent" if for some pair

of sides $S_{k_1,i}$ and $S_{k_2,j}$, (3.66) holds, for some open set (a, b).

$$(a, b) \subset S_{k_1,i} \cap S_{k_2,j} \tag{3.66}$$

In essence, triangles should not be conglomerated in the case that the intersection of their sides is the empty set or a single point. Numerically, we cannot directly compare values to test for adjacency of sides, due to round-off errors that are inherent to calculation with real numbers. Instead, the sides are taken as vectors. If the cross-product between sides (3.67) falls below a tolerance, the sides are said to be parallel.

$$||\vec{S}_{k_1,i} \times \vec{S}_{k_2,j}||_\infty \leq h^2\epsilon \tag{3.67}$$

If sides are found to be parallel, then the projection of the endpoints of one side onto the linear extension of the other side is found. If the difference between the endpoint and the projection of the endpoint falls below a certain tolerance (3.68), the sides are said to be co-linear. Finally, co-linear points can be parametrized, so that one may find the intersection of the two line segments. If the intersection is some open set, the triangles are said to be adjacent.

$$||\boldsymbol{x} - \boldsymbol{x}_{project}||_1 \leq h\epsilon \tag{3.68}$$

In all of these simulations, $h = \Delta x$ and $\epsilon = 10^{-10}$.

3.3.3 Conglomerate Configuration

Once all conglomerates have been found, several situations can exist.

1. There is only one conglomerate, i.e. the cell is a pure cell with only one material.
2. There are two conglomerates with different material ID's. In this case, the algorithm reverts to the two material MOF interface reconstruction.
3. There are multiple conglomerates with different material ID's. In this case, a decision must be made about interface topology.

In the case of a pure cell, there is nothing to be done in terms of either conglomeration or interface reconstruction because no interface is present. This can be identified by preprocessing during the conglomeration routine to improve performance. If all triangular elements have the same material ID, perform no conglomeration; write the input ID's to the output and continue. In the

case of one conglomerate of each material, the cell is cut by a single interface. This is the situation for which Moment-of-Fluid Interface Reconstruction was designed. The conglomeration algorithm will perform the grouping, discover that only one conglomerate of each material exists, then write the material ID's to output. The interface reconstruction will be identical to MOF because no twin material ID's are introduced. This is computationally less efficient because more work will be expended with no gain in accuracy.

In the final case, multiple conglomerates of a given material exist, meaning a filament may form. Let us define a filament as a material region that separates two like-material regions such that they are not adjacent. We will denote the material that is separated by the filament as the "bulk" material. Note that the labels filament and bulk make no assumption of relative size of the regions.

If a cell is cut by a filament with interfaces of zero curvature, we wish to be able to recover the material configuration exactly. With Multimaterial Moment of Fluid, it is possible to capture the interfaces within the tolerance of volume and centroid error in the reconstruction process. By assigning a twin third material ID to one of the disconnected bulk regions, the procedure of nested dissection can be used to recover the two interfaces. As with a true multimaterial case, it is necessary to choose the ordering that minimizes reconstruction error. If the filament material is selected first for reconstruction, the resulting material configuration may differ significantly from the true material configuration. While the optimal ordering for reconstruction is not known *a priori*, experiment shows that the conglomerate with the centroid farthest from the cell center is less likely to be the filamentary material. During the procedure of nested dissection, the material to be reconstructed is chosen as the material whose centroid is farthest from the uncaptured region. Shown in Fig. 3.2, improper material ordering can result in improper interface topology, changing from a filamentary configuration to a T-junction or triple point.

3.3.4 Disconnected Conglomerates

A common occurrence in numerical simulations of deforming interfaces is the formation of flotsam, small regions of material in a cell of a different material. Flotsam, typically indicative of error in the advection procedure or interface reconstruction, can separate from an interface and merge at a later time. Define flotsam as a material conglomerate such that no vertex of the conglomerate lies on the cell boundary. By this definition, flotsam cannot form a filament, as it

does not divide a cell into disconnected regions. There exist several possible scenarios with regards to flotsam.

1. The flotsam is the only conglomerate of its material type in the cell.
2. There exists one other conglomerate of the same material type as the flotsam in the cell.
3. There are multiple conglomerates of the same material type as the flotsam in the cell.

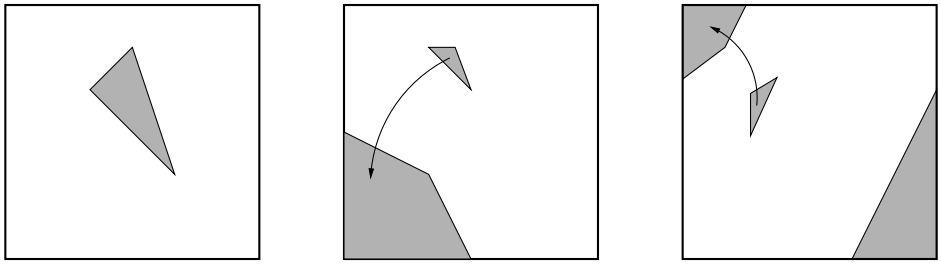

Figure 3.12: Left: Case 1. Flotsam in a cell. Middle: Case 2. Flotsam and one other like-material conglomerate in a cell. Right: Case 3. Flotsam with multiple material conglomerates in a cell. Arrow indicates inclusion in volume/moment calculation.

In the first case, only two material conglomerates are detected, so a filament does not form. The standard MOF interface reconstruction is used. In the second case, three material conglomerates are detected. However, logic in the conglomeration routine detects that one of the conglomerates is flotsam. The flotsam is included in the volume and moment calculation of the other like-material conglomerate, and the standard, two-material MOF reconstruction is used. In the final case, a filament may form. Flotsam is included in the volume of the "nearest" like-material conglomerate, according to distance between conglomerate centroids. In the last two scenarios, the location and orientation of the interface will be affected proportionally by the volume and location of the flotsam.

3.3.5 General Conglomerate Algorithm

Consider now the general case, with multiple conglomerates of two materials. We seek to identify "optimal" groupings in the sense that the conglomerates formed can best be represented by one or two interfaces. The full conglomeration algorithm is presented.

1. Form all Material 1 and Material 2 conglomerates: 1, 2, $1'$, $2'$, $1''$, $2''$, $1'''$, etc.

2. Choose at most the two largest conglomerates of each material that are adjacent to a cell edge. These are the candidates to be the bulk material. If one material has no conglomerates adjacent to the cell edge (i.e. all conglomerates are interior to the cell), use the standard MOF reconstruction and skip to the final step.

3. Attach all remaining conglomerates to the nearest like-material bulk candidate.

4. Introduce twin material ID's to account for any material with more than one conglomerate. Up to 4 materials may exist in a cell: 1, 2, $1'$, $2'$. Primes denote a twin material ID, an alternative material label for the same material.

5. Compute reference volumes and moments.

6. If two conglomerates of each material exist, then two separate interface reconstructions must be performed; continue to the next step. Otherwise, perform an interface reconstruction as in Multimaterial MOF and skip to the final step.

7. Perform an interface reconstruction assuming that Material 1 is the bulk material and Material 2 is the filament. Perform another reconstruction assuming that Material 2 is the bulk material and Material 1 is the filament.

8. Accept the interface reconstruction that minimizes centroid error.

Assigning remaining conglomerates to the "nearest" bulk candidate is performed based on distances between conglomerate centroids. The secondary twin fluid labels allow for differentiation between disconnected conglomerates of the same material. Multimaterial MOF will interpret them as different fluids and perform the dissection and reconstruction procedure accordingly. This is different from the procedure of standard MOF. During computation of volumes and moments of the materials in the cell in the standard Moment-of-Fluid procedure, only the material ID is taken into account when grouping polygonal elements. With no respect to the adjacency of like-material elements in a cell, all polygonal elements of the same material will be grouped into a single conglomerate, to use the language of this paper. In a two material simulation, this will result in two conglomerates in all mixed cells, regardless of the topology of the material configuration.

If a filament forms and generates a new, twin material ID, this does not prevent material from reforming at a later time step to a single material, eliminating the twin material ID. Prior to the detection of pure cells, a preprocessing step is carried out; all secondary, twin material ID's are replaced by the primary material ID before the conglomeration procedure begins. Grouping does

not differentiate between a material and its twin counterpart that was assigned a secondary, twin label at a previous time step. Physically, they represent the same material, and they should be grouped together if adjacent.

3.3.6 Adaptive Grid Initialization & Dynamic Regridding

It is possible to capture a filament by introducing a twin material and performing a MOF interface reconstruction for three materials. However, the case when the tip of a filament enters a cell is best captured using Adaptive Mesh Refinement (AMR). Adaptive Mesh Refinement uses the error in the reconstruction as a metric to refine the grid. To accurately capture the material configuration near the tip of a filament, a modification to the AMR procedure is used.

Conglomeration is performed in each cell at time $t = 0$, and the interface reconstruction is performed. The MOF error in each cell is computed over all materials m. The MOF error accounts for material volume (V^m) and the difference between the input, material reference centroid ($\boldsymbol{x}^m_{ref}$) and the actual reconstructed material centroid ($\boldsymbol{x}^m_{act}$).

$$E_{MOF} = \sum_m V^m ||\boldsymbol{x}^m_{ref} - \boldsymbol{x}^m_{act}|| \tag{3.69}$$

For each cell, a 3×3 stencil, or "super cell," is then considered. See Figure 3.13 - 3.14. The conglomeration procedure and interface reconstruction are carried out again over the super cell. The error over this "super cell" (E^S_{MOF}) is recomputed, as in (3.70).

$$E^S_{MOF} = \sum_m V^{m,S} ||\boldsymbol{x}^{m,S}_{ref} - \boldsymbol{x}^{m,S}_{act}|| \tag{3.70}$$

If the error in the "super cell" exceeds the error tolerance and the maximum level of refinement has not been reached, the cell is tagged for refinement. Additionally, cells in a narrow buffer zone are tagged for refinement. Tagged cells, along with cells in the buffer zone, are collected to form the refined grid. This procedure is then repeated on the refined grid, recomputing the reconstructed interface on the fine mesh and recalculating the MOF error.

Near the tip of a filament, error in a cell in the initial grid may be small, but error in the "super cell" will be large. This is due to the fact that in the 3×3 enlarged cell, the tip of the filament does not divide the cell into three distinct regions. During the conglomeration procedure, only two distinct material regions will be detected, so no filament will be introduced. This results in a large MOF error. Mesh refinement is essential to accurately capture the interface as the tip of the filament enters a new cell.

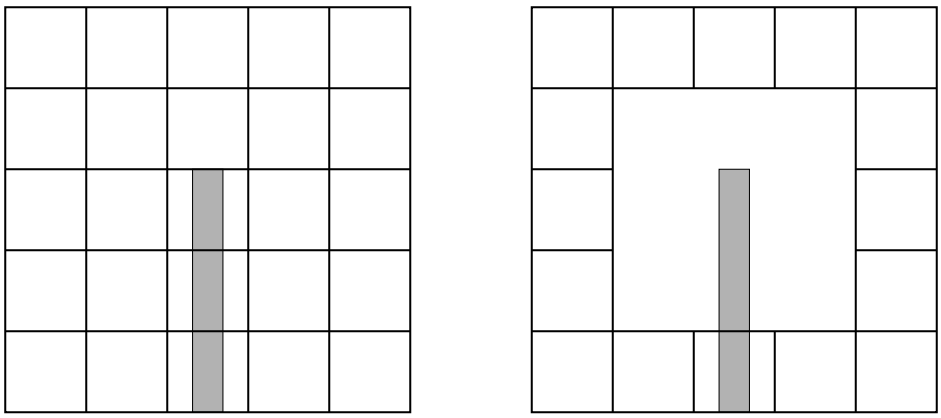

Figure 3.13: Left: A 5×5 grid with the tip of a filament. Filament material is shaded; bulk material is in white. Right: The 3×3 "super cell," shown embedded in the grid, for the central cell in the 5×5 mesh. The enlarged cell will yield only two material conglomerates, resulting in a large reconstruction error E^S_{MOF}.

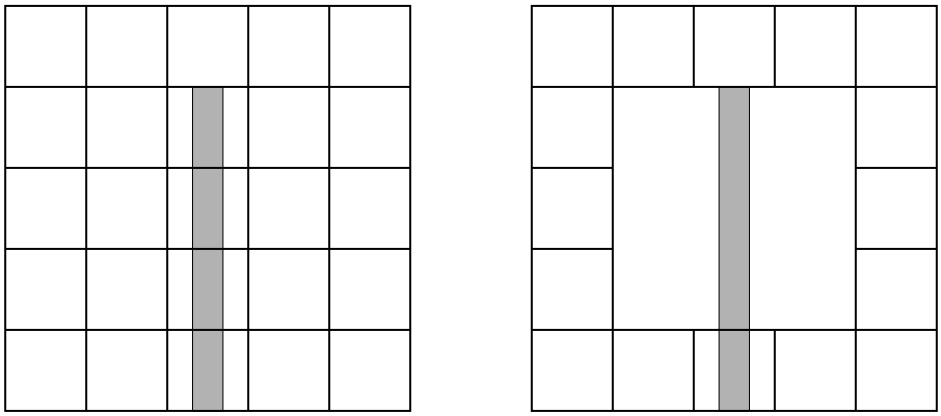

Figure 3.14: Left: A 5×5 grid with the tip of a filament. Right: The 3×3 "super cell," shown embedded in the grid. The enlarged cell will yield three material conglomerates. This results in a low reconstruction error E^S_{MOF} because the filament is well-resolved.

3.3.6.1 Coarse-To-Fine Data Interpolation. Where fine level grid cells are adjacent to coarse level grid cells, fine level "ghost cells" are formed using coarse level data. Using the reconstructed interface Γ_i from the coarse cell, the interface is reconstructed in the fine level ghost cells by cutting each fine level cell with Γ_i (Fig. 3.16)

3.3.6.2 Data Synchronization. Data is synchronized between fine levels and coarse levels using the following procedure, starting at the finest level.

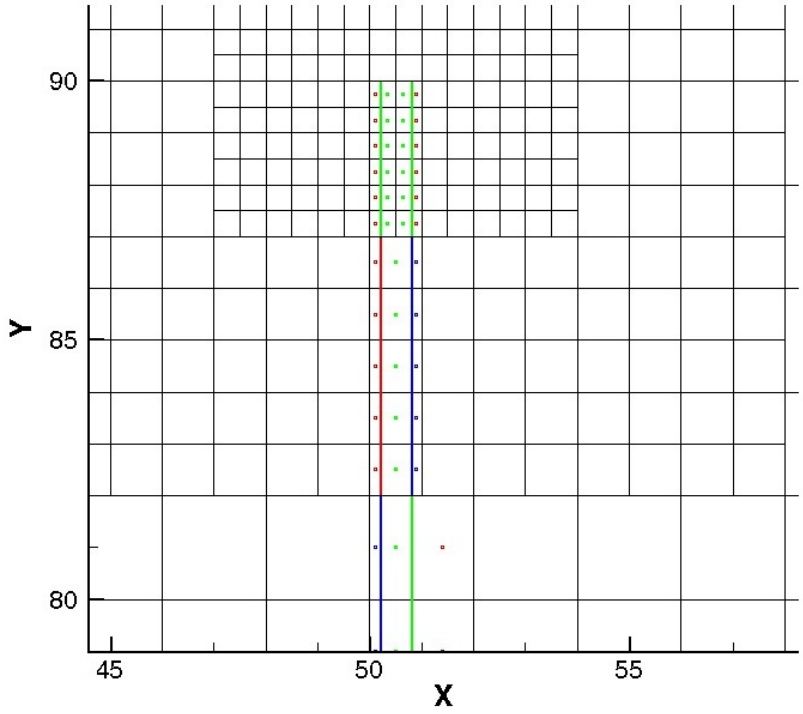

Figure 3.15: Filament tip with two level of Adaptive Mesh Refinement. Cells are refined when error in the 3×3 super cell exceeds the AMR error tolerance. Additionally, cells in a narrow band about a refined cell are tagged for refinement as well. Centroids and interfaces are shown for Material 1 (red), Material 2 (green), and twin Material $1'$ (blue).

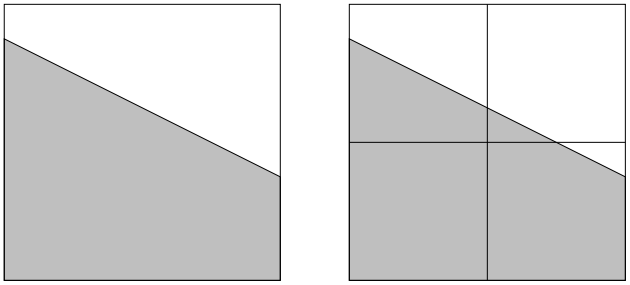

Figure 3.16: Left: Coarse cell cut by an reconstructed interface Γ_i. Right: Coarse cell data interpolated to fine cells. Each fine cell is cut by Γ_i.

1. Compute the slope reconstruction.

2. If the level is less than the maximum level of refinement, use the "average down" procedure (Fig. 3.17 - 3.18).

3. Compute the MOF error in conjunction with the conglomeration algorithm to account for filaments.

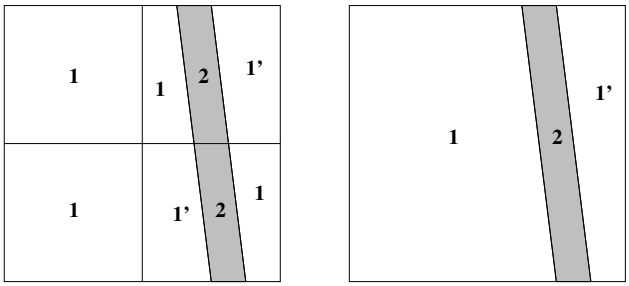

Figure 3.17: Left: A 2×2 stencil of fine level cells cut by a filament. Right: The corresponding coarse cell with resolved filament. The conglomeration procedure used in synchronizing fine to coarse data will group regions of Material 1 and its twin counterpart, Material $1'$.

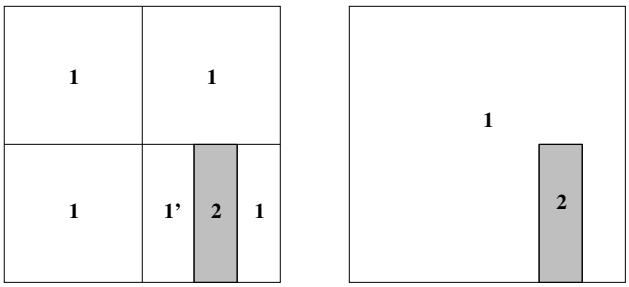

Figure 3.18: Left: A 2×2 stencil of fine level cells cut by a filament. Right: The corresponding coarse cell with unresolved filament. The conglomeration procedure at the coarse level will detect only two conglomerates. No filament will be introduced, leading to a large reconstruction error in the coarse cell.

3.3.7 Time Stepping

After the material configuration on the mesh has been initialized, the procedure of advection and interface reconstruction can be carried out.

1. Multimaterial slope reconstruction is performed.
2. The Cell-Integrated Semi-Lagrangian procedure advects material. When the departure region for a fine-level cell intersects a coarse-level cell, coarse-level data is interpolated to fine-level data, as in Section 3.3.6.1. Filaments in the advective preimage of a cell are detected and resolved using the conglomeration and reconstruction procedure.

3. Data is synchronized between fine levels and coarse levels, as in Section 3.3.6.2.

4. The refined grid is dynamically regridded.

   (a) Starting at the coarsest level L, tag cells with MOF error exceeding the threshold.

   (b) Group tagged cells, along with cells in the error buffer, to generate the next finest grid level, $L + 1$.

   (c) Generate the data on the new grid level, $L+1$. Either (i) interpolate data from a coarser level, or (ii) copy data from the previously existing level $L + 1$ where available. Return back to step (a) if the user defined maximum level has not been reached.

CHAPTER 4

NUMERICAL FLUID ALGORITHM

This is an introductory paragraph. Paragraphs in LaTeX are indicated in the document source by leaving a blank line between them. This is the last sentence of the paragraph.

And a second paragraph begins (and ends) here.

4.1 Pressure Evolution Equation

In this work, Euler's equations for multispecies flow are solved.

$$\begin{pmatrix} F^m \\ \rho^m F^m \\ \rho \boldsymbol{u} \\ \rho^m E^m F^m \end{pmatrix}_t + \nabla \cdot \begin{pmatrix} F^m \boldsymbol{u} \\ \rho^m F^m \boldsymbol{u} \\ \boldsymbol{u} \otimes \rho \boldsymbol{u} + p\mathbb{I} \\ (\rho^m E^m + p)\, \boldsymbol{u} \end{pmatrix} = \begin{pmatrix} F^m \nabla \cdot \boldsymbol{u} \\ 0 \\ \boldsymbol{f}_{body} \\ \boldsymbol{u} \cdot \boldsymbol{f}_{body} F^m \end{pmatrix} \tag{4.1}$$

F^m is the volume fraction of material m, ρ^m is the density for material m, $\rho^m E^m$ is the energy for material m, and $\boldsymbol{u}$ is the velocity vector. For this work, it is assumed that there are no energy source term, i.e. from Section 2.3, $\dot{q} \equiv 0$. Define the specific internal energy e_{int} such that

$$\rho^m E^m = \rho^m e^m_{int} + \frac{1}{2}\rho^m |\boldsymbol{u}|^2.$$

Here, $\boldsymbol{f}_{body}$ represents the body force. The system is closed by defining the pressure evolution equation

$$p_t + \boldsymbol{u} \cdot \nabla p = -\rho c^2 \nabla \cdot \boldsymbol{u}. \tag{4.2}$$

This equation can be derived, as in [23] by assuming that pressure is some general function of density and internal energy of a given material, $p \equiv p(\rho, e_{int})$. Therefore, taking the material derivative of both sides,

$$\frac{Dp}{Dt} = \frac{\partial p}{\partial \rho}\frac{D\rho}{Dt} + \frac{\partial p}{\partial e_{int}}\frac{De_{int}}{Dt}. \tag{4.3}$$

Here, $\frac{D\varphi}{Dt} = \frac{\partial \varphi}{\partial t} + \boldsymbol{u} \cdot \nabla \varphi$ is the material derivative. The following components of Euler's equations, written in non-conservative form, are then used:

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho = -\rho \nabla \cdot \boldsymbol{u}, \tag{4.4}$$

$$\frac{Du}{Dt} = \frac{\partial u}{\partial t} + \boldsymbol{u} \cdot \nabla u = -\frac{p_x}{\rho}, \tag{4.5}$$

$$\frac{Dv}{Dt} = \frac{\partial v}{\partial t} + \boldsymbol{u} \cdot \nabla v = -\frac{p_y}{\rho}, \tag{4.6}$$

$$\frac{De_{int}}{Dt} = \frac{\partial e_{int}}{\partial t} + \boldsymbol{u} \cdot \nabla e_{int} = -\frac{p}{\rho} \nabla \cdot \boldsymbol{u}, \tag{4.7}$$

and the expression for the speed of sound,

$$c^2 = \frac{\partial p}{\partial \rho} + \frac{p}{\rho^2} \frac{\partial p}{\partial e_{int}}. \tag{4.8}$$

Substituting (4.4 - 4.7) into (4.3), we recover

$$\frac{Dp}{Dt} = (-\rho \nabla \cdot \boldsymbol{u}) \frac{\partial p}{\partial \rho} + (-\frac{p}{\rho} \nabla \cdot \boldsymbol{u}) \frac{\partial p}{\partial e_{int}} = -\rho \nabla \cdot \boldsymbol{u} \left(\frac{\partial p}{\partial \rho} + \frac{p}{\rho^2} \frac{\partial p}{\partial e_{int}} \right). \tag{4.9}$$

Notice that the right-hand side of (4.9) can be rewritten as $-\rho c^2 \nabla \cdot \boldsymbol{u}$ to recover the pressure evolution equation (4.10).

$$\frac{Dp}{Dt} = p_t + \boldsymbol{u} \cdot \nabla p = -\rho c^2 \nabla \cdot \boldsymbol{u} \tag{4.10}$$

Note that the expression for specific internal energy (4.7) can be derived using the definition of total specific energy ($E = e_{int} + \frac{||\boldsymbol{u}||^2}{2}$) and expressions (4.4 - 4.6).

4.2 1D All-Speed Finite Volume Method

The 1D Finite Volume Method for discretizing the all-speed pressure projection for multiphase flow is done in much the same way as the original work of Kwatra, et. al. [44]. The momentum equation is used to derive an implicit equation for pressure, taking the divergence of (4.11) to obtain (4.12).

$$u^{n+1} = u^a - \Delta t \frac{\nabla p^{n+1}}{\rho^{n+1}} \tag{4.11}$$

$$\nabla \cdot u^{n+1} = \nabla \cdot u^a - \Delta t \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) \tag{4.12}$$

To model incompressible flow, we would set $\nabla \cdot \boldsymbol{u}^{n+1} = 0$. However, utilizing the pressure evolution equation derived in [23], a method that preserves the incompressible pressure projection method in the limit of infinite sound speed is recovered. Fixing the right-hand side term $\nabla \cdot \boldsymbol{u}$ at time t^{n+1} in (4.13) and substituting (4.12) in the pressure evolution equation, we recover the semi-discretized equation (4.14).

$$\frac{\partial p}{\partial t} + \boldsymbol{u} \cdot \nabla p = -\rho c^2 \nabla \cdot \boldsymbol{u} \tag{4.13}$$

$$\frac{\partial p}{\partial t} + \boldsymbol{u} \cdot \nabla p = -\rho c^2 \nabla \cdot \boldsymbol{u}^a + \rho c^2 \Delta t \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right). \tag{4.14}$$

The time discretization is completed by fixing (ρc^2) to be at time t^{n+1} and discretizing the advective pressure term in an explicit fashion, using a forward Euler method

$$\frac{p^a - p^n}{\Delta t} + \boldsymbol{u}^n \cdot \nabla p^n = 0.$$

$$p^a = p^n - (\boldsymbol{u}^n \cdot \nabla p^n) \Delta t \tag{4.15}$$

An $\mathcal{O}(\Delta t)$ consistency error is made by fixing $\nabla \cdot \boldsymbol{u}$ to be at time level t^{n+1}. This is observed by substituting (4.12) into the continuous relation (4.10). The expression for advective pressure (4.15) is substituted in (4.14) and the result is divided by $(\rho c^2)^{n+1}$ to recover the discretized equation

$$\frac{p^{n+1}}{(\rho c^2)^{n+1}} - \Delta t^2 \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{p_i^a}{(\rho c^2)^{n+1}} - \Delta t \nabla \cdot \boldsymbol{u}^a. \tag{4.16}$$

Unlike the work of [44], the advective pressure term p^a is discretized using an equation of state under the sharp interface paradigm, rather than using a linear combination, Riemann solver, or Runge-Kutta advection strategy. Consider materials $m = 1, 2, \ldots, M$, each with equation of state $p = EOS^m(\rho_i^m, e_{int,i}^m)$ and squared sound speed $c^2 = SOS(\rho^m, e_{int}^m)$ occupying a cell. Let $m^\star$ be the

material that occupies the maximum volume of a cell Ω_i. Then, the advective pressure and sound speed are discretized as

$$p_i^a = EOS(\rho^{m^\star}, e_{int,i}^{m^\star})$$
$$c_i^2 = SOS(\rho^{m^\star}, e_{int,i}^{m^\star}).$$

Use of an equation of state was found to produce less oscillatory results than the Runge-Kutta methods used in [44] in the presence of strong shocks and large density ratios.

The Finite Volume discretization of the pressure equation proceeds in the standard fashion. Data is located at cell centers, x_i, and gradients exist at cell faces $x_{i\pm1/2}$. Equation (4.16) is integrated over each cell Ω_i. The pressure gradient term is discretized using a centered difference scheme, and Green's Theorem is used to discretize the divergence term. Finally, the result is divided by the cell volume. Here, a uniform mesh spacing is assumed (i.e. for all cells, $|\Omega_i| = |x_{i+1/2} - x_{i-1/2}| = \Delta x$).

$$\frac{p_i^{n+1}}{(\rho_* c^2)_i^{n+1}} - \Delta t^2 \left(\frac{\left(\frac{p_{i+1}^{n+1} - p_i^{n+1}}{\rho_{i+1/2,*}^{n+1} \Delta x} \right) - \left(\frac{p_i^{n+1} - p_{i-1}^{n+1}}{\rho_{i-1/2,*}^{n+1} \Delta x} \right)}{\Delta x} \right) = \frac{p_i^a}{(\rho_* c^2)_i^{n+1}} - \Delta t \frac{u_{i+1/2}^a - u_{i-1/2}^a}{\Delta x} \quad (4.17)$$

In this work, let $(\rho c^2)^{n+1} = \rho^{n+1} c^2(\rho^{n+1}, e_{int}^a)$. The value $\rho_{i,*}$ is taken to be the total mass in the cell Ω_i divided by the volume of the cell. If each material m occupies the region Ω_i^m then

$$\rho_{i,*} = \frac{\sum_{m=1}^{M} \int_{\Omega_i^m} \rho_i^m d\Omega}{|\Omega_i|}.$$

Just as the centered difference in the pressure gradient is discretized over the face-centered control volume, the face-centered density $\rho_{i\pm1/2,*}$ is evaluated over the face-centered control volume

$$\Omega_{i+1/2} = [x_i, x_{i+1}].$$

Mass for each of m materials in the right-half of cell Ω_i and left-half of cell Ω_{i+1} is calculated, with material subregions denoted $\Omega_{i,R}$ and $\Omega_{i+1,L}$ respectively (see Fig. 4.1). The face density is then calculated as the mass in the cell face control volume divided by the volume of the face control volume,

$$\rho_{i+1/2,*} = \frac{\sum_{m=1}^{M} \int_{\Omega_{i,R}^m} \rho_i^m d\Omega + \sum_{m=1}^{M} \int_{\Omega_{i+1,L}^m} \rho_{i+1}^m d\Omega}{|\Omega_{i+1/2}|}. \quad (4.18)$$

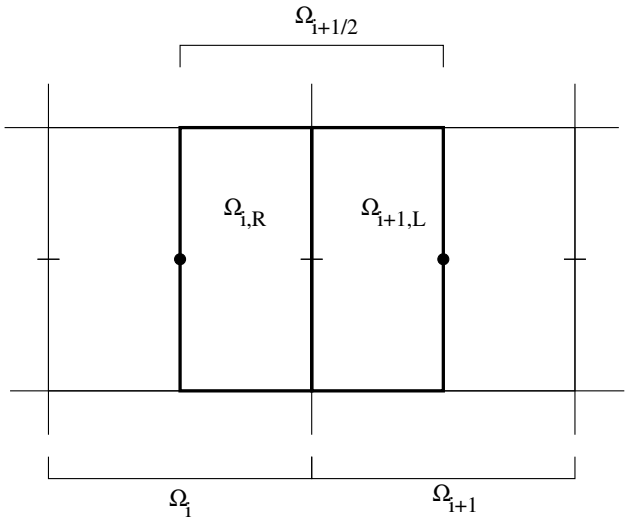

Figure 4.1: Cells Ω_i and Ω_{i+1} with face-centered control volume $\Omega_{i+1/2}$ and half-cell regions $\Omega_{i,R}$ and $\Omega_{i+1,L}$.

This is contrary to the density averaging scheme in [44]. It is found that for large density ratios, as in the Oscillating Water Column test, use of a dual-cell averaged density, i.e.

$$\rho_{i+1/2} = \frac{\rho_{i,*} + \rho_{i+1,*}}{2}$$

produces irregularities at the material interface. By instead choosing the discretization (4.18), the fully discrete pressure equation (4.17) is conservative over the entire domain.

It should be noted that this discretization of the all-speed pressure equation extends naturally to multiple dimensions, as in [38]. In two dimensions, the face-centered control volumes become $\Omega_{i+1/2,j} = [x_i, x_{i+1}] \times [y_{j-1/2}, y_{j+1/2}]$ and $\Omega_{i,j+1/2} = [x_{i-1/2}, x_{i+1/2}] \times [y_j, y_{j+1}]$. Similarly for the half-face control volumes $\Omega_{i,j,L/R}$, and the new top/bottom half-face control volumes $\Omega_{i,j,T/B}$. In three dimensions, there are 6 half-face regions, two each in the $x-$ and $y-$directions, plus two more in the $z-$direction.

4.2.1 Equation of State Time Level Consistency

An equation of state is used to calculate the advected pressure, that appears on the right-hand side of (4.16), as a function of density and specific internal energy. A decision must be made about the time level of the density and specific internal energy used to evaluate the equation of state. Through a Taylor series expansion argument, it can be shown that for smooth solutions in an ideal

gas, it is valid to evaluate the density as the advected density ρ^a (3.55) or the time advanced density ρ^{n+1} (3.56).

Discretize (3.55 - 3.56) respectively as

$$\frac{\rho^a - \rho^n}{\Delta t} + \boldsymbol{u}^n \cdot \nabla \rho^n = 0 \tag{4.19}$$

and

$$\frac{\rho^{n+1} - \rho^a}{\Delta t} + \nabla \cdot (\boldsymbol{u}^n \rho^n) = 0 \tag{4.20}$$

Then, we can rewrite (4.20) in non-conservative form as

$$\frac{\rho^{n+1} - \rho^a}{\Delta t} + \boldsymbol{u}^n \cdot \nabla \rho^n = -\rho^n \nabla \cdot \boldsymbol{u}^n. \tag{4.21}$$

Thus, we can write the ρ^{n+1} in terms of ρ^a, with some perturbation,

$$\rho^{n+1} = \rho^a - \Delta t \rho^n \nabla \cdot \boldsymbol{u}^n. \tag{4.22}$$

Consider the ideal gas equation of state, derived from thermodynamics relations in Section 2.4,

$$p(\rho, e_{int}) = (\gamma - 1)\rho e_{int}. \tag{4.23}$$

The advected pressure is a function of the advected density and specific internal energy, $p^a = (\gamma - 1)\rho^a e^a_{int}$. If we instead use the density at time level t^{n+1}, we recover the following,

$$p^a = (\gamma - 1)(\rho^a - \Delta t \rho^n \nabla \cdot \boldsymbol{u}^n) e^a_{int}. \tag{4.24}$$

In the case that $\rho^n \nabla \cdot \boldsymbol{u}^n e^a_{int}$ is small, then the consistency error made in (4.13) by evaluating the equation of state using the new density is of $\mathcal{O}(\Delta t)$. This is of the same order error that was made by assuming that the velocity term on the right-hand side of (4.13) is fixed at time level t^{n+1}. In other words, for flows with low density and energy or flows that are nearly divergence free, it introduces no larger order consistency error to eliminate the variable ρ^a and simply use ρ^{n+1}.

In practical application, use of the non-conservatively transported density ρ^a can be less robust, when slope limiting (i.e. linear reconstruction) of density is performed, compared to using ρ^{n+1} to evaluate the equation of state. In some applications, it is observed that use of the advective density ρ^a to evaluate advected pressure can result in a negative specific internal energy e_{int} at time

level t^{n+1}, when given a positive specific internal energy at time level t^n. The positivity preserving aspects of using ρ^{n+1} to evaluate the advected pressure has not yet been fully explored and is an area for future research.

It is noted that (4.24) seems to imply a strict criterion for maintaining consistency in the pressure evolution equation. However, use of ρ^{n+1} to evaluate the equation of state still produces results that converge with nearly first order accuracy for various shock tube tests with analytical solutions, including strong shocks (Section 5.1). Similarly for the Underwater Explosion test (Section 5.8), comparison of results using ρ^{n+1} to evaluate the EOS compare favorably for shock speed and magnitude, as well as bubble location, to previously published Lagrangian results [83]. For these same tests, the use of ρ^a to evaluate the EOS combined with MINMOD slope limiting produces severe ringing at the shock front.

4.2.2 State Variable Update

Note that from the CISL advection procedure and the pressure evolution equation, that velocity and pressure are collocated at cell centers. To update the cell-centered velocity using the momentum equation, a face-centered pressure must be defined. This method proceeds as in [44], with similar changes as before to account for material discontinuity. First, the momentum equation is applied in each of the half-cell control volumes surrounding the face located at $x_{i+1/2}$.

$$\frac{du_{i,R}}{dt} = \frac{p_{i+1/2} - p_i}{\rho_{i,R,*}\Delta x}$$
$$\frac{du_{i+1,R}}{dt} = \frac{p_{i+1} - p_{i+1/2}}{\rho_{i+1,L,*}\Delta x}$$

In the framework of Lagrangian advection, adjoining cells must remain in constant contact, otherwise a vacuum is created. This "constant contact" condition is applied by setting

$$\frac{du_{i,R}}{dt} = \frac{du_{i+1,R}}{dt}$$

and solving for the unknown $p_{i+1/2}$. Thus face pressures are given as

$$\begin{aligned} p_{i+1/2} &= \frac{\rho_{i+1,L,*}p_i + \rho_{i,R,*}p_{i+1}}{\rho_{i,R,*} + \rho_{i+1,L,*}} \\ p_{i-1/2} &= \frac{\rho_{i,L,*}p_{i-1} + \rho_{i-1,R,*}p_i}{\rho_{i-1,R,*} + \rho_{i,L,*}}. \end{aligned} \tag{4.25}$$

The half-cell densities are defined, similar to before, as the amount of mass of all materials m in the region $\Omega_{i,L/R}$ divided by the magnitude of the half-cell control volume. Note that the cell density

is defined as a function of x, to allow for some reconstruction of the density profile in the cell. It is also noted that an alternate physical interpretation of the "constant contact" condition is that we are solving for the pressure at the cell face such that flux at the face is of equal magnitude and opposite sign, when viewed with respect to the two cells that share the given face. This condition will reappear in the 2D Support Operator Finite Volume method.

$$\rho_{i,R,*} = \frac{\sum_{m=1}^{M} \int_{\Omega_{i,R}^m} \rho_i^m d\Omega}{|\Omega_{i,R}|}$$
$$\rho_{i,L,*} = \frac{\sum_{m=1}^{M} \int_{\Omega_{i,L}^m} \rho_i^m d\Omega}{|\Omega_{i,L}|}$$

The evolution of cell-centered velocity can then be discretized as

$$u_i^{n+1} = u_i^a - \Delta t \frac{p_{i+1/2}^{n+1} - p_{i-1/2}^{n+1}}{\rho_i^{n+1} \Delta x}. \tag{4.26}$$

Energy is updated in a similar fashion, although a face velocity must be specified. Face velocity is defined as the momentum conservative interpolation of velocity to the cell face. In this way, momentum on the cell-centered and face-centered grids are equal.

$$\begin{aligned} u_{i+1/2} &= \frac{\sum_{m=1}^{M} \left(\int_{\Omega_{i,R}^m} \rho_i^m u_i d\Omega + \int_{\Omega_{i+1,L}^m} \rho_{i+1}^m u_{i+1} d\Omega \right)}{|\Omega_{i+1/2}|} \\ u_{i-1/2} &= \frac{\sum_{m=1}^{M} \left(\int_{\Omega_{i-1,R}^m} \rho_{i-1}^m u_{i-1} d\Omega + \int_{\Omega_{i,L}^m} \rho_i^m u_i d\Omega \right)}{|\Omega_{i-1/2}|} \end{aligned} \tag{4.27}$$

Using the face pressure and face velocity, then the update of cell-centered energy for each material m is discretized as

$$E_i^{n+1,m} = \begin{cases} E^{a,m} - \Delta t \frac{p_{i+1/2}^{n+1} u_{i+1/2}^{n+1} - p_{i-1/2}^{n+1} u_{i-1/2}^{n+1}}{\rho_i^{n+1,m} \Delta x}, & |\Omega_i^m| > 0 \\ 0, & \text{otherwise} \end{cases}. \tag{4.28}$$

4.3 2D Unstructured All-Speed Support Operator Finite Volume Method

The Support Operator Method (SOM) formulation of the diffusion operator on unstructured triangular grids, developed in [12], is used in the discretization of the pressure equation for incompressible flow tests. An unstructured, cell-centered gradient operator, consistent with the diffusion operator, is developed as well. A brief summary of the method follows.

4.3.1 Support Operator Method Diffusion Operator Formulation

Let Ω be a closed subset of $\mathbb{R}^2$ and $p \equiv p(\boldsymbol{x}, t)$ be the solution to the general diffusion equation (4.29), with diffusion coefficient D, general forcing term $f(\boldsymbol{x})$, and initial condition p^0.

$$\begin{cases} \frac{\partial p}{\partial t} - \nabla \cdot (D \nabla p) = f(\boldsymbol{x}), & \boldsymbol{x} \in \Omega \\ p(\boldsymbol{x}, 0) = p^0(\boldsymbol{x}) & \\ p(\boldsymbol{x}, t) = g_1(\boldsymbol{x}), & \boldsymbol{x} \in \partial\Omega_D \\ \frac{\partial p(\boldsymbol{x}, t)}{\partial n} = g_2(\boldsymbol{x}), & \boldsymbol{x} \in \partial\Omega_N \end{cases} \tag{4.29}$$

Let $\boldsymbol{\Phi} = -D\nabla p$ denote the diffusion flux. In [12], boundary conditions of both Dirichlet and Neumann type are discussed. In this work, the method for applying periodic boundary conditions will be studied as well. The SOM diffusion scheme assumes that unknowns are located at cell centers, with value p_i (4.30) defined as the mean value of $p(x, y, t)$ in cell Ω_i, i.e. a piecewise-constant basis function is used in each cell.

$$p_i(t) = \frac{1}{V_i} \int_{\Omega_i} p(x, y, t) d\Omega \tag{4.30}$$

Here, V_i denotes the volume of cell Ω_i. The vertices P_r of triangular cell Ω_i are assumed to be numbered in a counter-clockwise, periodic fashion (i.e. $P_4 = P_1$). Let C_i be the centroid of cell Ω_i. Let $L_{r,r+1}$ denote the length of the segment connecting vertex P_r and vertex P_{r+1}; let $P_{r+1/2}$ denote the midpoint of the segment; and let $\boldsymbol{N}_{r,r+1}$ be the outward unit normal to the segment, with outward defined relative to cell Ω_i. Integrate (4.29) over the cell Ω_i and apply Green's formula to obtain the integral form of the PDE (4.31).

$$\frac{dp_i}{dt} V_i + \int_{\partial\Omega_i} \boldsymbol{\Phi} \cdot \boldsymbol{N} dl = f_i V_i \tag{4.31}$$

The cell Ω_i is subdivided into nodal control regions Ω_r^i for each node P_r. The nodal control region is defined as the convex hull of centroid C_i, vertex P_r, and face centroids $P_{r+1/2}$ and $P_{r-1/2}$.

Two additional degrees of freedom per element face are introduced at half-face centroids. Let $\bar{p}_{r,r+1/2}$ and $\bar{p}_{r+1/2,r+1}$ denote degrees of freedom on the cell half-face associated with vertices P_r and P_{r+1} respectively, as in Figure 4.2 . These additional degrees of freedom will be used to enforce that the operator is second-order accurate on the unstructured mesh. Additionally, define half-face cell fluxes $\Phi^i_{r,r+1/2}$ and $\Phi^i_{r+1/2,r+1}$. Thus, the integral over the half-face regions can be written as

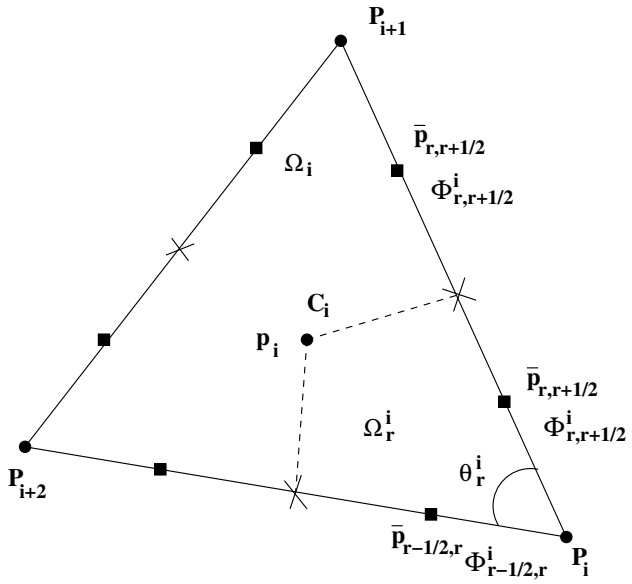

Figure 4.2: Triangular cell Ω_i is shown with data locations. Face centroids are shown as an "X" and half-face centroids, where half-face degrees of freedom and half-face fluxes live, are shown with a filled square. The cell nodal control volume Ω_r^i is bounded by a dotted region. Nodal angle θ_r^i is depicted as well.

(4.32 - 4.33).

$$\frac{1}{2}\Phi^i_{r,r+1/2} = \int_{P_r}^{P_{r+1/2}} \boldsymbol{\Phi} \cdot \boldsymbol{N} dl \tag{4.32}$$

$$\frac{1}{2}\Phi^i_{r+1/2,r+1} = \int_{P_{r+1/2}}^{P_{r+1}} \boldsymbol{\Phi} \cdot \boldsymbol{N} dl \tag{4.33}$$

The integral form of the PDE (4.31) can then be discretized in terms of nodes P_r (4.34).

$$\frac{dp_i}{dt} V_i + \sum_{r=1}^{3} \frac{1}{2} \left(L_{r-1,r} \Phi^i_{r-1/2,r} + L_{r,r+1} \Phi^i_{r,r+1/2} \right) = f_i V_i \tag{4.34}$$

The flux across each cell half-face in this control volume is written in terms of both the cell-centered and cell-face pressure values. By enforcing equality of the flux written in terms of cell-centered and face-centered pressure, we can write the face-centered pressure as a function of the cell-centered pressure. Thus, edge-fluxes can be written only in terms of cell-centered values.

Flux on the half-face on either side of the vertex r in cell Ω_i is equal to (after some algebra)

$$\begin{pmatrix} \Phi^i_{r-1/2,r} \\ \Phi^i_{r,r+1/2} \end{pmatrix} = -\frac{D_i}{2\omega^i_r} \begin{pmatrix} 1 & -\cos\theta^i_r \\ -\cos\theta^i_r & 1 \end{pmatrix} \begin{bmatrix} L_{r-1,r}\left(\bar{p}_{r-1/2,r} - p_i\right) \\ L_{r,r+1}\left(\bar{p}_{r,r+1/2} - p_i\right) \end{bmatrix}. \tag{4.35}$$

Angle θ^i_r is the angle in cell Ω_i associated with vertex r, $L_{r,r+1}$ is the length of the face between vertices r and $r+1$, D_i is the cell averaged diffusion weight in cell Ω_i, and $\omega^i_r = V_i/3$ is the interpolation weight that ensures exactness for linear functions.

Writing the corresponding flux relation for the adjacent cell, we then get the two equations defining flux across a shared half-face. Introducing the notation Φ^{k-1}_k to denote flux across the k^{th} edge as seen from the $k-1^{th}$ cell and Φ^k_k to denote flux across the k^{th} edge as seen from the k^{th} cell, we get the following expressions for flux in terms of cell-centered and face-centered values. Here, $\alpha_k = \frac{D_k}{\omega_k}$.

$$\Phi^{k-1}_k = -\frac{1}{2}\alpha_{k-1}\left[-\cos\theta_{k-1}L_{k-1}(\bar{p}_{k-1} - p_{k-1}) + L_k(\bar{p}_k - p_{k-1})\right] \tag{4.36}$$

$$\Phi^k_k = -\frac{1}{2}\alpha_k\left[L_k(\bar{p}_k - p_k) - \cos\theta_k L_{k+1}(\bar{p}_{k+1} - p_k)\right] \tag{4.37}$$

Note that the flux through face "k" as seen from cell Ω_{k-1} will be of equal magnitude and opposite sign as the flux through face "k" as seen from cell Ω_k, i.e. an outward flux for cell Ω_{k-1} will be an inward flux for cell Ω_k.

$$L_k\left(\Phi^k_k + \Phi^{k-1}_k\right) = 0 \tag{4.38}$$

Applying the flux condition (4.38) across each face for K cells sharing a node, we get K equations for K unknowns, $\bar{p}_k$.

$$\begin{aligned} -\alpha_{k-1}\cos\theta_{k-1}L_{k-1}L_k\bar{p}_{k-1} + (\alpha_{k-1}+\alpha_k)L^2_k\bar{p}_k - \alpha_k\cos\theta_k L_k L_{k+1}\bar{p}_{k+1} \\ = \alpha_{k-1}L_k(L_k - L_{k-1}\cos\theta_{k-1})p_{k-1} + \alpha_k L_k(L_k - L_{k+1}\cos\theta_{k+1})p_k \end{aligned} \tag{4.39}$$

This is a matrix equation that specifies that flux in terms of cell-centered values is equal to the flux in terms of face-centered values. This can be written as

$$M\overline{\boldsymbol{P}} = S\boldsymbol{P}. \tag{4.40}$$

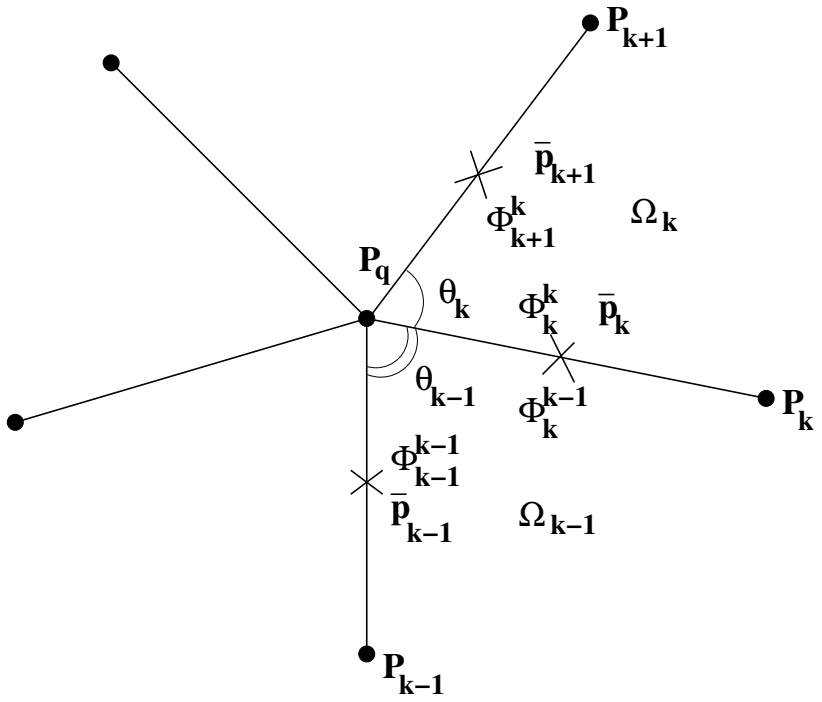

Figure 4.3: Data locations for node P_q. Nodes connected to P_q are given a local ordering k, $k+1$, etc. Local half-face degrees of freedom $\bar{p}$ and half-face fluxes Φ are shown, along with local angle θ_k. Face centroids are shown with an "X."

Because elements and vertices are labeled periodically, matrix M will have periodic tridiagonal non-zero structure. The "sub-diagonal" entry of the first row will appear in the last column of the first row, and the "super-diagonal" entry of the final row will appear in the first column of the final row. Similarly, matrix S will have periodic structure, with non-zero entries appearing on the diagonal and "sub-diagonal." It is worth noting that the entries in S can be written as

$$\begin{aligned} S_{k,k-1} &= \alpha_{k-1} L_k (L_k - L_{k-1} \cos\theta_{k-1}) \\ S_{k,k} &= \alpha_k L_k (L_k - L_{k+1} \cos\theta_k) \end{aligned} \tag{4.41}$$

Define $F = M^{-1}S$ as the matrix that expresses face-centered values as a function of cell-centered values, so that

$$\overline{\boldsymbol{P}} = M^{-1} S \boldsymbol{P} = F \boldsymbol{P}. \tag{4.42}$$

It is worth noting that (4.42) is the equivalent of the "constant contact" condition that was applied in the 1D Finite Volume method. Pressure at the cell face is solved in terms of the cell-centered

values such that flux condition (4.38) at the face is satisfied. Define P_q as the global node index of local node P_r. Thus, the contribution of internal node P_q to the flux in cell k can be written as

$$C_k = \frac{1}{2}\left(L_k\Phi_k^k + L_{k+1}\Phi_{k+1}^k\right). \tag{4.43}$$

Rewriting (4.35) in terms of the local cell flux terms.

$$\begin{pmatrix}\Phi_k^k\\ \Phi_{k+1}^k\end{pmatrix} = -\frac{\alpha_k}{2}\begin{pmatrix}1 & -\cos\theta_k\\ -\cos\theta_k & 1\end{pmatrix}\begin{bmatrix}L_k\left(\bar{p}_k - p_k\right)\\ L_{k+1}\left(\bar{p}_{k+1} - p_k\right)\end{bmatrix}, \tag{4.44}$$

the contribution of node P_q to flux in cell $q(k)$ can be written.

$$C_k = -\frac{\alpha_k}{4}\left[L_k\left(L_k - L_{k+1}\cos\theta_k\right)\left(\bar{p}_k - p_k\right) + L_{k+1}\left(L_{k+1} - L_k\cos\theta_k\right)\left(\bar{p}_{k+1} - p_k\right)\right] \tag{4.45}$$

Note that the entries from matrix S (4.41) appear in (4.45), which can be written as

$$C_k = -\frac{\alpha_k}{4}\left[\sum_{j=1}^{K} S_{i,j}^T(\bar{p}_j - p_k)\right] \tag{4.46}$$

due to the bidiagonal structure of S. Then, note that

$$\sum_{j=1}^{K} S_{k,j}^T p_k = (S_{k,k} + S_{k+1,k})p_k = \alpha_k(L_k^2 + L_{k+1}^2 - 2L_kL_{k+1}\cos\theta_k)p_k \tag{4.47}$$

So, we can define $\lambda_k^2 = L_k^2 + L_{k+1}^2 - 2L_kL_{k+1}\cos\theta_k$ and diagonal matrix Λ by

$$\Lambda_{k,l} = \alpha_k\lambda_k^2\delta_{k,l}. \tag{4.48}$$

Here, let $\delta_{k,l}$ be the Kronecker delta function. Using (4.47 - 4.48) along with the fact that $\overline{\boldsymbol{P}} = M^{-1}S\boldsymbol{P}$, we can rewrite (4.46) in matrix form for the vector $\boldsymbol{C}$ of local flux contributions,

$$\boldsymbol{C} = -\frac{1}{4}(S^TM^{-1}S - \Lambda)\boldsymbol{P}. \tag{4.49}$$

$$\boldsymbol{A} = -\frac{1}{4}(S^TM^{-1}S - \Lambda) \tag{4.50}$$

So, the local diffusion operator is defined by the matrix $\boldsymbol{A}$ for an internal node P_q. The global diffusion matrix is the sum of $\boldsymbol{A}$ over all nodes P_q. Given that the stencil for each node is the set of all elements that share the node, then the stencil for the diffusion operator for a given cell is the set of all elements that share any node with the element of interest (see Fig. 4.4). The derivation of the diffusion operator in this section addresses only nodes that lie in the interior of the domain, i.e. $P_q \notin \partial\Omega$. The derivation of the contribution of boundary nodes to the diffusion operator will be discussed in Section 4.3.2.

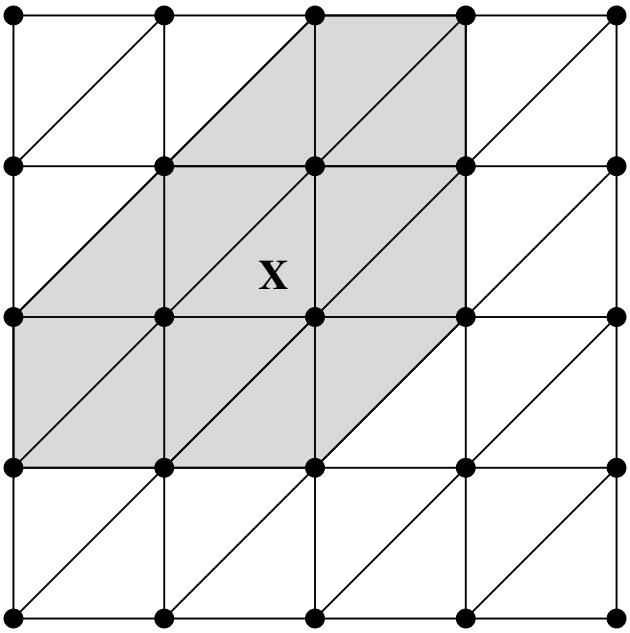

Figure 4.4: Triangulated mesh, with the computational stencil for the cell marked with an "X" shown as a shaded region.

4.3.2 Boundary Conditions

For the tests conducted in this work, Neumann boundary conditions and periodic boundary conditions are utilized. The original work of Breil and Maire [12] develops the implementation of Dirichlet and Neumann boundary conditions. The implementation of Periodic boundary conditions in the context of the diffusion operator and global face-projection operator is developed in this work. Suppose that there exist Q_e nodes such that $P_q \in \partial\Omega$, and let $\{P_q\}_{q=1}^{Q_e}$ denote the set of boundary nodes.

4.3.2.1 Neumann Boundary Conditions. Consider a node $P_q \in \partial\Omega_N$. Suppose that there are K cells and $K+1$ edges that share node P_q. Let $k \equiv k(q)$ denote the local ordering of the elements and nodes connected by an edge to the node in question, with local nodes and elements arbitrarily taken to be numbered in counter-clockwise fashion. Let the flux on the faces on the boundary be denoted by $\Phi_1^\star$ and $\Phi_{K+1}^\star$, the flux specified by the boundary Neumann condition. Recall that Φ_k^k denotes the flux through edge k as seen from cell k and Φ_k^{k-1} denotes the flux through edge k as seen from cell $k-1$. Additionally, let vectors $\boldsymbol{P} \in \mathbb{R}^K$ and $\overline{\boldsymbol{P}} \in \mathbb{R}^{K+1}$ denote

the local vector of cell-centered and face-centered pressure values respectively.

$$\Phi_k^k = -\frac{1}{2}\alpha_k \left[L_k(\bar{p}_k - p_k) - \cos\theta_k L_{k+1}(\bar{p}_{k+1} - p_k)\right] \tag{4.51}$$

$$\Phi_k^{k-1} = -\frac{1}{2}\alpha_{k-1} \left[-\cos\theta_{k-1} L_{k-1}(\bar{p}_{k-1} - p_{k-1}) + L_k(\bar{p}_k - p_{k-1})\right]. \tag{4.52}$$

The flux equations are written as before, with flux on edge $k = 1$ and edge $k = K + 1$ replaced with the flux prescribed by the Neumann boundary condition. The flux relation (4.38) and matrix equations for internal edges is unchanged.

- First edge, $k = 1$.

$$L_1\Phi_1^1 = L_1\Phi_1^\star \tag{4.53}$$

- Internal edges, $k = 2 \ldots K$

$$L_k\left(\Phi_k^k + \Phi_k^{k-1}\right) = 0 \tag{4.54}$$

- Final edge, $k = K + 1$

$$L_{K+1}\Phi_{K+1}^K = L_{K+1}\Phi_{K+1}^\star \tag{4.55}$$

To recover the equation that relates cell-centered values to face-centered values for the first edge, set $k = 1$ in (4.51) and plug the result into (4.53).

$$\alpha_1 L_1 \left(L_1\bar{p}_1 - \cos\theta_1 L_2\bar{p}_2\right) = \alpha_1 L_1(L_1 - L_2\cos\theta_1)p_1 - 2L_1\Phi_1^\star \tag{4.56}$$

The equation for internal edges is identical to the flux relation derived previously for internal edges $k = 2 \ldots K$.

$$\begin{aligned} -\alpha_{k-1}\cos\theta_{k-1}L_{k-1}L_k\bar{p}_{k-1} + (\alpha_{k-1} + \alpha_k)L_k^2\bar{p}_k - \alpha_k\cos\theta_k L_k L_{k+1}\bar{p}_{k+1} \\ = \alpha_{k-1}L_k(L_k - L_{k-1}\cos\theta_{k-1})p_{k-1} + \alpha_k L_k(L_k - L_{k+1}\cos\theta_{k+1})p_k \end{aligned} \tag{4.57}$$

Similarly, to recover the equation for the final edge, set $k = K + 1$ in (4.52) and plug the result into (4.55).

$$\alpha L_{K+1}\left(L_{K+1}\bar{p}_{K+1} - \cos\theta_K L_K\bar{p}_K\right) = \alpha_K L_{K+1}\left(L_{K+1} - L_K\cos\theta_K\right)\bar{p} - 2L_{K+1}\Phi_{K+1}^\star \tag{4.58}$$

Note that the application of Neumann boundary conditions does not require the definition of "ghost cells" or edges outside the mesh. Boundary conditions are applied as a forcing term to the matrix equation relating flux in terms of cell-centered and face-centered values.

The set of flux relations (4.56 - 4.58) for edge nodes can be written in matrix form, similar to the form for internal nodes. Let $\widetilde{\cdot}$ denote an operator for a boundary node.

$$\widetilde{M}\overline{\boldsymbol{P}} = \widetilde{S}\boldsymbol{P} - \boldsymbol{B} \tag{4.59}$$

Note that linear functions $\widetilde{M}$ and $\widetilde{S}$ compute edge fluxes, operating on face-centered and cell-centered values respectively. As such, $\widetilde{M} \in \mathbb{R}^{K+1\times K+1}$ and $\widetilde{S} \in \mathbb{R}^{K+1\times K}$. Matrix $\widetilde{M}$ has tridiagonal non-zero structure. Matrix $\widetilde{S}$ has non-zero entries on the diagonal and sub-diagonal. The vector $\boldsymbol{B}$ defines the contribution of the boundary flux. The values of $\boldsymbol{B}$ are defined as

$$B_i = \begin{cases} 2L_1\Phi_i^\star, & i = 1 \\ 0, & i = 2\dots K \\ 2L_{K+1}\Phi_{K+1}^\star, & i = K+1. \end{cases} \tag{4.60}$$

The derivation of the diffusion operator on boundary nodes then proceeds as in Section 4.3.1. The contribution of boundary node P_q to the flux in local cell "k" is defined as

$$\widetilde{C}_k = \frac{1}{2}(L_k\Phi_k^k + L_{k+1}\Phi_{k+1}^k). \tag{4.61}$$

Substituting the edge flux terms into (4.61) and noting that the entries of $\widetilde{S}$ appear, the nodal contribution can be written in a similar fashion as before,

$$\widetilde{C}_k = -\frac{1}{4}\left[\sum_{j=1}^{K+1} \widetilde{S}_{i,j}^T(\overline{p}_j - p_k)\right]. \tag{4.62}$$

Rewrite the face-centered values in terms of the cell-centered values and, this time, the boundary conditions, using (4.59). Define the matrix Λ as before to recover the vector $\widetilde{\boldsymbol{C}}$ of local flux contributions,

$$\widetilde{\boldsymbol{C}} = -\frac{1}{4}\left(\widetilde{S}^T\widetilde{M}^{-1}\widetilde{S} - \Lambda\right)\boldsymbol{P} + \frac{1}{4}\widetilde{S}^T\widetilde{M}^{-1}\boldsymbol{B}. \tag{4.63}$$

Then, the Laplacian operator for node $\boldsymbol{P}_q$ can be written as a matrix $\widetilde{\boldsymbol{A}}_q$, with a forcing term G_q accounting for the flux prescribed by the boundary condition,

$$\widetilde{\boldsymbol{A}}_q = -\frac{1}{4}\left(\widetilde{S}^T\widetilde{M}^{-1}\widetilde{S} - \Lambda\right) \tag{4.64}$$

$$G_q = -\frac{1}{4}\widetilde{S}^T\widetilde{M}^{-1}\boldsymbol{B}. \tag{4.65}$$

4.3.2.2 Periodic Boundary Conditions. Consider the case of four-way periodic boundary conditions on a rectangular domain $\Omega = [x_{min}, x_{max}] \times [y_{min}, y_{max}]$, triangulated with a set of Q nodes $\{P_q\}_{q=1}^{Q}$. It is assumed that for every node on the East wall of the domain has a corresponding node on the West wall, i.e. if $(x_{max}, y^\star) \in \partial\Omega_{East}$ then $(x_{min}, y^\star) \in \partial\Omega_{West}$. Similarly, every node on the South wall of the domain has a corresponding node on the North wall of the domain. Additionally, it may be said that the nodes on the North East, South East, and South West corners of the mesh correspond to the node in the North West corner of the mesh. Finally, note that the solution on a boundary node and its periodic equivalent will be equal.

We can then define a set of "reduced nodes" or "unique nodes" $\left\{\widehat{P}_q\right\}_{q=1}^{\widehat{Q}}$ in the domain that completely determines the solution on the set of all nodes. Here, this is taken to be the set of nodes in the interior of the domain ($\widehat{P}_q \notin \partial\Omega$) and the set of nodes that lies of the North and West edges of the domain, removing the nodes in the North East and South West corners, as seen in Figure 4.5. The choice of a North/West preference is arbitrary, but it was what was implemented in this work, so further discussion will assume the same directional preference.

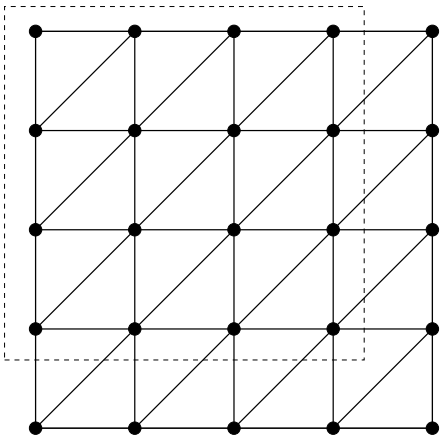

Figure 4.5: A triangulated mesh, with the set of "unique nodes" surrounded by a dotted line box.

It is useful to define a "ghost grid," a set of grid elements and associated nodes that are a periodic translation of boundary elements to outside of the true computational domain (see Fig.

4.6). The solution in these "ghost" elements will be equal to the solution of the associated true element in the domain. Thus, the matrix can be formed by computing the local diffusion matrix (4.50) for all unique nodes $\widehat{P}_q$ as if the node were an internal node. If a local node in the diffusion stencil corresponds to a "ghost element," then the local flux contribution associated with the ghost element is attributed to the corresponding true mesh element, i.e. the periodic translation of the ghost element. Note that no calculation is necessary for the nodes not in the "unique node" subset, as these nodes correspond to the periodic extension of some unique node, and as such the contribution of that node to the local flux is computed by the unique node.

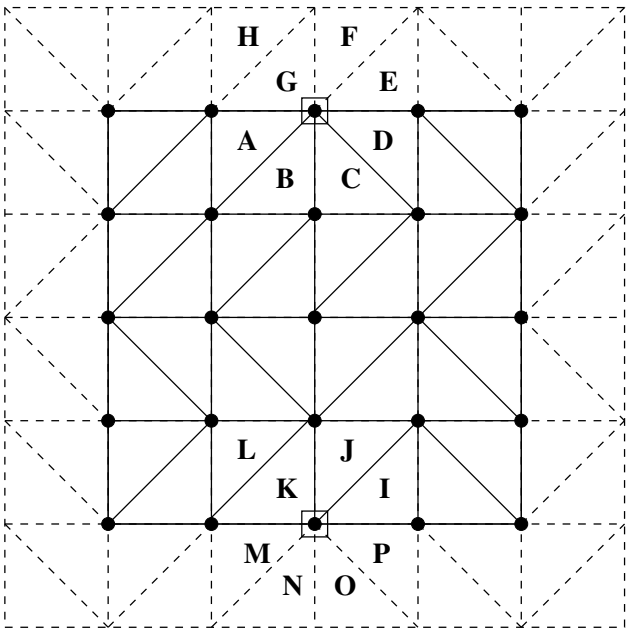

Figure 4.6: A triangulated mesh with periodically translated "ghost elements" is shown, in dotted lines. Nodes are shown as filled circles. The computational node and its periodic translation are shown with a small square around the node. The stencil for the node in question is show. Elements A, B, C, D correspond to ghost elements M, N, O, P. Similarly, elements I, J, K, L correspond to ghost elements E, F, G, H.

4.3.3 Global All-Speed SOM-FV Operator Formulation

Consider a computational domain Ω with E elements and Q nodes. Let $\boldsymbol{A}_q$ represent the matrix for the Laplacian operator at some vertex $\boldsymbol{P}_q \in \Omega \cup \partial\Omega$. Let $full(\cdot)$ be the operator that maps entries of a dense local matrix to entries of a sparse global matrix. Let $\mathbb{A} \in \mathbb{R}^{E\times E}$ be the sparse

global matrix for the Laplacian operator on Ω. Then, the global Laplacian operator can be written in terms of the local Laplacian operators $\boldsymbol{A}_q$,

$$\mathbb{A} = \sum_{q=1}^{Q} full(\boldsymbol{A}_q). \tag{4.66}$$

Additionally, let $full_v(\cdot)$ be the operator that maps the local vector forcing vector G_q to the global forcing vector $\mathcal{G} \in \mathbb{R}^E$,

$$\mathcal{G} = \sum_{\boldsymbol{P}_q \in \partial\Omega} full_v(G_q)$$

For the all-speed pressure update equation,

$$\frac{\partial p}{\partial t} + \boldsymbol{u} \cdot \nabla p = -\rho c^2 \nabla \cdot \boldsymbol{u} \tag{4.67}$$

discretized as

$$\frac{1}{(\rho c^2)^{n+1}} p^{n+1} - \Delta t^2 \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{1}{(\rho c^2)^{n+1}} p^a(\rho^{n+1}, e^a_{int}) - \Delta t \nabla \cdot \boldsymbol{u}^a, \tag{4.68}$$

we have that the diffusion coefficient is $D = \frac{1}{\rho^{n+1}}$, and the source term is $S = \frac{1}{(\rho c^2)^{n+1}} p^a(\rho^{n+1}, e^a_{int}) - \Delta t \nabla \cdot \boldsymbol{u}^a$. Define the entries of the forcing vector $\mathcal{S} \in \mathbb{R}^E$

$$\mathcal{S}_i = \int_{\Omega_i} \left(\frac{1}{(\rho c^2)^{n+1}_i} p^a(\rho^{n+1}_i, e^a_{i,int}) - \Delta t \nabla \cdot \boldsymbol{u}^a_i \right) d\Omega \tag{4.69}$$

Define the diagonal matrix $\Upsilon \in \mathbb{R}^{E \times E}$ as in (4.70), where $\delta_{i,j}$ is the Kroenecker delta function and V_e is the volume of cell Ω_e.

$$\Upsilon_{i,j} = \frac{1}{(\rho c^2)^{n+1}} \delta_{i,j} V_e \tag{4.70}$$

Then, by calculating matrix $\mathbb{A}$ as in (4.66) for $D = \frac{1}{\rho^{n+1}}$, the equation (4.68) for the global vector $\boldsymbol{p}^{n+1}$ of pressure values can be written as

$$(\Upsilon + \Delta t^2 \mathbb{A}) \boldsymbol{p}^{n+1} = \mathcal{S} + \mathcal{G}. \tag{4.71}$$

For the case of an incompressible material (i.e. $c^2 \to \infty$), then (4.71) simplifies to

$$\mathbb{A} \boldsymbol{p}^{n+1} = -\int_{\Omega} \nabla \cdot \boldsymbol{u}^a d\Omega. \tag{4.72}$$

The divergence term appearing on the right-hand side of (4.69) and (4.72) is evaluated using Green's theorem, with velocity on the cell face defined as the average of the nodal velocities that make up the endpoints of the line segment defining the cell face.

4.3.4 Consistent Projection-Based Gradient Operator

A new method for calculating the cell-centered gradient, in a manner consistent with the Support Operator Method discretization of the diffusion operator, is presented. The original application of the method of Breil and Maire [12] did not require the computation of cell-centered gradients. However, a pressure gradient operator is necessary for the update of state variables in a fluid algorithm. A first-order accurate, cell-centered gradient operator on unstructured triangular meshes can be derived using the building blocks of the SOM. First, note that the matrix F (4.42) is the projection operator that maps local cell-centered values P to local half-face centered values $\overline{P}$. Given local cell-centered values $P \in \mathbb{R}^K$, individual face values $\bar{p}_k$ are computed from (4.42) as

$$\bar{p}_k = \sum_{l=1}^{K} F_{k,l} p_l. \tag{4.73}$$

A cell-centered pressure gradient can be defined in standard finite volume form (4.74) using face-centered pressure values, p_e. Values L_e and $\hat{n}_e$ denote the length and outward unit normal for face e, respectively.

$$\frac{1}{|\Omega_i|} \int_{\Omega_i} \nabla p \cdot dV = \frac{1}{|\Omega_i|} \int_{\partial\Omega_i} p \ \hat{n} \cdot dS \approx \frac{1}{|\Omega_i|} \sum_{e} p_e L_e \hat{n}_e \tag{4.74}$$

Denote the global projection operator matrix $\mathbb{F}$ as the sum of contributions from the local projection operators over all nodes. Note that for a given node P_q with "k" local element neighbors, local projection operator F^n is a dense matrix operating on the vector of local cell centered values $P \in \mathbb{R}^k$. Let $full(\cdot)$ denote the operator that maps entries in the dense local projection matrix to entries in a sparse global projection matrix. Then, the global operator which maps cell-centered values to face-centered values can be written as (4.75). For a mesh with N nodes, E elements, and D edges, the global projection operator $\mathbb{F} \in \mathbb{R}^{D \times E}$ maps the vector of global cell-centered values $\mathbb{P} \in \mathbb{R}^E$ to the vector of global face-centered values $\bar{\mathbb{P}} \in \mathbb{R}^D$.

$$\mathbb{F} = \sum_{n=1}^{N} full(F^n) \tag{4.75}$$

$$\bar{\mathbb{P}} = \mathbb{F}\mathbb{P} \tag{4.76}$$

Thus, the set of global face-values $\bar{\mathbb{P}}$ can be used, as in (4.74), to compute a cell-centered gradient operator from cell-centered data. Let $\widetilde{\mathbb{G}} \in \mathbb{R}^{E\times D}$ be the operator (4.74) that computes a cell-centered gradient from cell face values. Then, the gradient operator that computes cell-centered gradients given cell-centered values can be written as

$$\nabla p \approx \widetilde{\mathbb{G}}\mathbb{F}\mathbb{P} = \mathbb{G}\mathbb{P}. \tag{4.77}$$

Validation of this operator can be found in Section 5.3.

4.3.4.1 Projection Operator for Periodic Domains. Special consideration is again given to periodic domains. As in Section 4.3.2.2, a set of "ghost" elements is defined as the periodic translation about the domain of any element such that any vertex of the element lies on $\partial\Omega$. The local projection matrix F_q is calculated for all "unique" elements $\{P_q\}$ as in 4.42. To ensure that the projection is properly computed for all edges, care must be taken with respect to contributions to the projection operator from ghost elements. If an edge lies outside the mesh, the projection weights calculated from F_q must be distributed to *all* periodic translations of that edge which lie in $\Omega \cup \partial\Omega$. Figure 4.7 demonstrates that a ghost edge outside of the domain may have more than one edge in the domain that corresponds to a periodic translation. The edges outside the domain A (red), B (blue), C (green), and D (yellow) are marked with colored squares. Each of the periodic translations of these edges is marked with a circle of the corresponding color.

4.3.5 State Variable Update

State variables for the 2D Unstructured Support Operator Finite Volume method are updated using the projection operator defined in Section 4.3.4. Let pressure $\boldsymbol{p}^{n+1}$ be the solution to (4.71) and $\mathbb{F} \in \mathbb{R}^{D\times E}$ be the global projection operator that maps cell-centered values to face centered values, on a mesh with E elements and D edges/faces. Then the pressure p_f^{n+1} on face f is

$$p_f^{n+1} = \sum_{j=1}^{E} \mathbb{F}_{f,j} p_j^{n+1}. \tag{4.78}$$

Nodal velocities are defined using the inverse distance-weighted least squares procedure over all elements that share a node, as in the 2D unstructured advection algorithm. Assuming that face f is defined as the line segment with endpoints at nodes n_1 and n_2, then the face velocity is defined

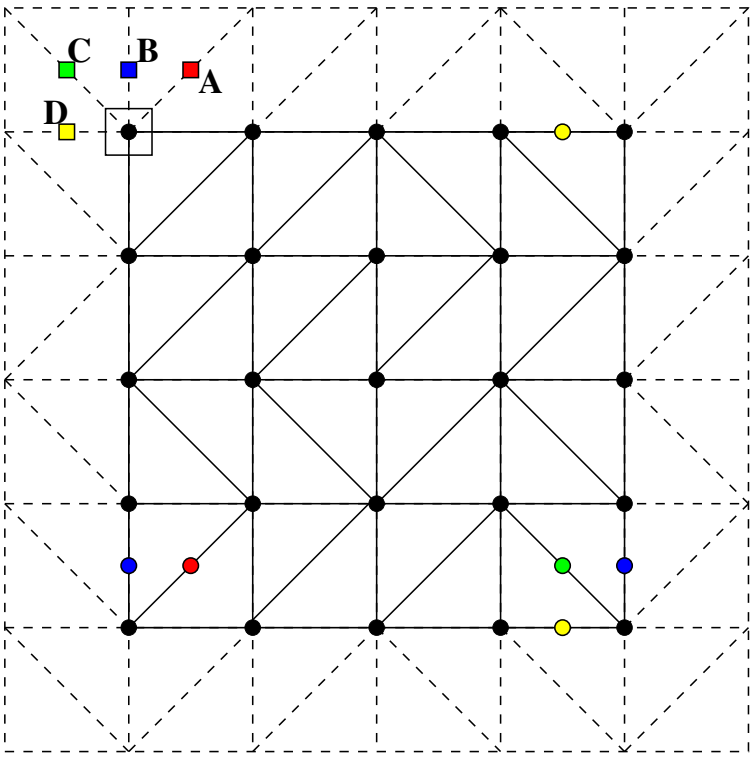

Figure 4.7: Ghost element face A (red), B (blue), C (green), and D (yellow) computed for the node in the North West corner (marked with a square) are marked with colored squares. The projection weights computed for the ghost faces must be applied to the faces marked with circles of the corresponding color.

as

$$u_f^{n+1} = \frac{u_{n_1}^{n+1} + u_{n_2}^{n+1}}{2}.$$

Velocity and energy are advanced in time using a Green's theorem discretization. For a cell Ω_i with a function $i(f)$ mapping local indices f to a global face index, edge outward unit normal vectors $\widehat{n}_{i(f)}$, and edge lengths $L_{i(f)}$, then we have

$$\boldsymbol{u}_i^{n+1} = \boldsymbol{u}_i^a - \Delta t \frac{\sum_{f=1}^{3} p_{i(f)}^{n+1} L_{i(f)} \widehat{n}_{i(f)}}{\rho_{i,*}^{n+1} |\Omega_i|} \tag{4.79}$$

$$E_i^{n+1,m} = \begin{cases} E_i^{a,m} - \Delta t \dfrac{\sum_{f=1}^{3} p_{i(f)}^{n+1} \boldsymbol{u}_{i(f)}^{n+1} \cdot \widehat{n}_{i(f)} L_{i(f)}}{\rho_i^{n+1,m} |\Omega_i|}, & |\Omega_i^m| > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.80}$$

4.3.6 2D Nodal Velocity Computation

Nodal velocities, used for defining characteristics for CISL advection, evolving energy forward in time, evaluating the right-hand side of the pressure equation, and vorticity calculation are computed using an inverse-distance weighted least squares procedure [30, 64]. This technique is second-order accurate for smooth velocity profiles. While it produces diffusion at shock fronts, it is seen to perform robustly on a number of shock capturing tests for the Finite Element Method.

The linear least squares method seeks the linear interpolant $\phi(x, y)$ that minimizes error for an overdetermined matrix-vector system.

$$\phi(x, y) = ax + by + c$$

Consider, a mesh node $\boldsymbol{v}_n$ with $K \geq 3$ element neighbors with cell-centered velocity values that are used to define the least squares problems for the $x-$ and $y-$velocities. The weighted least squares applies a metric defining the relative importance of minimizing error for a given point, where taking a weight to be zero effectively removes the point from the minimization problem. Let $\mathcal{X} \in \mathbb{R}^{K\times 3}$ be the matrix of coordinate locations of neighbor element centroids for a node $\boldsymbol{v}_n$, and let $\boldsymbol{x}_i$, with $i = 1, \ldots, K$ be the centroid of the i^{th} element neighbor of $\boldsymbol{v}_n$ under a local ordering.

$$\mathcal{X} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_K & y_K & 1 \end{pmatrix} \tag{4.81}$$

Let $\mathcal{W} \in \mathbb{R}^{K\times K}$ be the matrix of weights. The entries of $\mathcal{W}$ are defined as

$$\mathcal{W}_{i,j} = \begin{cases} \frac{1}{1+||\boldsymbol{v}_n - \boldsymbol{x}_i||_2}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}.$$

The inverse-distance weighting was found in a number of tests to produce the most accurate results for smooth problems and a number of shock capturing problems for the All-Speed FEM method.

The weighted least squares solution is defined as the vector $\boldsymbol{p} \in \mathbb{R}^3$ that exactly solves

$$\mathcal{X}^T\mathcal{W}\mathcal{X}\boldsymbol{p} = \mathcal{X}^T\mathcal{W}\boldsymbol{z}. \tag{4.82}$$

While the matrices $\mathcal{X}$,$\mathcal{W}$ are the same for constructing the interpolant for velocity in each of the two cardinal directions, the right-hand side is different. Here, z_i is the velocity component of the

cell centered velocity located at centroid $\boldsymbol{x}_i$. Letting $\phi_u(\boldsymbol{x})$ and $\phi_v(\boldsymbol{x})$ be the weighted least squares interpolant for velocity in the $x-$ and $y-$directions respectively, then the velocity at node $\boldsymbol{v}_n$ is equal to

$$\boldsymbol{u}(\boldsymbol{v}_n) = (\phi_u(\boldsymbol{v}_n), \phi_v(\boldsymbol{v}_n))^T.$$

Letting (a_u, b_u, c_u) and (a_v, b_v, c_v) be the coefficients of ϕ_u and ϕ_v respectively, then vorticity at node $\boldsymbol{v}_n$ can be calculated as

$$\nabla \times \boldsymbol{u}(\boldsymbol{v}_n) = b_u - a_v. \tag{4.83}$$

The weighted least squares procedure is used to define nodal velocity rather than a momentum conservative interpolation due to tests that demonstrated severe irregularities for compressible, incompressible, and multiphase test problems when momentum conservative interpolation is used. It should be noted that for structured, uniform grids in 1D, 2D, and 3D Cartesian coordinates, a face-centered control volume $\Omega_{i+1/2}$ has the same volume as a given computational cell, and the face centroid $\boldsymbol{x}_{i+1/2}$, where advective velocity lives, lies at the centroid of the face-centered control volume. For a general unstructured triangular grid, neither of these statements is true. Defining a nodal control region in a cell Ω_r^i for cell i and local node r as in Figure 4.2, it is clear that size and centroid of a nodal control volume over all K cells that share a given node can differ depending on mesh quality. While it may be possible to improve results by computing nodal velocity using a multiphase Riemann solver as in [22], this work was developed to extend the methods of [38] to 2D unstructured grids without the use of a Riemann solver.

4.3.7 Verification Problems

Verification problems were tested to demonstrate functionality of the pressure solver. I have verified exactness for both the constant and linear Helmholtz problems with Neumann boundary conditions.

$$p - \nabla \cdot (D(x, y)\nabla p) = S \tag{4.84}$$

Second order convergence was verified for polynomial and trigonometric functions, using constant and smoothly varying coefficients. Test functions were taken to vary in the x- and y-directions. First order convergence of the derivative was verified for each of these tests as well. All tests were

performed on both pseudo-structured and truly unstructured grids. Verification results are presented in Section 5.3.

4.4 2D All-Speed Finite Element Method

Again, we are discretizing the following equation.

$$\frac{p^{n+1}}{(\rho c^2)^{n+1}} - \Delta t^2 \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{p^a(\rho^{n+1}, e^a_{int})}{(\rho c^2)^{n+1}} - \Delta t \nabla \cdot \boldsymbol{u}^a \tag{4.85}$$

Assume that the solution p^{n+1} is a linear combination of N piecewise-linear basis functions φ_j that are 1 at node j and 0 at every other node, with weights α_j.

$$p^{n+1} = \sum_{j=0}^{N} \alpha_j \varphi_j(x, y) \tag{4.86}$$

The equation (4.86) is substituted into (4.85). The result is multiplied by test basis function φ_i and integrated over the domain to recover the formulation of the FEM problem (4.87).

$$\sum_{j=0}^{N} \alpha_j \int_{\Omega} \frac{\varphi_j \varphi_i}{(\rho c^2)^{n+1}} d\Omega + \Delta t^2 \sum_{j=0}^{N} \alpha_j \int_{\Omega} \frac{\nabla \varphi_j \cdot \nabla \varphi_i}{\rho^{n+1}} d\Omega = \int_{\Omega} \left(\frac{p^a}{(\rho c^2)^{n+1}} - \Delta t \nabla \cdot \boldsymbol{u} \right) \varphi_i d\Omega \tag{4.87}$$

The left-hand side is straight-forward, but the right-hand side can be viewed in two ways.

1. Define a forcing function $f(x, y) = \frac{p^a}{(\rho c^2)^{n+1}} - \Delta t \nabla \cdot \boldsymbol{u}^a$. The right-hand side can be written as $\int_{\Omega_i} f(\boldsymbol{x}) \varphi d\Omega$, with the term $\nabla \cdot \boldsymbol{u}^a$ discretized at cell centers in a finite volume fashion. This is then discretized as is standard for the second-order approximation, evaluating the right-hand side at $\vec{c}$, the element centroid.

$$\int_{\Omega_i} \left(\frac{p^a}{(\rho c^2)^{n+1}} - \Delta t \nabla \cdot \boldsymbol{u} \right) \varphi_i d\Omega \approx f(\vec{c}_i) \varphi_i(\vec{c}_i) |\Omega_i|. \tag{4.88}$$

2. Evaluate the pressure term in sharp interface fashion as a function of $p^a = p(\rho_i^{n+1,max}, e_i^{a,max})$, the density and specific internal energy of the material that occupies the most volume in cell Ω_i. Evaluate the divergence using Green's formula.

$$\frac{p^a(\rho_i^{n+1,max}, e_i^{max}) \varphi_i(\vec{c})}{(\rho_i^{n+1} c^2(\rho_i^{n+1max}, e_i^{a,max}))} |\Omega_i| - \Delta t \int_{\partial\Omega} \varphi_i \boldsymbol{u} \cdot \hat{n} dl \tag{4.89}$$

The second term is then discretized as

$$\int_{\partial\Omega} \varphi_i \boldsymbol{u} \cdot \hat{n} dl \approx \sum_{e=1}^{3} \frac{1}{2} \frac{\boldsymbol{u}_{n(e)} + \boldsymbol{u}_{n(e)+1}}{2} \cdot \hat{n}_e L_e, \tag{4.90}$$

where $n(e)$ and $n(e)+1$ are the global indices of the nodes at the start and end of local edge e. Implementation of the right-hand side as in Case (1) leads to oscillation at at contact discontinuities and rapid decay in wall pressure for the Oscillating Water Column test. This behavior is non-physical and therefore undesirable. Implementation of the right-hand side as in Case (2) is found to produce less oscillatory behavior at a contact discontinuity, even in the presence of large density ratios. All tests using the all-speed Finite Element Method are presented using the implementation of Case (2).

4.4.1 State Variable Updates

The pressure gradient is discretized in a manner consistent with the Finite Element discretization, using nodal values from the FEM solver. Let $\varphi_{n(l)}$ denote the basis function with local index l for a given element. For a cell Ω_i,

$$\nabla p \approx \sum_{l=1}^{3} \nabla \varphi_{n(l)}(\boldsymbol{x}) \tag{4.91}$$

Velocity is updated as

$$\boldsymbol{u}_i^{n+1} = \boldsymbol{u}_i^a - \frac{\Delta t}{\rho_{i,*}} \sum_{l=1}^{3} \nabla \varphi_{n(l)}(\boldsymbol{x}) \tag{4.92}$$

Here, $\rho_{i,*}$ is the volume weighted density of material in cell i. Note that while this uses volume information about materials occupying a given cell, it does not take into account interfacial orientation information. So, given a volume fraction partitioning $\{V_m\}_{m=1}^{M}$, then every interface reconstruction $\Gamma(\widehat{n}, b)$ will yield the same density value $\rho_{i,*}$, regardless of orientation $\widehat{n}$. This is in contrast to the 1D Finite Volume method and its extension to higher dimensions, which calculate "face density" in a reduced, face-centered control volume. Additionally, as the unknowns for pressure are located at nodes for the Finite Element method, rather than cell centers as for Finite Volume methods, it is not necessary to solve any additional equations for pressure on cell faces. Energy is updated using Green's Theorem, integrating over the cell face.

$$E_i^{n+1,m} = \begin{cases} E_i^{a,m} - \Delta t \dfrac{\sum_{e=1}^{3} \left(\dfrac{u_{n(e)} + u_{n(e)+1}}{2} \right) \left(\dfrac{p_{n(e)} + p_{n(e)+1}}{2} \right) p_e L_e \hat{n}_e}{|\Omega_i| \rho_i^{n+1,m}}, & |\Omega_i^m| > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.93}$$

CHAPTER 5

NUMERICAL RESULTS

The acoustic time step Δt_{u+c} is chosen according to the following scheme.

$$\Delta t_{u+c}^{n+1} = \begin{cases} \Delta t^* = CFL \frac{\Delta x}{\max_i(|u_i|+c_i)}, & \Delta t^* \leq cmax \cdot \Delta t^n \\ cmax \cdot \Delta t^n, & \text{otherwise} \end{cases} \tag{5.1}$$

The advective time step Δt_u does not consider the sound speed:

$$\Delta t_{u}^{n+1} = \begin{cases} \Delta t^* = CFL \frac{\Delta x}{\max_i(|u_i|)}, & \Delta t^* \leq cmax \cdot \Delta t^n \\ cmax \cdot \Delta t^n, & \text{otherwise} \end{cases} \tag{5.2}$$

The factor *cmax* denotes the maximum allowed increase in the time step from one step to the next. For all tests presented, *cmax* is set to be 1.1.

5.1 One Dimensional, Single Phase, Shock Tube and Shock-Turbulence Interaction Test Problems

The new algorithm is tested on the Sod, Lax, and Strong Shock Tube test problems. Also the new algorithm is tested on a shock/turbulence interaction test problem (see section 4.6 of [48]). In each case, the gas is governed by the ideal gas equation (5.3) with $\gamma = 1.4$ and we prescribe zero viscosity.

$$p = (\gamma - 1)\rho e_{int} \tag{5.3}$$

Also, for each case, the time step was derived from the acoustic time step (5.1) with CFL= 1/2. In other words the time step

$$\Delta t_{u+c} = \frac{1}{2} \frac{\Delta x}{\max_i(|u_i| + c_i)}$$

was used. For the Sod shock tube problem (5.4), we also report results when the acoustic time step with CFL= 1.5 is used.

The initial conditions for these three shock wave test problems are:

Sod Shock tube

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (1,0,1), & x \leq 0.5 \\ (0.125,0,0.1), & x > 0.5 \end{cases} \tag{5.4}$$

Lax Shock tube

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (0.445, 0.698, 3.528), & x \leq 0.5 \\ (0.5,0,0.571), & x > 0.5 \end{cases} \tag{5.5}$$

Strong Shock tube

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (1,0,10^{10}), & x \leq 0.5 \\ (0.125,0,0.1), & x > 0.5 \end{cases} \tag{5.6}$$

Shock-Turbulence interaction

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (3.857148, 2.629369, 10.333333), & 0 \leq x \leq 1 \\ (1 + 0.2\sin(5x-5), 0, 1), & 1 < x < 10 \end{cases} \tag{5.7}$$

For the shock tube test cases, the error E_N and rate of convergence p_N are reported for density in Table 5.1, using an exact Riemann-solver as the reference solution. The error E_N is the L^1 error and is defined as,

$$E_N = \sum_{i=0}^{N-1} |\rho_i - \rho^{\text{exact}}(x_i)| \Delta x. \tag{5.8}$$

$$x_i = (i + 1/2)\Delta x \tag{5.9}$$

The rate of convergence is defined as

$$\tau_N = \log_2 |\frac{E_{N/2}}{E_N}| \tag{5.10}$$

In Figures 5.1 through 5.7, plots for the shock tube problems (Sod $t = 0.15$, Lax $t = 0.12$, Strong $t = 2.5E-6$) and for the shock-turbulence interaction problem ($t = 1.8$) are shown. The correct shock speed and shock strength are captured under grid refinement for these test problems. Errors reported in Table 5.1 show convergence is slightly less than first order, which is expected due to use of a shock-capturing method and due to the fact that the asymptotically-preserving pressure equation makes an $\mathcal{O}(\Delta t)$ error by fixing velocity on the right-hand side at time t^{n+1}.

Remarks:

Table 5.1: L^1 error E_N and convergence rate τ_N for density for the Sod Shock Tube problem at $t = 0.15$ (5.4) and the Strong Shock Tube problem at $t = 2.5e\text{-}6$ (5.6). A linear MINMOD reconstruction in all conserved quantities is used.

N	Sod E_N CFL= 0.5	Sod τ_N CFL= 0.5	Sod E_N CFL= 1.5	Sod τ_N CFL= 1.5	Strong E_N CFL= 0.5	Strong τ_N CFL= 0.5
50	1.875e-02	N/A	2.126e-02	N/A	—	—
100	1.174e-02	0.676	1.417e-02	0.586	2.308e-02	N/A
200	7.291e-03	0.687	1.031e-02	0.458	1.326e-02	0.800
400	4.406e-03	0.726	6.938e-03	0.572	7.889e-03	0.749
800	2.634e-03	0.743	4.415e-03	0.652	4.607e-03	0.776
1600	1.592e-03	0.726	2.785e-03	0.665	2.761e-03	0.739

Table 5.2: L^1 error E_N for the Sod, Lax, and Strong Shock Tube problem at respective final times at a variety of CFL numbers for a fixed resolution, $N = 400$, with 2 velocity sweeps.

CFL	Sod E_N	Lax E_N	Strong E_N
0.5	4.406e-03	9.187e-03	7.889e-03
0.75	5.148e-03	9.450e-03	8.505e-03
1.00	5.817e-03	9.756e-03	9.035e-03
1.25	6.413e-03	1.002e-02	9.409e-03
1.50	6.938e-03	1.027e-02	9.751e-03

- Non-conservative schemes fail for the Strong Shock Tube test, due to their inability to correctly compute shock speeds [34, 85].
- While the density jump for the Strong Shock tube problem at the material interface (5.4) is not captured as sharply as in [31], less oscillation is observed at the head of the rarefaction wave.
- For the Sod shock tube test, the sensitivity of results due to different values for CFL is illustrated; see Figure 5.2. Discontinuities are smeared as Δt is increased beyond the acoustic time step constraint.
- A study of the effect of varying the CFL number from 0.5 to 1.5 at a fixed mesh resolution, $N = 400$, is presented in Table 5.2 for the Sod, Lax, and Strong shock tube tests. For each problem, although error increases with the CFL number, the numerical method does not become unstable, even as the time step increases past the standard acoustic time step restriction. This indicates the benefits of the CISL advection method's ability to take large time steps in a stable fashion, while also computing the time evolution of pressure in a stable fashion.

- The results for the Lax shock tube test (Fig. 5.3) show overshoot at the shock front. The CISL algorithm will calculate the initial face velocity at the contact velocity as the mass-weighted average of velocity in the cells on either side, rather than via Riemann solver. This leads to a spurious overshoot that travels with the shock.

- For the shock-turbulence interaction problem, this method does not capture the shock ringing that appears upstream of the shock as well as the method developed in [48]. Although this method converges under grid refinement, the grid required to approximate the waves immediately downstream of the shock with a given accuracy is about twice as fine as in [48]. While this method performs worse in the smooth oscillatory regions, out method does capture the shock wave with a comparable number of grid points when compared to [48].

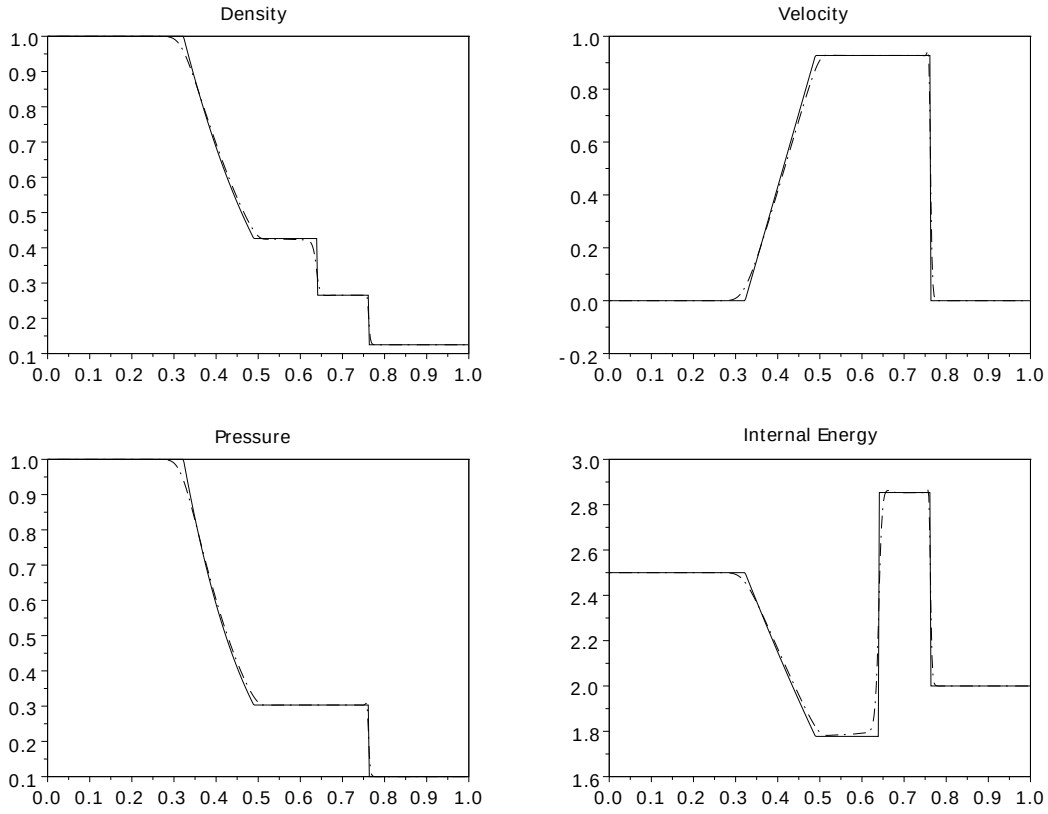

Figure 5.1: Sod Shock Tube at time t = 0.15. The solid line indicates the exact solution via Riemann solver. The dotted line is the reference solution for N=400 cells. A linear MINMOD reconstruction in all conserved quantities is used.

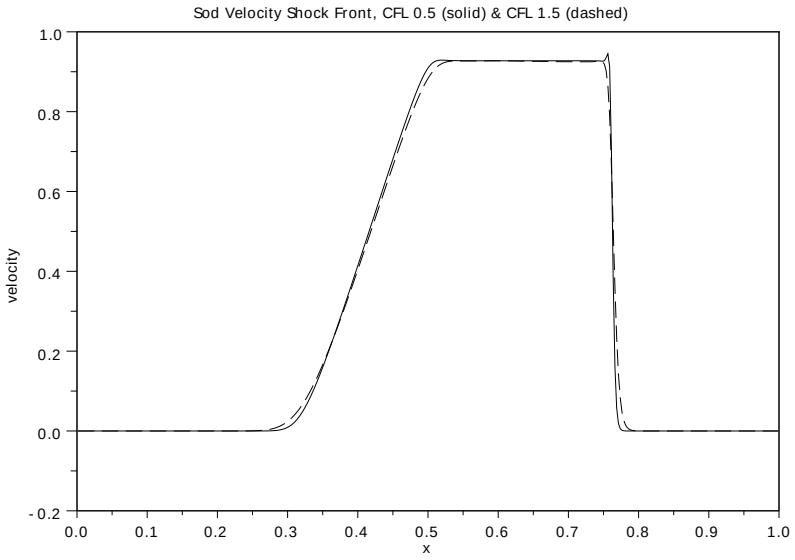

Figure 5.2: Velocity for the Sod Shock Tube at time t = 0.15. The solid line indicates the numerical results using the acoustic time step constraint with CFL= 0.5 and the dashed line represent the results using CFL= 1.5. The number computational cells is $N = 400$. A linear reconstruction for all conserved quantities with MINMOD reconstruction is used.

Table 5.3: Errors and Total Variation for Sod, Lax, and Strong Shock Tube problems, taken at final times t = 0.15, 0.12, 2.5E-6 respectively, for a given number of velocity fixed point iterations, as in Section 3.2.5.3. Increased fixed point iterations causes a slight increase in L^1 error but reduces non-physical overshoots/oscillations at the shock front.

	Sod		Lax		Strong	
Sweeps	L^1 Error	TV	L^1 Error	TV	L^1 Error	TV
1	3.874e-03	0.914	8.954e-03	2.079	6.917e-03	1.807e+00
2	4.406e-03	0.882	9.187e-03	1.935	7.889e-03	1.750e+00
3	4.406e-03	0.882	9.201e-03	1.933	7.921e-03	1.748e+00
4	4.407e-03	0.882	9.201e-03	1.933	7.924e-03	1.748e+00
5	4.407e-03	0.882	9.201e-03	1.933	7.924e-03	1.748e+00

5.2 1D Oscillating Water Column

A one-dimensional tube with reflecting boundaries at the left and right ends is considered. A column of inviscid water occupies the middle of the tube, with inviscid air on either end (Fig. 5.8), with uniform pressure across the domain. At time $t = 0$, water begins moving from left to right

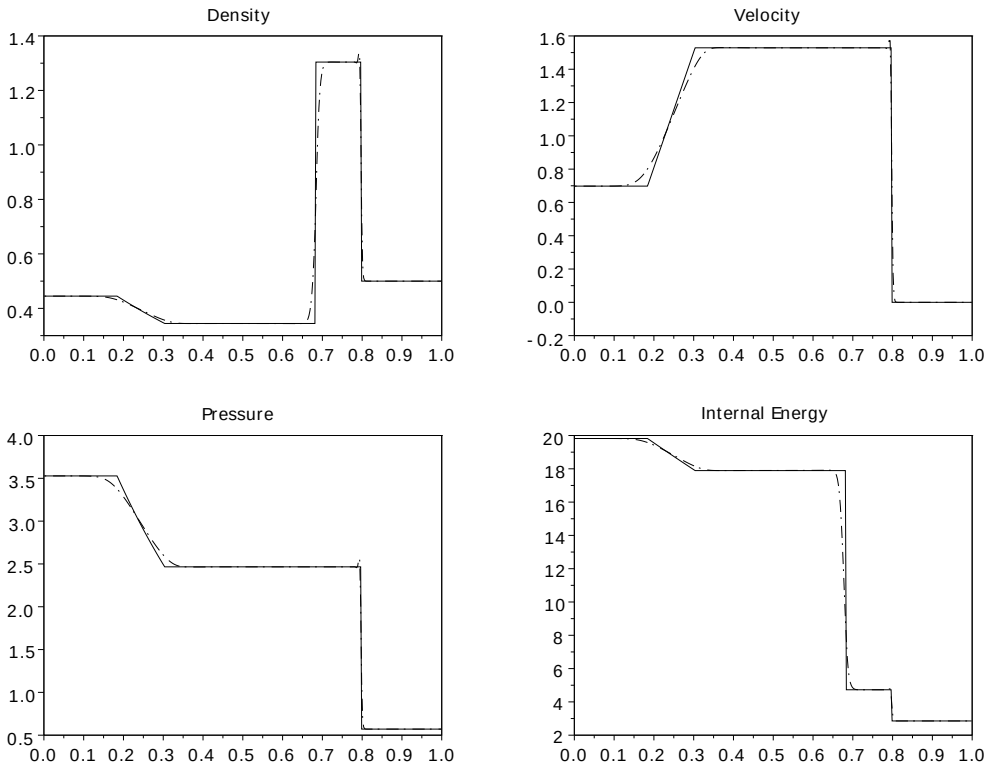

Figure 5.3: Lax Shock Tube at time t = 0.12. The solid line indicates the exact solution via Riemann solver. The dotted line is the reference solution for N=400 cells.

with velocity one, compressing the gas to the right and expanding the gas to the left. This causes a gradient in pressure across the system, which leads the water column to decelerate, stop, and then accelerate moving from right to left. The equation of state (5.11) is used to govern both materials, with the appropriate coefficients taken for air and water. A lower bound on pressure is added to account for potential cavitation in the liquid, to prevent pressure from becoming negative. If density falls below the cutoff value ρ_c, then the pressure and sound speed are computed using the cutoff density. A reflecting wall boundary condition is used at both ends of the domain.

$$\frac{p + Bp_{ref}}{(1+B)p_{ref}} = \left(\frac{\rho}{\rho_{ref}}\right)^{\gamma} \tag{5.11}$$

As in Koren, et. al. [43] and Kadioglu, et. al. [41], the coefficients are taken as $B_w = 3000$, $B_a = 0$, $\rho_{ref,w} = 1$, $\rho_{ref,a} = 0.001$, $\gamma_w = 7$, $\gamma_a = 1.4$, $p_w(t=0) = p_a(t=0) = 1$, $x_w = 0.1$. To be consistent

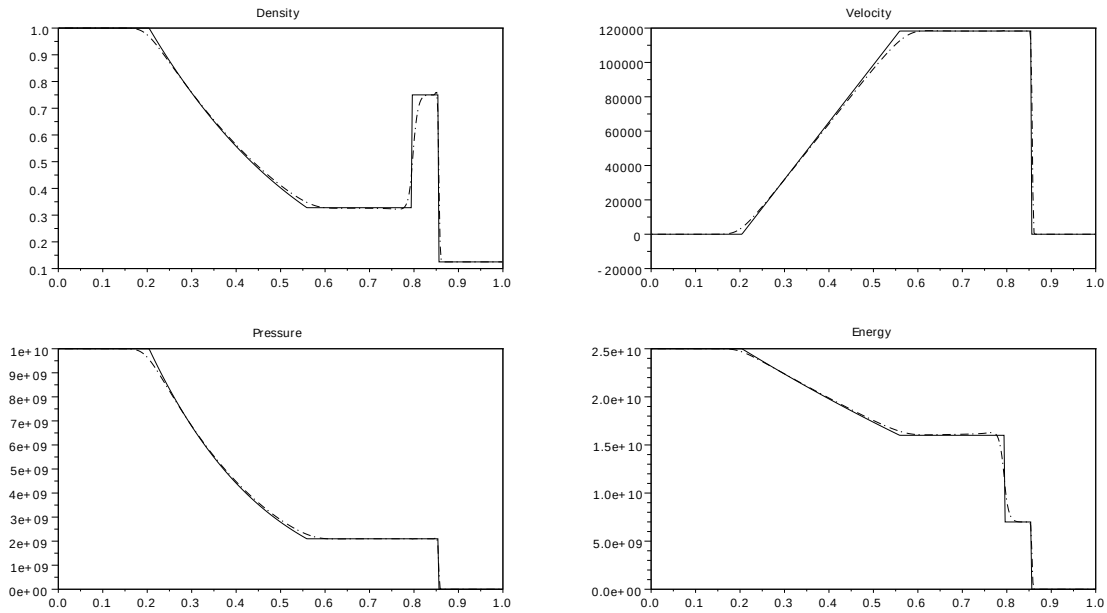

Figure 5.4: Strong Shock Tube at time t = 2.5E-6. The solid line indicates the exact solution via Riemann solver. The dotted line is the reference solution for N=400 cells. A linear MINMOD reconstruction in all conserved quantities is used.

with Kadioglu, the acoustic time step (5.1) with $CFL = 3$ is used. Notes: At the very first time step, the velocity in the gas cells directly adjacent to the interface is initialized to one as well, to obtain a velocity of +1 for interfacial cells. A non-conservative momentum update is applied at cells adjacent to the walls. The non-conservative velocity update

$$u_0^{n+1} = \frac{\rho_{0,R}|\Omega_{0,R}|u_0^a}{\rho_{0,R}|\Omega_{0,R}| + \rho_{0,L}|\Omega_{0,L}|} \tag{5.12}$$

$$u_{N-1}^{n+1} = \frac{\rho_{N-1,L}|\Omega_{N-1,L}|u_{N-1}^a}{\rho_{N-1,R}|\Omega_{N-1,R}| + \rho_{N-1,L}|\Omega_{N-1,L}|} \tag{5.13}$$

is used at the walls in cells $i = 0,\ N-1$ rather than the conservative form (4.26). Updating velocity at the wall using (4.26) can result in spurious ringing. Mass is conserved in both water and air to order $\mathcal{O}(10^{-12})$. The maximum fluctuation in the volume of the water column is 0.14%, approximately $\Delta x/50$.

Results are shown in Figure 5.9, for resolution $N = 160$. Note that the amplitude and frequency of oscillation agree with the results shown in Koren and Kadioglu. This result is novel, in that it

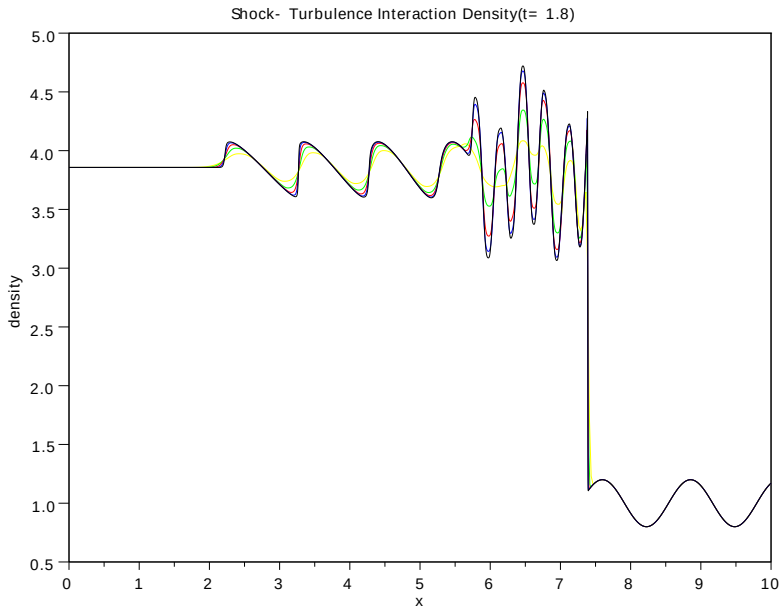

Figure 5.5: Shock-Turbulence interaction at time t = 1.8. Results for five different grid resolutions are compared: $N = 400$ (yellow), $N = 800$ (green), $N = 1600$ (red), $N = 3200$ (blue), and $N = 6400$ (black). A linear MINMOD reconstruction in all conserved quantities is used.

does not make the assumption of uniform density in the gas region, or that the gas is adiabatic. Use of CISL advection captures the incompressible dynamics of the liquid in the system, with very little overall fluctuation of liquid volume in the system. Additionally, there is no mass loss, up to machine precision.

5.2.1 1D JWL Shock Tube

A tube contains an inviscid gas governed by the Jones-Wilkins-Lee (JWL) equation of state and inviscid water, taken to be governed by the Tait equation of state, as in Wardlaw [83]. This makes the assumption that water is adiabatic. Again, a lower bound on pressure is added to account for potential cavitation in the liquid, to prevent pressure from becoming negative. If density falls below the cutoff value ρ_c, then the pressure and sound speed are computed using the cutoff density.

$$p = \begin{cases} B\left(\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right) + A, & \text{if } \rho > \rho_c \\ p_c, & \text{otherwise} \end{cases} \tag{5.14}$$

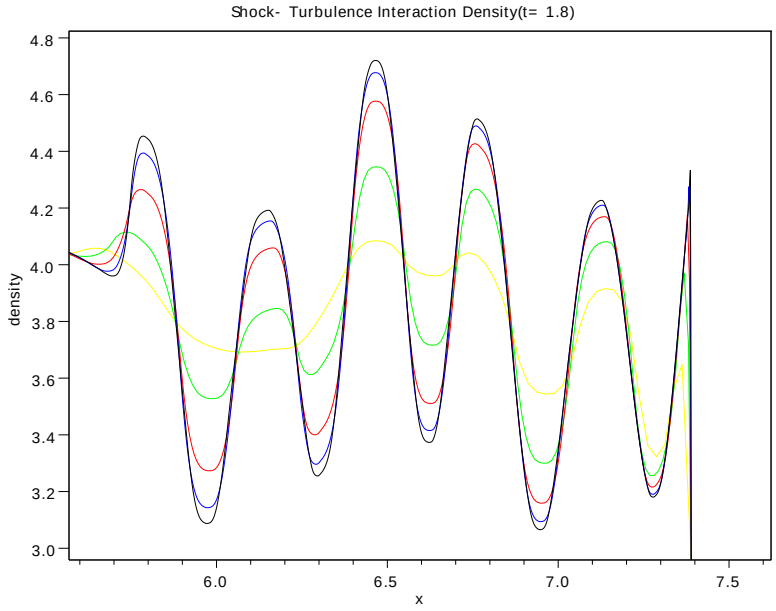

Figure 5.6: Shock-Turbulence interaction at time t = 1.8. Zoomed in view just downstream of the shock wave. Results for five different grid resolutions are compared: $N = 400$ (yellow), $N = 800$ (green), $N = 1600$ (red), $N = 3200$ (blue), and $N = 6400$ (black). A linear MINMOD reconstruction in all conserved quantities is used.

$A = 1.0\times10^6$ d/cm^2, $B = 3.31\times10^6$ d/cm^2, $\gamma = 7.15$, $\rho_0 = 1.0$g/cm^3, $\rho_c = 1.0-4.225\times10^{-5}$ g/cm^3, and $p_c = 220.2726$ d/cm^2.

The Jones-Wilkins-Lee (JWL) equation of state is a model of explosive gas, relating pressure to density and energy.

$$p = A\left(1 - \frac{\omega\rho}{R_1\rho_0}\right)\exp\left(-R_1\frac{\rho_0}{\rho}\right) + B\left(1 - \frac{\omega\rho}{R_2\rho_0}\right)\exp\left(-R_2\frac{\rho_0}{\rho}\right) + \omega\rho e_{int} \tag{5.15}$$

$A = 5.484 \times 10^{12}$ d/cm^2, $B = 0.09375 \times 10^{12}$ d/cm^2, $R_1 = 4.94$, $R_2 = 1.21$, $\omega = 0.28$, and $\rho_0 = 1.63$.

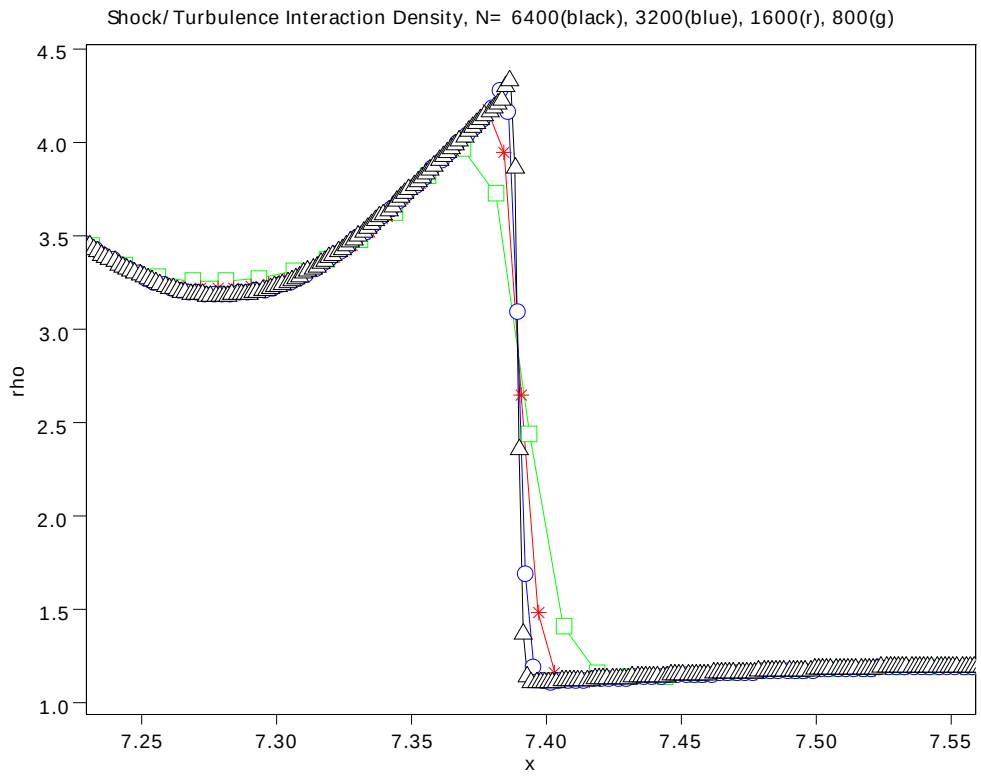

Figure 5.7: Shock-Turbulence interaction at time t = 1.8. Zoomed in view at the shock wave. Results for four different grid resolutions are compared: $N = 800$, $N = 1600$, $N = 3200$, and $N = 6400$. A linear MINMOD reconstruction in all conserved quantities is used.

The simulation is stopped before the flow interacts with the boundary. Units are in CGS. The acoustic time step (5.1) with $CFL = 1/2$ was used.

$$\begin{cases} \text{JWL Gas: } \rho = 1.63, e_{int} = 4.2814E+10, u = 0, p = 7.81E+10, & x \leq 200 \\ \text{Water: } \rho = 1.0, e_{int} = \text{N/A}, u = 0, p = 1.00E+6, & x > 200 \\ x \in [0, 400] \end{cases} \tag{5.16}$$

Results are shown at resolution $N = 400$ for comparison with Wardlaw [83]. Some overshoot at the shock front can be seen. Mass in the liquid and gas regions is conserved to $\mathcal{O}(10^{-12})$. Comparison can be made to shock capturing method of Fedkiw, et. al [23], which eliminates the

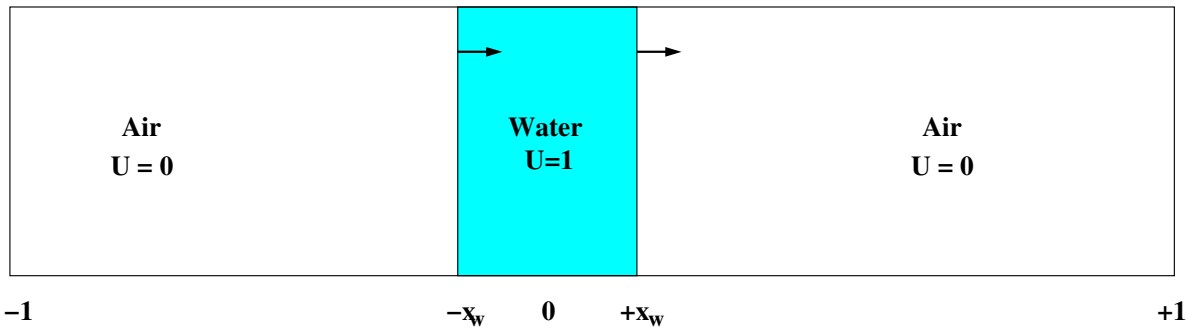

Figure 5.8: Initial configuration of the oscillating water column problem. Movement of the water compresses the air, causing flow to slow down and reverse direction.

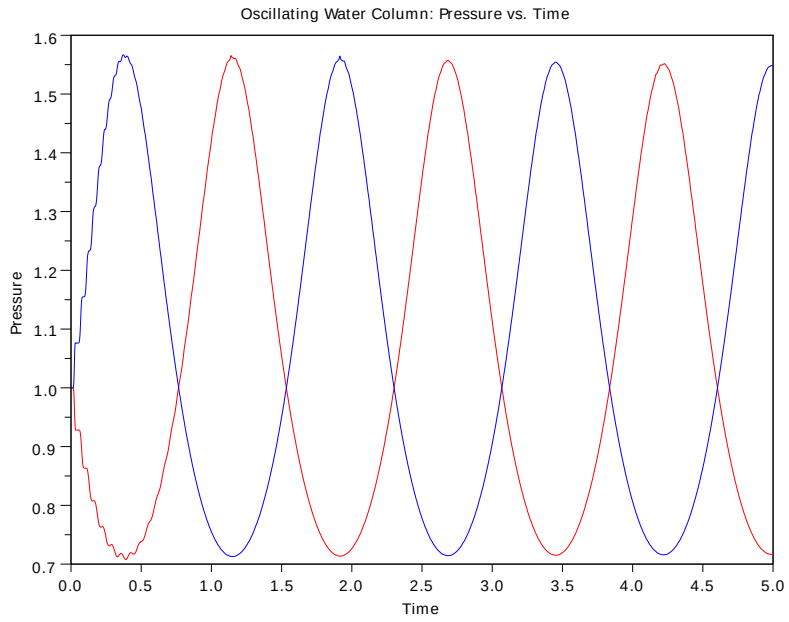

Figure 5.9: Pressure at the ends of the oscillating water column. The blue line indicates pressure at the right end, red indicates pressure at the left end. A linear MINMOD reconstruction for all conserved quantities is used.

overshoot at the shock front by extrapolating material to a "ghost cell." However, doing so is non-conservative and requires a correction to capture the correct shock speed.

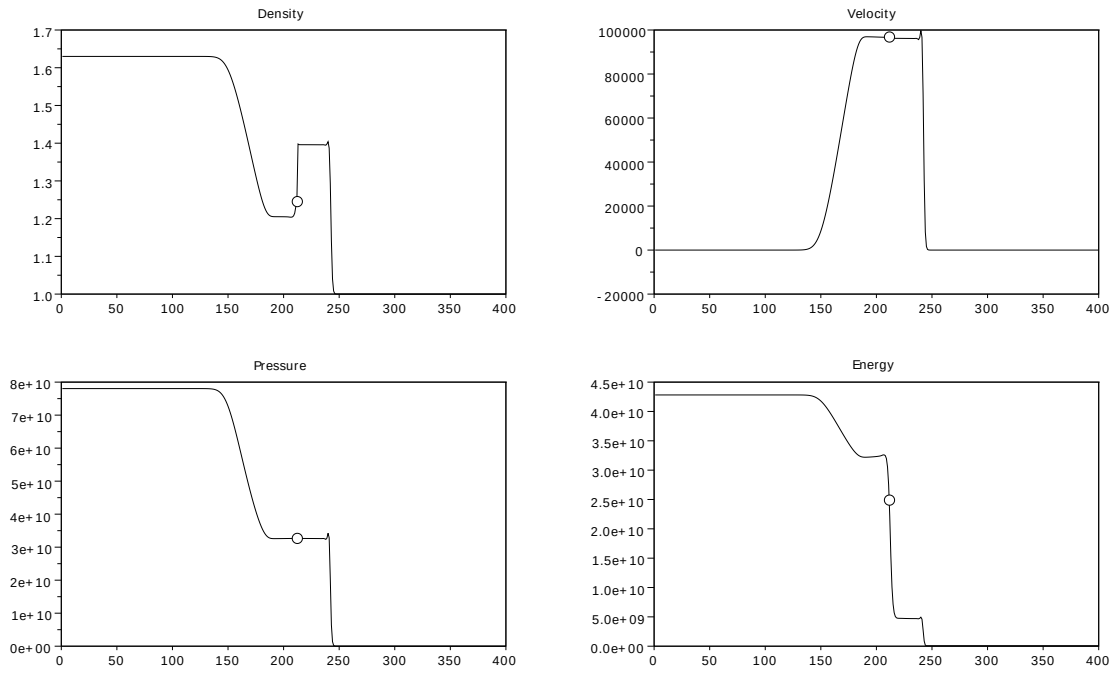

Figure 5.10: Results for the JWL-Tait shock tube at time $t = 0.125$. The material interface is shown as a circle. A linear MINMOD reconstruction for all conserved quantities is used in each cell.

5.3 2D Support Operator-Finite Volume Projection Operator Verification

The projection-based gradient operator proposed in Section 4.3.4 is tested against a number of validation problems. The method is tested using Poisson's equation with variable coefficients and homogeneous Neumann or periodic boundary conditions.

As in [12], the solution is observed to converge with second-order accuracy. The L^1 error norms for the solution E_N and cell-centered gradient G_N for a mesh with N triangular elements. Let $f(\boldsymbol{x})$ denote the exact solution, s_i denote the computed solution in cell i, and g_i denote the computed gradient at the centroid of cell Ω_i. The exact gradient is evaluated at the centroid $\boldsymbol{c}_i$ of cell Ω_i. The solution error is computed as

$$E_N = \sum_{i=1}^{N} ||f(\boldsymbol{c}_i) - s_i||_1 V_i, \tag{5.17}$$

and the gradient error over the domain is computed as

$$G_N = \sum_{i=1}^{N} ||\nabla f(\boldsymbol{c}_i) - g_i||_1 V_i, \tag{5.18}$$

where the computed gradient is computed as in (4.77).

We are solving the PDE (5.19) on the unit square with homogeneous Neumann or periodic boundary conditions, where $\boldsymbol{x}_p$ is the periodic extension of the point $\boldsymbol{x}$.

$$\begin{cases} Ap - \nabla(D(\boldsymbol{x})\nabla p) = s(\boldsymbol{x}), & \boldsymbol{x} \in [0,1]^2 \\ \frac{\partial p}{\partial n} = 0 & \boldsymbol{x} \in \partial\Omega_N \\ p(\boldsymbol{x}) = p(\boldsymbol{x}_p), & \boldsymbol{x} \in \partial\Omega_P. \end{cases} \tag{5.19}$$

Table 5.4 shows the L^1 error for the cell-centered solution and gradient values, along with convergence rates $\tau(E_N)$ and $\tau(G_N)$ for the solution and cell-centered gradients, respectively, for Poisson's problem (5.20) with periodic boundary conditions on all edges. Parameters and solution p are as specified in (5.20).

$$\begin{aligned} &A = 0 \\ &D(x,y) = 1 \\ &s(x,y) = 8\pi^2 \cos(2\pi(x-0.1))\cos(2\pi(y+0.1)) \\ &p(x,y) = \cos(2\pi(x-0.1))\cos(2\pi(y+0.1)). \end{aligned} \tag{5.20}$$

Second-order convergence in the solution and first-order convergence of the cell-centered gradient is observed.

Table 5.4: Errors and convergence rates for periodic boundary value problem (5.19) with parameter (5.20). Second-order convergence for the solution and first order convergence for the derivative are observed.

Elements	E_N	$\tau(E_N)$	G_N	$\tau(G_N)$
128	2.133e-02	—	6.497e-01	—
512	5.554e-03	1.94	2.740e-01	1.25
2048	1.413e-03	1.97	1.299e-01	1.08
8192	3.560e-04	1.97	6.292e-02	1.05
32768	9.074e-05	1.97	3.111e-02	1.02

Table (5.5) shows the L^1 error for the cell-centered solution and gradient values, along with convergence rates $\tau(E_N)$ and $\tau(G_N)$ for the solution and cell-centered gradients, respectively, for

Helmholtz's equation with homogeneous Neumann boundary conditions on all edges. Parameters and solution p are as specified in (5.21).

$$\begin{aligned} A &= 1 \\ D(x,y) &= 1 + x \\ s(x,y) &= 21x^2 - 2x^3 - 6 \\ p(x,y) &= x^2(3 - 2x). \end{aligned} \tag{5.21}$$

Second-order convergence in the solution and first-order convergence of the cell-centered gradient is observed. These results are novel in that a cell-centered gradient operator, consistent with the

Table 5.5: Errors and convergence rates for homogeneous Neumann boundary value problem (5.19) with parameters (5.21).

Elements	E_N	$\tau(E_N)$	G_N	$\tau(G_N)$
128	1.563e-02	—	6.923e-02	—
512	3.906e-03	2.00	3.404e-02	1.02
2048	9.766e-04	2.00	1.695e-02	1.01
8192	2.441e-04	2.00	8.467e-03	1.00

definition of the Support Operator-Finite Volume Laplacian operator [12], can be recovered using cell-centered degrees of freedom. This operator is applied to problems with homogeneous Neumann and periodic boundary conditions, and it is demonstrated to be first-order accurate.

5.4 New Moment-of-Fluid Initialization Technique

Choosing an accurate initial guess for any gradient descent-based optimization procedure is important, to prevent the method from getting terminating in a non-optimal local minimum. For the Moment-of-Fluid method, the an initial angle (determined by the outward unit normal $\widehat{n}$) is proposed as the orientation of the interface. On triangular meshes, the standard initialization procedure

$$\widehat{n} = \frac{\boldsymbol{x}_i - \boldsymbol{x}_{ref}}{|\boldsymbol{x}_i - \boldsymbol{x}_{ref}|} \tag{5.22}$$

can result in a poor guess that will lead the method to converge to a non-optimal solution. For tests that rely on physics at the interface, this can lead to non-physical behavior. All results in this work use the initialization procedure described in Section 3.1.5.

5.4.1 Stationary Interface Reconstruction

The results for the new technique for initializing the optimization procedure for the Moment-of-Fluid method on triangular meshes is presented. First, exactness of the method is demonstrated in Cartesian coordinates. This is compared to the initial guess produced by the standard method (5.22) in the literature. The results after convergence of the Gauss-Newton (Section 3.1.1) method are presented, demonstrating that use of (5.22) can result in convergence to some local minimum, rather than the global minimum.

A set of uniform nodes n_i on the unit square $[0,1]^2$ is initialized, and each internal node is perturbed by some random value $\gamma_i \Delta x/4$, for $\Delta x = 0.1$ and some random number $\gamma_i \in [-1,1]$. Two stationary problems are tested, with the interface defined as the zero level curve of the function $\Gamma_i(x,y)$.

$$\Gamma_1(x,y) = -0.5x + 0.75 - y \tag{5.23}$$

$$\Gamma_2(x,y) = 0.55 - \sqrt{x^2+y^2} \tag{5.24}$$

For both tests, 121 nodes are distributed uniformly on the unit square, with the perturbation parameter taken to be $\Delta x = 1/10$. No restarting of the MOF optimization procedure is used. Table 5.6 shows the error and average number of Gauss-Newton iterations to convergence for the Moment-of-Fluid reconstruction of the linear interface $\Gamma_1 = 0$ (5.23). This result demonstrates that the method proposed is exact for linear interfaces and outperforms the standard initialization (5.22) for the MOF optimization procedure. Additionally, it is observed that using the standard initial guess can result in exiting with non-optimal error, due to the gradient-descent optimization procedure terminating in a local minimum. Figures 5.11 - 5.12 show the initial interfaces prescribed by the proposed and standard MOF initialization methods, respectively.

Table 5.7 shows the average error and number of Gauss-Newton iterations to convergence per cut cell for MOF reconstruction of the curved interface $\Gamma_2 = 0$ (5.24). Again, we see that the newly proposed initialization procedure results in a lower error than using the standard method. It is noted that the newly proposed method performs robustly even for problems with a reference volume fraction close to one, in which case the optimization problem becomes stiff. Additionally,

the proposed method does not demonstrate sensitivity to the relative orientation of the interface to the edges of the triangle. This method is used in all further 2D unstructured, multi-material tests.

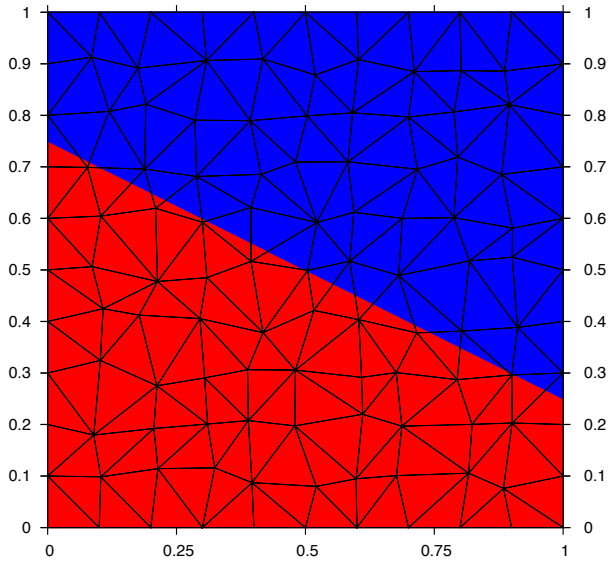

Figure 5.11: Initial Moment-of-Fluid interface orientation for the linear interface (5.23) using the new method described in Section 3.1.5. Average error $E_{MOF} = 4.911\text{e-}10$ over 24 cut cells. $\Gamma_1(x, y) > 0$ is shown in red, and $\Gamma_1(x, y) < 0$ is shown in blue.

Table 5.6: Results for the Moment-of-Fluid reconstruction of Γ_1 (5.23) Average initial and final errors for the two Moment-of-Fluid interface initialization techniques, with the average number of iterations to convergence per cut cell, over 24 cut cells.

Method	Avg. Initial E_{MOF}	Avg. Final E_{MOF}	Avg. Gauss- Newton Iterations
Standard (5.22)	1.653e-3	8.564e-4	6
Proposed (Section 3.1.5)	4.911e-10	4.911e-10	0

5.4.2 Moment-of-Fluid Initialization in Action

It is observed that the new initialization procedure performs well on stationary interface. Moving interfaces, which deform over time, present new challenges. Small cut-off volumes, referred to as "flotsam," can separate and re-attach to the material volume, typically due to errors in the advection procedure. When these pieces of flotsam reattach, they perturb the orientation of the interface. A

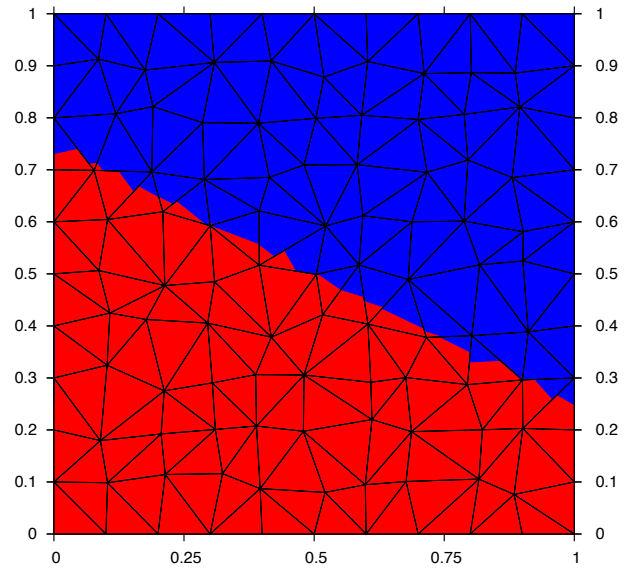

Figure 5.12: Initial Moment-of-Fluid interface orientation using the standard method (5.22). The reference material is taken as the material with the largest volume fraction. E_{MOF} = 1.653e-3 over 24 cut cells.

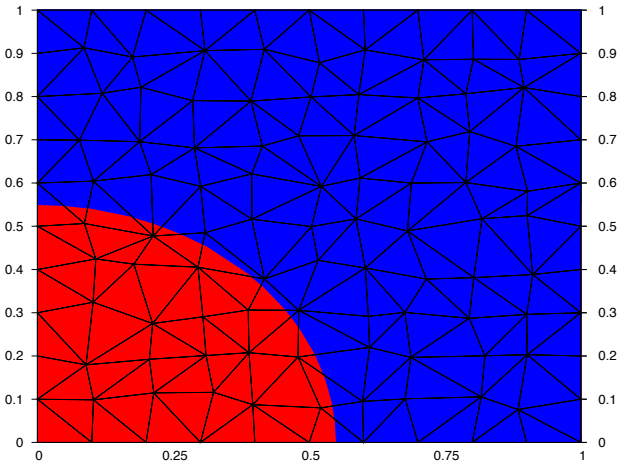

Figure 5.13: Initial Moment-of-Fluid interface orientation for the linear interface (5.24) using the new method described in Section 3.1.5. Average error E_{MOF} = 2.408e-6 over 17 cut cells. $\Gamma_1(x, y) > 0$ is shown in red, and $\Gamma_1(x, y) < 0$ is shown in blue.

robust interface reconstruction technique will not display undue sensitivity to these small volume pieces of flotsam.

These two standard benchmark tests, the reversible vortex rollup and the rotation of the notched

Table 5.7: Results for the Moment-of-Fluid reconstruction of curved interface Γ_2 (5.23) Average initial and final errors for the two Moment-of-Fluid interface initialization techniques, with the average number of iterations to convergence per cut cell, for 17 cut cells..

Method	Avg. Initial E_{MOF}	Avg. Final E_{MOF}	Avg. Gauss- Newton Iterations
Standard (5.22)	2.304e-3	1.130e-3	6
Proposed (Section 3.1.5)	2.408e-6	2.408e-6	0

disk, demonstrate that the newly proposed initialization procedure performs well in the context of moving, deforming interfaces, in addition to previously presented stationary problems.

5.4.2.1 Reversible Vortex Rollup. The initial material configuration is defined by a circle of radius R = 0.15 with center $(0.50, 0.75)$ located within the unit square domain $[0,1] \times [0,1]$. The circular region is deformed by a non-linear, unsteady velocity field given by stream function (5.25).

$$\psi(x,y,t) = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y)\cos\left(\frac{\pi t}{T}\right) \tag{5.25}$$

Parameter T defines the time for the full period of the flow field, with larger T leading to greater deformation. For the long period reversible vortex, take $T = 8$. Time $t = T$ would see the analytical solution returning to the initial material configuration. Time $t = \frac{T}{2}$ defines the time of maximum material deformation. Results are shown for a perturbed rectangular mesh, with $\Delta x = 1/128$, with uniformly distributed nodes n_i perturbed by some random number $\gamma_i \Delta x/6$, for some random number $\gamma_i \in [-1, 1]$. It is observed in image 5.14, that the thin roll-up structure is captured with minimal breakup. Flotsam is seen near the thin tail of the roll-up, generated by the accumulation of interface reconstruction and advection errors. The new MOF initialization procedure appears to perform robustly for this test case, in the presence of flotsam and break-up.

5.4.2.2 Rotating Notched Disk. The initial material configuration is defined by a circle of radius R = 0.5 with center $(0, 1)$ on the square domain $[-2, 2]^2$. A rectangular notch $[-w, w] \times [0, 1]$ is removed from the circular disk, with width parameter $w = 1/16$. The velocity field (5.26) is applied, rotating the notched disk in a counter-clockwise direction around the domain.

$$\boldsymbol{u}(x,y,t) = \begin{pmatrix} -y \\ x \end{pmatrix} \tag{5.26}$$

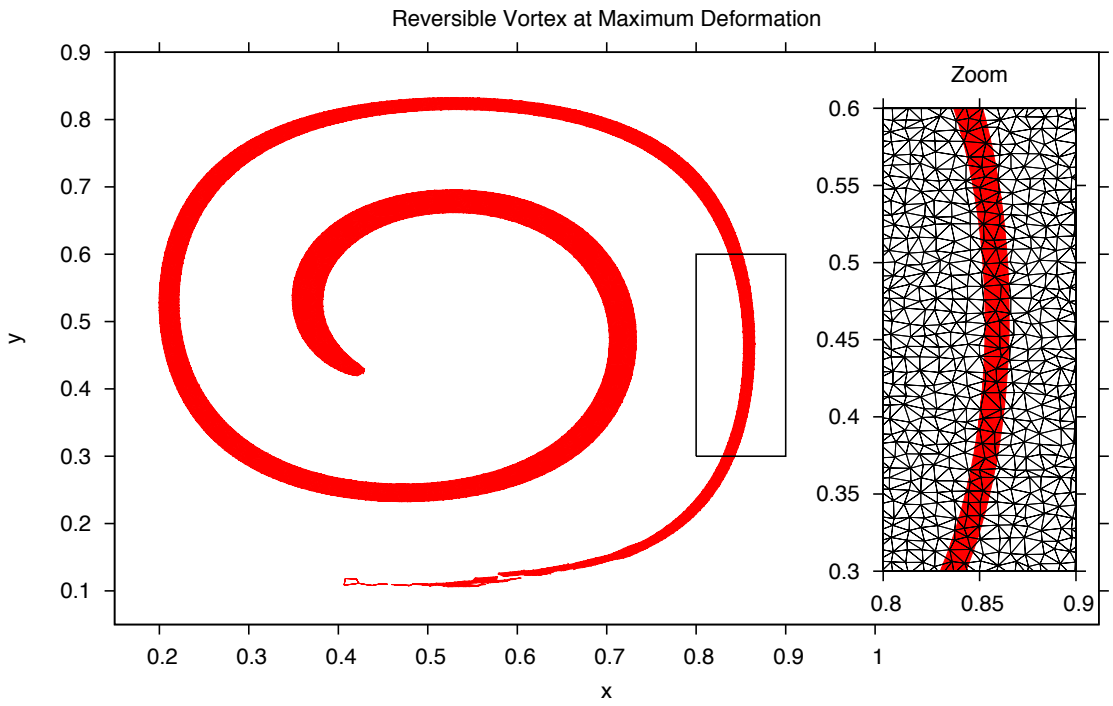

Figure 5.14: Interface reconstruction of disk material at maximum deformation, time $t = 4$, with a close-up of the thin region on the unstructured grid. No filament capturing is used. Note the break-up in the thin structure of the tail as it is under-resolved by the mesh. Note the presence of "flotsam", erroneous, small cut-off volumes disconnected due to accumulated advection and reconstruction errors.

Capturing the sharp corners of the notched disk can be difficult for PLIC methods, as a corner in a cell cannot be captured exactly as a single linear interface. This will result in "diffusion" of material as the reconstruction errors are propagated in time. See Figures 5.15 - 5.16, which shows the initial reconstruction of the notched disk, and the reconstructed notched disk after one full revolution. Note that the sharp tips of the disk are rounded, due to propagating compounding errors of the PLIC reconstruction. Again, the new MOF initialization procedure performs well, capturing the linear regions in the notch and the circular structure of the disk.

Remarks:

- The left sharp corner where the rectangular notch intersects the perimeter of the disk is not captured due to how the level-set function defining the disk cuts the unstructured mesh. The zero level-curve cuts the cell such that the only one edge of the cell is cut. The MOF initialization algorithm (see Sec. 3.1.3) does not flag the cell as cut, because the level-set function has the same sign at all vertices of the cell.

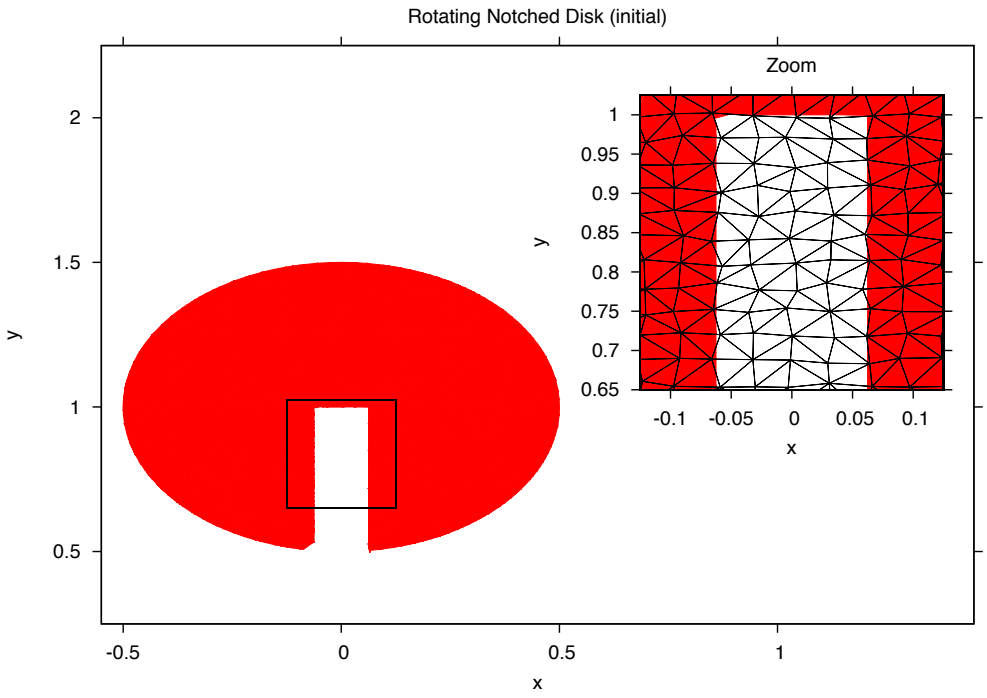

Figure 5.15: Initial Moment-of-Fluid interface reconstruction of the notched disk at time $t = 0$, with a close-up view of the unstructured mesh near the top of the notched region.

5.5 New Filament Capturing Technique

The novel filament capturing technique proposed in Section 3.3 uses the Multi-material Moment-of-Fluid method along with a geometrically-based conglomeration procedure to detect and capture filaments. When a potential filament is detected, the interface reconstruction is posed as finding two linear interfaces that separate three materials, rather than a single linear interface that separates two materials. The filament-capturing Moment-of-Fluid method then reconstructs the material region on either side of the filament, rather than reconstructing the filament itself. By doing so, the cost of the filament capturing method is asymptotically constant, even as the width of the filament goes to zero.

Using the filament-capturing method, highly deformational interfacial flows can captured well, without resorting to introducing surface Lagrangian particles, as in [89].

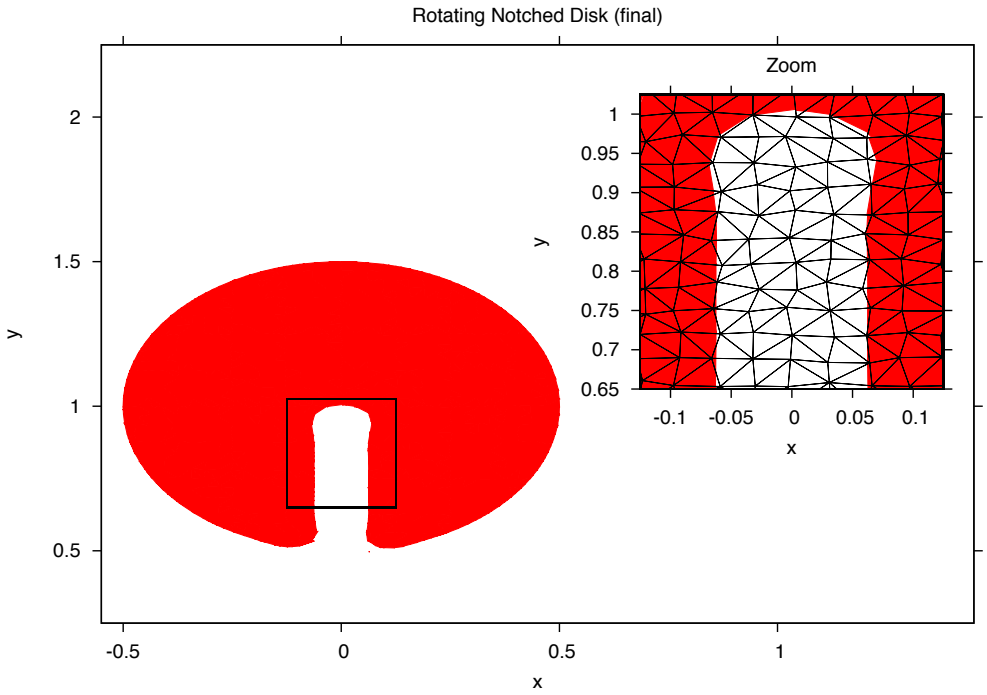

Figure 5.16: Initial Moment-of-Fluid interface reconstruction of the notched disk at time $t = 2\pi$ after one full revolution, with a close-up view of the unstructured mesh near the top of the notched region. Note the mass diffusion near the sharp corners, which cannot be accurately captured by a PLIC method.

5.5.1 Reversible Vortex

The set-up is identical to the reversible vortex in Section 5.4.2.1. The circular region is deformed by a non-linear, unsteady velocity field given by stream function

$$\psi(x, y, t) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right). \tag{5.27}$$

Parameter T defines the time for the full period of the flow field, with larger T leading to greater deformation. For the long period reversible vortex, take $T = 8$. Time $t = T$ would see the analytical solution returning to the initial material configuration. Time $t = \frac{T}{2}$ defines the time of maximum material deformation. Taking a large reversal period T will ensure that a filament forms when the material is at maximum deformation. Unlike results shown in Section 5.4.2.1, computation is performed on a structured square mesh with filament capturing used to eliminate breakup in the

Table 5.8: Results for the long-period deforming vortex at full reversal. Starting mesh resolution is 32^2. The standard MOF algorithm takes longer than the filament MOF algorithm when number of AMR levels is 2 because the moment error is large in filamentary regions and requires a larger patch of the computational domain to be fully adapted.

	AMR Level	Error	Runtime (sec)
Filament MOF	0	3.122×10^{-3}	59.316
	1	6.380×10^{-4}	480.147
	2	2.329×10^{-4}	1941.891
Standard MOF	0	1.990×10^{-2}	53.992
	1	4.809×10^{-3}	438.114
	2	8.790×10^{-4}	2306.643

thin structure. On a square mesh, the standard initial interface normal direction

$$\widehat{n} = \frac{\boldsymbol{x}_i - \boldsymbol{x}_{ref}}{|\boldsymbol{x}_i - \boldsymbol{x}_{ref}|} \tag{5.28}$$

is used.

Error is taken as the symmetric difference error (5.29), with Ω_E and Ω_C representing the exact and computed material configurations, respectively.

$$E_{SD} = |(\Omega_E \cup \Omega_C)/(\Omega_E \cap \Omega_C)| \tag{5.29}$$

Symmetric difference error at full-reversal is shown in Table 5.8. At coarser grid levels, when the number of cells marked for refinement is comparable between the Standard MOF and MOF with filament capturing, the additional overhead of the conglomeration procedure leads to a larger runtime. Faster runtime for MOF with filaments is observed when two levels of AMR are used, because there are fewer Level 2 grid cells (see Fig. 5.17 and Fig. 5.19).

The advantage of using filament capturing is seen in the gains in accuracy at a given mesh refinement level. Symmetric difference error (5.29) when using filament capturing at a given level of mesh refinement is comparable (slightly lower) than the error calculated from using the standard MOF reconstruction with an additional level of mesh refinement.

5.5.2 Rotating Filament

A finite, rectangular filament of length 30 and width 0.6 is initialized with center at the point $(50.5, 75)$ in the domain $\Omega = [0, 100]^2$,. A velocity field is applied to rotate the filament counter-clockwise about the center of the domain. Initial mesh resolution is 50×50. The top and bottom

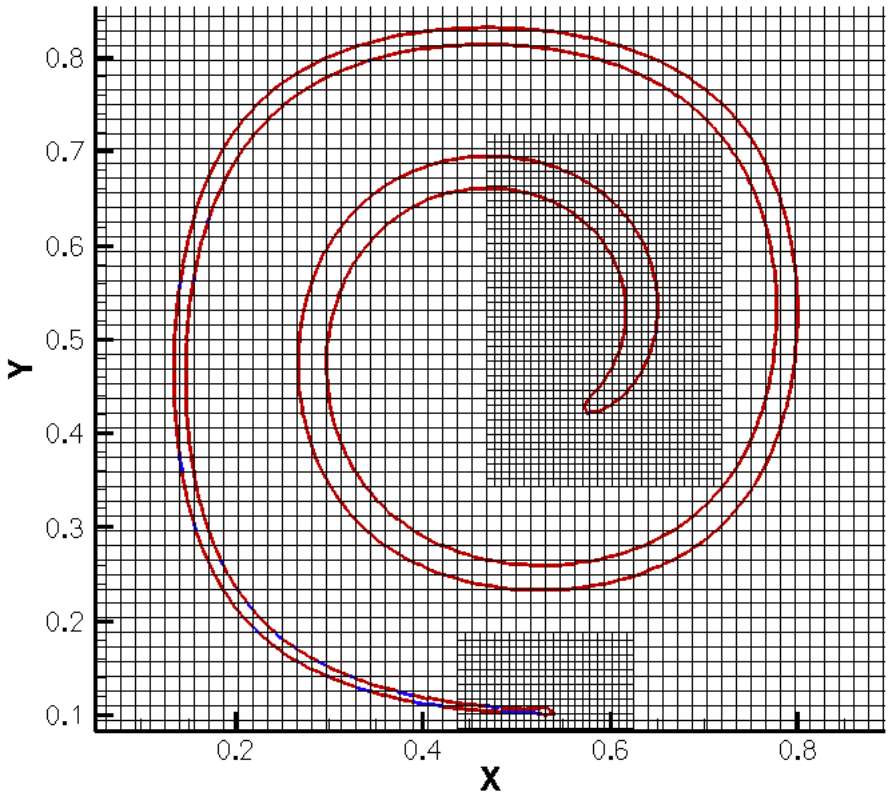

Figure 5.17: Reversible vortex shown at maximum deformation for $N = 32$ grid resolution and 2 levels of AMR with filament machinery. The blue segments represent the interface between material 1 and material 2. The red segments represent the interface between the "twin" third material and material 2.

ends of the filament are initialially aligned with the mesh. Two levels are AMR are used to resolve the tips of the filament. Note that while the center of the filament is under-resolved by the mesh, AMR is not triggered because the reconstruction error is below the AMR error threshold.

Figure 5.21 shows the filament after completing 1/8 of its revolution. Notice that no breakup is exhibited, even though the filament is under-resolved by the mesh. AMR is triggered by reconstruction errors only in a narrow region around the tips of the filament. Figures 5.22 - 5.23 show the tip of the filament before and after completing one revolution. Error in the reconstruction, manifested as mass diffusion near the tip of the filament, is due to the inability to exactly resolve a corner within a cell. Note that while this diffusion has caused the rotated filament near the tip to become wider, the final, rotated filament is shorter. Mass is well-conserved, with a relative final

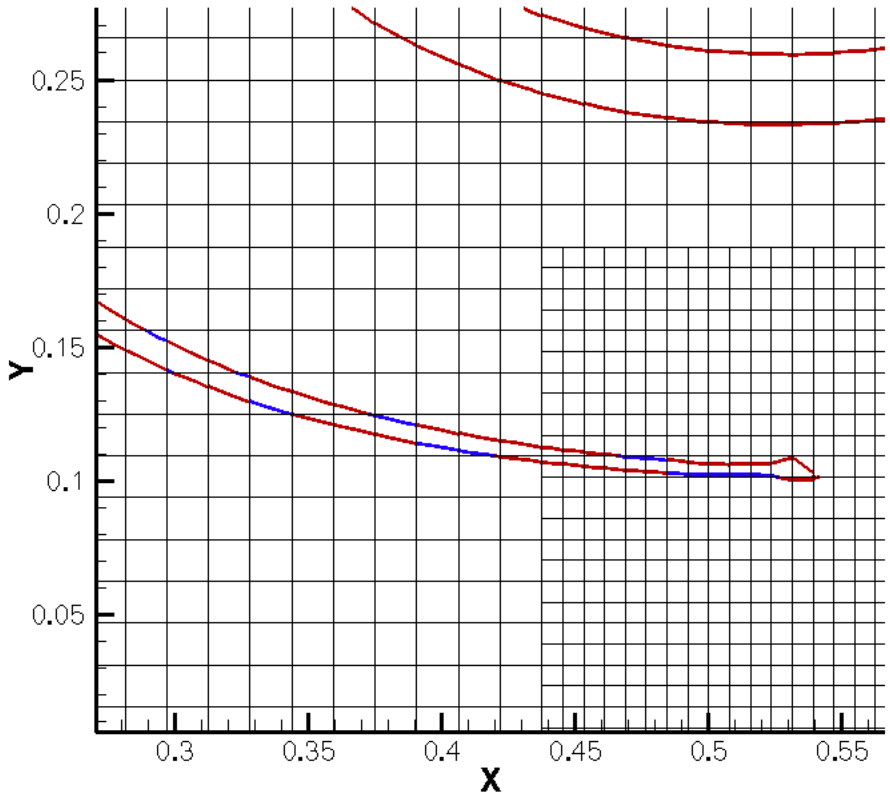

Figure 5.18: Reversible vortex shown at maximum deformation for $N = 32$ grid resolution and 2 levels of AMR with filament machinery, zoomed at the thin tail of the interface. Note that the thin structure is accurately captured.

mass loss in the filamentary material equal to 5.26×10^{-9}.

5.5.3 Droplet Flow

A nonlinear, divergence-free velocity field acts upon a circular region, causing edges to shear and form a filament. At time $t = 0$, the circular region is located in the center of the $[0, 1]^2$ domain, with initial radius $R_0 = 0.125$. This test case, introduced in [3], presents a different challenge from the test case in Section 5.5.1. In this test case, filaments form as the circular region is deformed into a droplet shape with two sharp edges. The filaments at these sharp tips must be accurately captured to prevent spurious breakup.

Flow runs to maximum deformation until time $T_{maxdef} = 0.75$, at which point the flow undergoes a smooth transition to reversal for $t_\epsilon = 0.1$, and runs to the full period time $T = 1.6$.

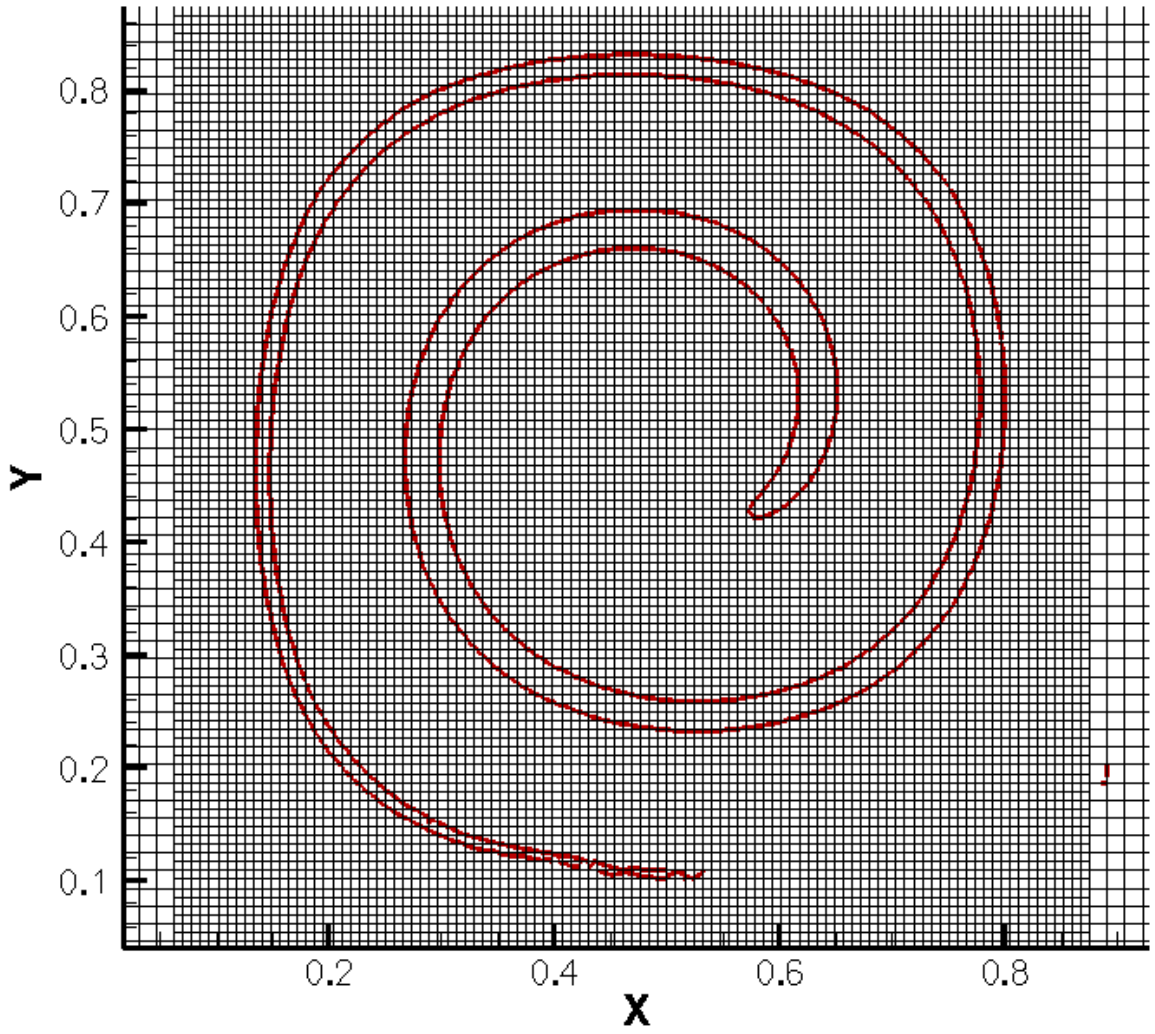

Figure 5.19: Reversible vortex shown at maximum deformation for $N = 32$ grid resolution and 2 levels of AMR using standard Moment of Fluid. The red segments represent the interface between material 1 and material 2. Note that much more of the grid is refined than when using MOF with filament capturing, leading to longer runtime for the same effective resolution.

The following reversible velocity field is used, with error calculated at full reversal when the exact solution is the initial material configuration, a circle.

$$\boldsymbol{u}(x,y,t) = \begin{bmatrix} \frac{1}{8}(8x-4) \\ \frac{1}{8}\left\{-(8y-4)-4-\left(1-(8x-4)^2-(8x-4)^4\right)\right\} \end{bmatrix} f(t) \tag{5.30}$$

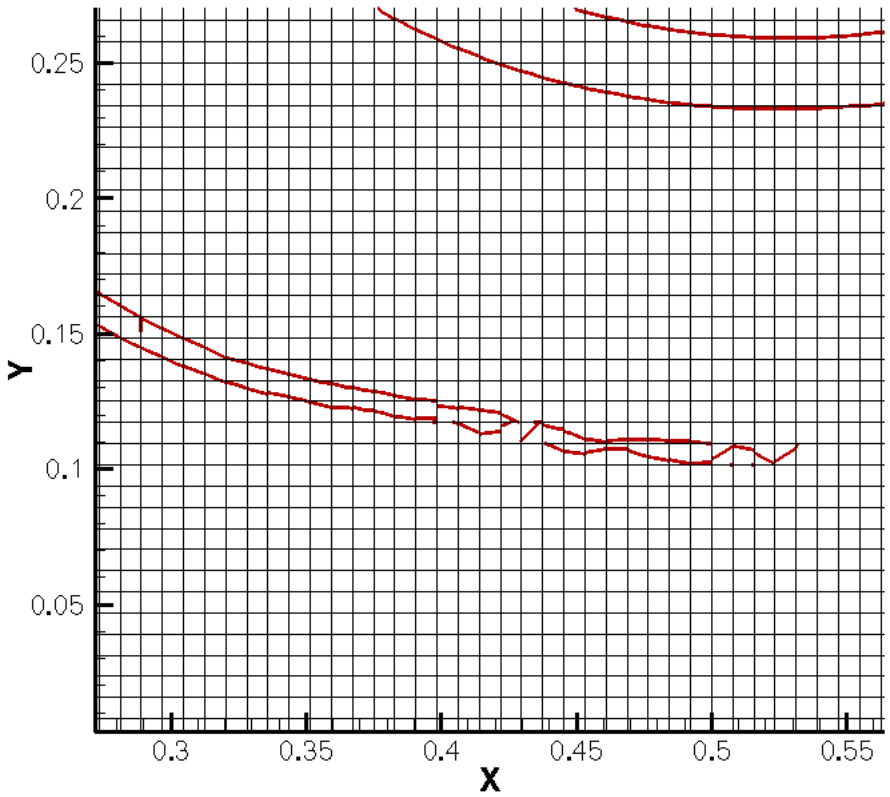

Figure 5.20: Reversible vortex shown at maximum deformation for $N = 32$ grid resolution and 2 levels of AMR using standard Moment-of-Fluid, zoomed at the thin tail of the interface. Note the break-up of the thin structure.

$$f(t) = \begin{cases} 1, & 0 \le t < T_{maxdef} \\ \cos\left(\frac{\pi(t-T_{maxdef})}{t_\epsilon}\right), & T_{maxdef} \le t \le T_{maxdef} + t_\epsilon \\ -1 & T_{maxdef} + t_\epsilon < t \le T \end{cases} \tag{5.31}$$

5.5.4 S-shape Flow

A nonlinear, divergence-free velocity field acts on a circular region, initially located at the center of a $[0,1]^2$ domain with radius $r_0 = 0.25$. The velocity field, introduced in [3], causes the circular region to roll-up into an S-shape. This test case presents the challenge that filaments of each material are formed. The following reversible velocity field is used, with parameters $T_{maxdef} = 3.0$,

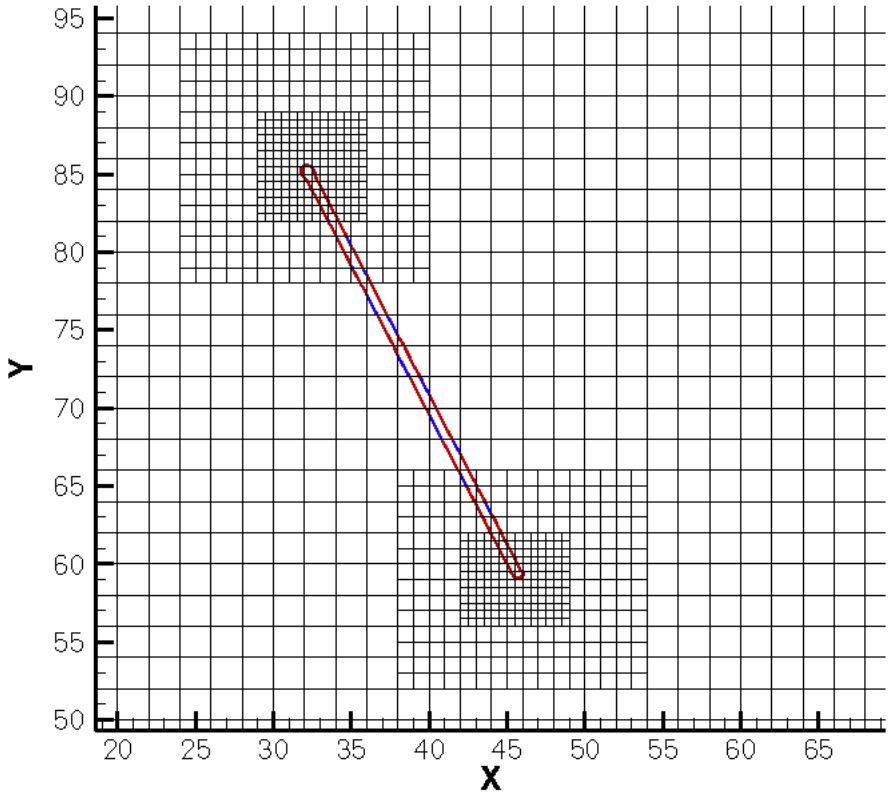

Figure 5.21: Rotating filament after 1/8 of a revolution. A blue interface indicates the introduction of a twin material to capture a filament in the cell. Two levels of AMR are used to resolve the tips of the filament.

$t_{epsilon} = 2.0$, and $T = 8.0$. Function $f(t)$ defining velocity field reversal is

$$\boldsymbol{u}(x,y,t) = \begin{bmatrix} \frac{1}{4}\left\{(4x-2)+(4y-2)^3\right\} \\ \frac{-1}{4}\left\{(4y-2)+(4x-2)^3)\right\} \end{bmatrix} f(t) \tag{5.32}$$

Again, use of filament capturing produces error results comparable to using the standard Moment-of- Fluid method with an additional level of mesh refinement, while the increase in cost associated with filament capturing at a given grid resolution is much lower than adding an additional level of refinement. Note that at one level of mesh refinement (Fig. 5.27 - 5.29), filaments of each material are formed. This configuration is resolved when using filament capturing (Fig. 5.27 - 5.28), while spurious break-up is observed when filament capturing is not applied (Fig. 5.29)

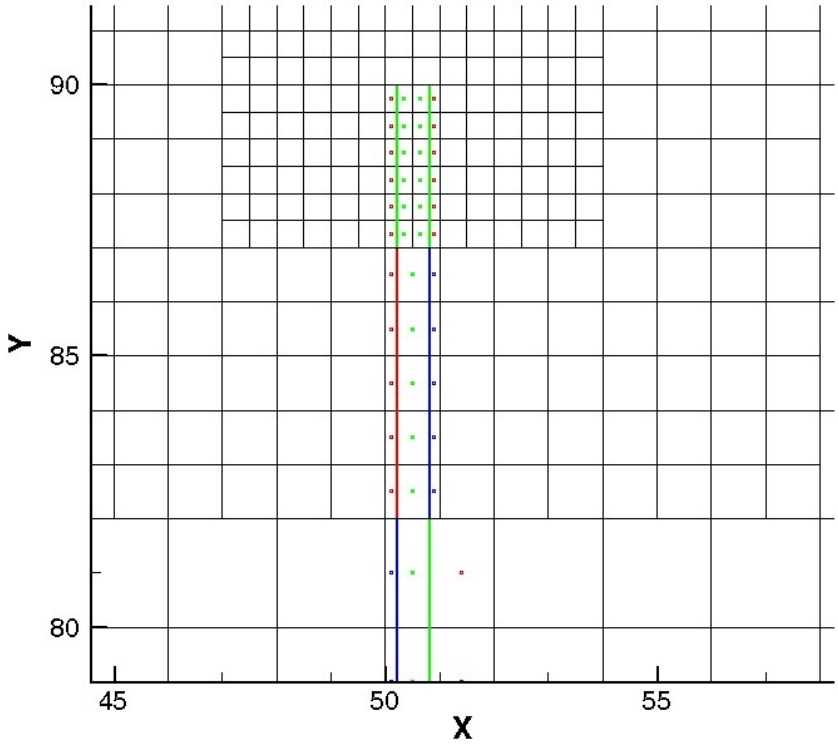

Figure 5.22: The tip of the filament before revolution. Interfaces and centroids are shown for each material.

5.5.5 Pulsating Membrane

We introduce a new test problem to demonstrate the unique functionality of this new method. A thin, circular shell with outer-radius $r_0 = 1.0$ and inner-radius $r_1 = 1.0 - \epsilon$ is initially located at the center of the $[-2, 2]^2$ domain. A nonlinear, divergence-free velocity field causes expansion in the horizontal direction and compression in the vertical direction, until time $T/2$, at which point the flow reverses and the material returns to its initial configuration. The use of the Moment-of-Fluid method with filament capturing is able to capture the periodic motion of a thin, moving interface, under-resolved in nearly every cell, without causing break-up. The velocity field is given by (5.33). Here, period time $T = 1.0$, and the membrane has thickness $\epsilon = 0.02$. Mesh resolution is 32^2.

$$\boldsymbol{u}(x, y, t) = \begin{bmatrix} \sin(\pi/2x)\cos(\pi/2y)\sin(2\pi t) \\ -\cos(\pi/2x)\sin(\pi/2y)\sin(2\pi t) \end{bmatrix} \tag{5.33}$$

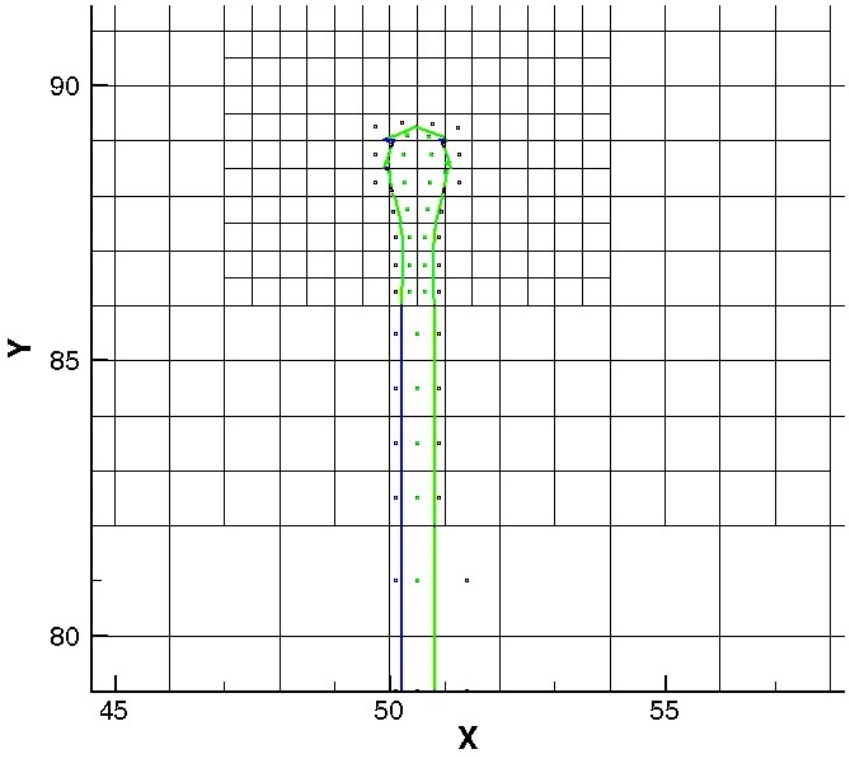

Figure 5.23: The tip of the filament after revolution. Diffusion of mass has occurred near the tip, but the body of the filament is accurately captured.

5.6 Unstructured 2D Fluid Algorithm Testing

The tests in this section address various aspects of the fluid algorithm on 2D unstructured meshes. Accuracy is examined for an incompressible single material test (Sec. 5.6.1) and comparison is made to previously published incompressible two-material tests (Sec. 5.6.2). Compressible tests are run for one and two materials in Cartesian coordinates. Finally, the Spherical Explosion Shock test is examined in 2D unstructured curvilinear coordinates.

5.6.1 Single Fluid Horizontal Shear Layer

The CISL advection procedure and incompressible Finite Element and Support Operator Finite Volume methods are tested on a standard benchmark test, evolution of an inviscid horizontal shear layer [75]. The problem is computed on the doubly periodic 1×1 box, with the initial velocity

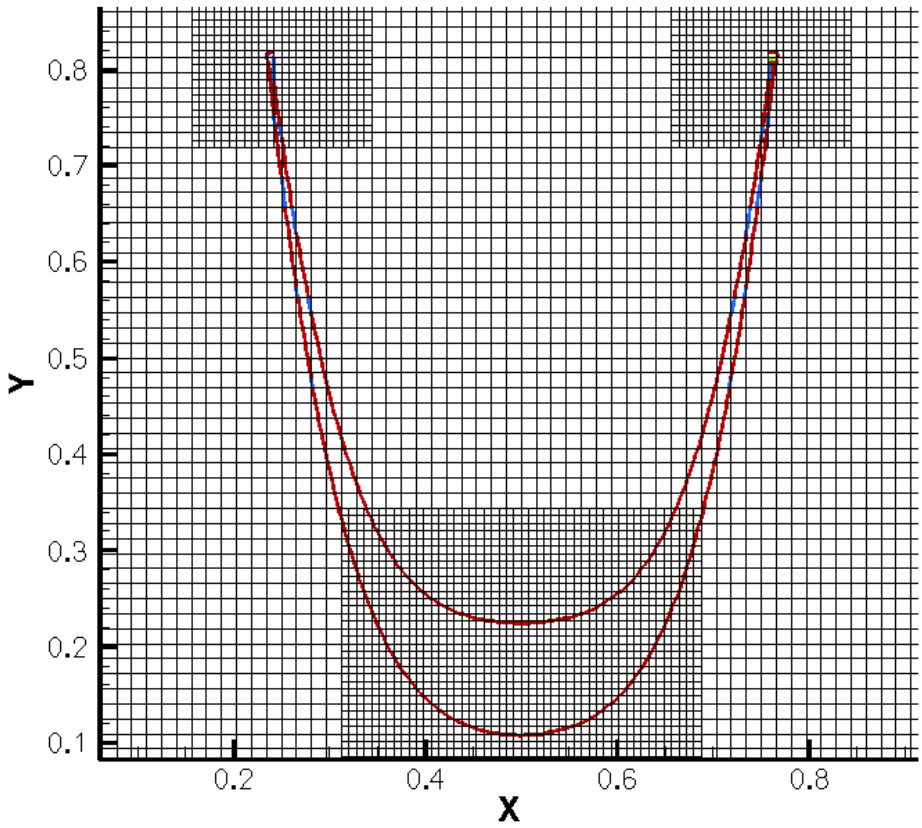

Figure 5.24: Droplet flow at maximum deformation, using filament capturing, with 2 levels of AMR.

profile given by

$$u_0(x,y) = \begin{cases} \tanh(30(y-1/4)), & y \le 1/2 \\ \tanh(30(3/4-y)), & y > 1/2 \end{cases} \tag{5.34}$$
$$v_0(x,y) = (1/20)\sin(2\pi x).$$

Given the initial flow profile, two vortices form. Vorticity contours are plotted at the same times as in [75], for comparison with previously published results.

While the unstructured Finite Element and Finite Volume methods presented are second order, accuracy will be limited by the accuracy of the CISL advection procedure. While a piecewise-linear density reconstruction is used in 1D, a piecewise-constant reconstruction in all state variables is implemented on 2D unstructured grids. This will be more diffusive, leading to degradation to first order accuracy for smooth problems. Additionally, the cell-centered gradient operator, used

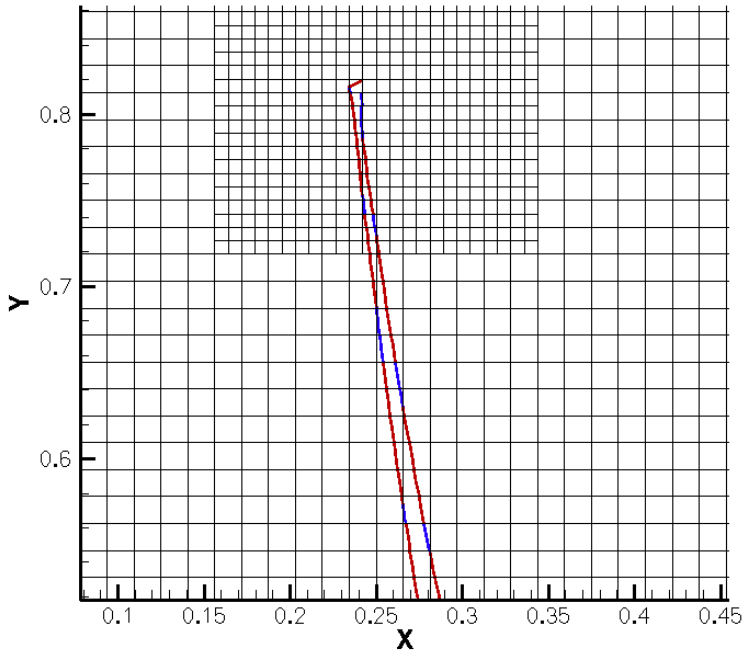

Figure 5.25: Droplet flow at maximum deformation, using filament capturing, with 2 levels of AMR. The view at the sharp tip is enlarged. Note that the thin structure is accurately captured.

to update momentum and energy, is only first order accurate. As such, approximate first order accuracy for this problem is expected. In Figure 5.31, this is manifested as roughness in the vorticity contours and noise in the domain. Table 5.11 shows convergence of nodal vorticity. The convergence rate is seen to be approximately first order, as predicted.

Remarks:

- The mesh is generated by uniformly distributing nodes over the domain and triangulating the result, so the mesh is pseudo-structured.
- At each node, an inverse distance-weighted linear least squares reconstruction of each velocity component is calculated, using all cells that share the given node.
- Given a linear reconstruction of $\boldsymbol{U} = (u(x, y), v(x, y))$, the vorticity is calculated as the curl of $\boldsymbol{U}$.
- Because mesh nodes have a logically rectangular structure, coarse and fine node values can be compared directly without use of data interpolation.

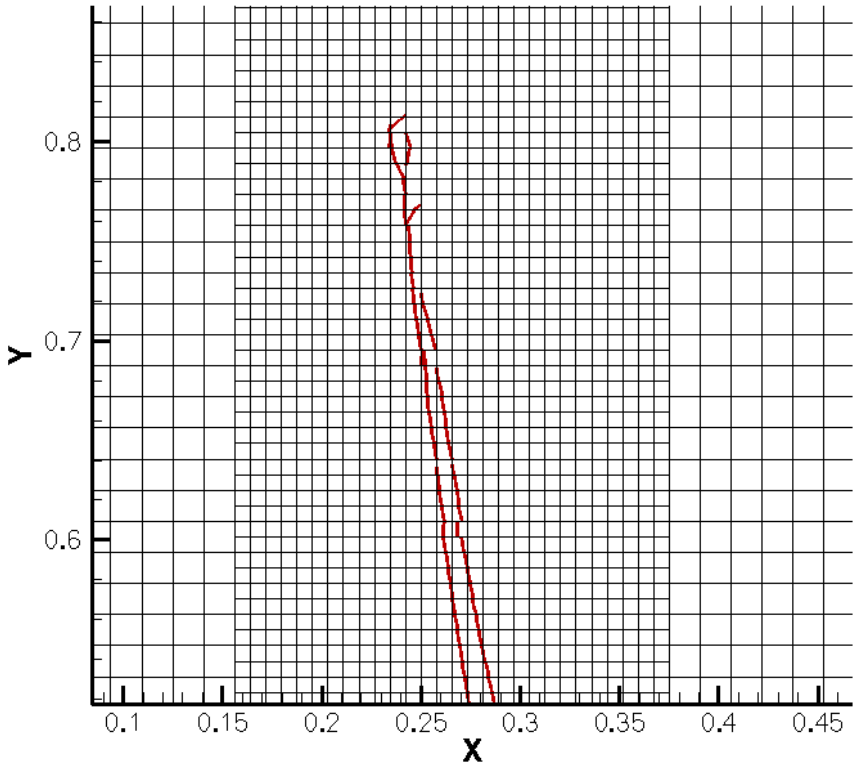

Figure 5.26: Droplet flow at maximum deformation, using standard MOF, with 2 levels of AMR. The view at the sharp tip is enlarged. Note the irregularity in the thin structure near the tip.

- The vorticity error E_{vort} is calculated as the average relative L^2 difference in vorticity over all coarse mesh nodes.

5.6.2 Rayleigh-Taylor Instability

Rayleigh-Taylor instability is a well-known and well-studied phenomenon involving an evolving interface between two incompressible fluids [80, 9, 14, 28, 69], used here for validation of the multi-material fluid algorithm. Suppose a heavy fluid lies on top of a lighter fluid, separated by a perturbed interface. Due to gravitational body force, the heavier material will penetrate the lighter material and cause the interface to roll-up. The characteristics of the penetration and roll-up depend on the non-dimensional Atwood number A of the system, where

$$A = \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2}. \tag{5.35}$$

Table 5.9: Symmetric difference error and runtime for the droplet flow test case at full reversal. Starting mesh resolution is 32^2.

	AMR Level	Error	Runtime (sec)
Filament MOF	0	2.456×10^{-3}	359.373
	1	6.282×10^{-4}	1410.156
	2	2.362×10^{-4}	4250.341
Standard MOF	0	7.427×10^{-3}	323.745
	1	7.087×10^{-3}	1216.777
	2	1.418×10^{-3}	5063.813

Table 5.10: Symmetric difference error and runtime for the S-shape at full reversal. Starting mesh resolution is 32^2.

	AMR Level	Error	Runtime (sec)
Filament MOF	0	2.110×10^{-2}	223.922
	1	1.470×10^{-3}	1878.856
	2	4.108×10^{-4}	7139.634
Standard MOF	0	6.376×10^{-2}	186.030
	1	2.244×10^{-2}	1680.118
	2	2.573×10^{-3}	6883.039

Table 5.11: Relative vorticity errors and convergence rates at two times for horizontal shear layer, solved with the incompressible finite element method. First order convergence is seen at early times. Accumulation of errors causes degradation of accuracy at later times.

Coarse/Fine Mesh Δx	E_{vort} $t = 0.819$	$\tau(E_{vort})$ $t = 0.819$	E_{vort} $t = 1.806$	$\tau(E_{vort})$ $t = 1.806$
$\frac{1}{64}/\frac{1}{128}$	3.586e-02	—	2.373e-2	—
$\frac{1}{128}/\frac{1}{256}$	1.720e-02	1.06	1.367e-2	0.796

Here, ρ_1 is the density of the heavier material on the top of the domain, and ρ_2 is the density of the lighter material on the bottom of the domain. Thus, the Atwood number has the range $A \in [0, 1)$, where the lower bound is achieved when $\rho_1 = \rho_2$, and the upper bound is approached in the limit as $\frac{\rho_2}{\rho_1} \to 0$. For this work, the materials are assumed to be be inviscid, and surface tension is neglected.

As in the work by Galera, et. al. [27], the domain is defined to be $\Omega = [0, 1/6] \times [0, 1]$. The initial perturbed interface, with amplitude a_0, is defined as

$$\Gamma_0(x) = \frac{1}{2} + a_0 \cos(6\pi x). \tag{5.36}$$

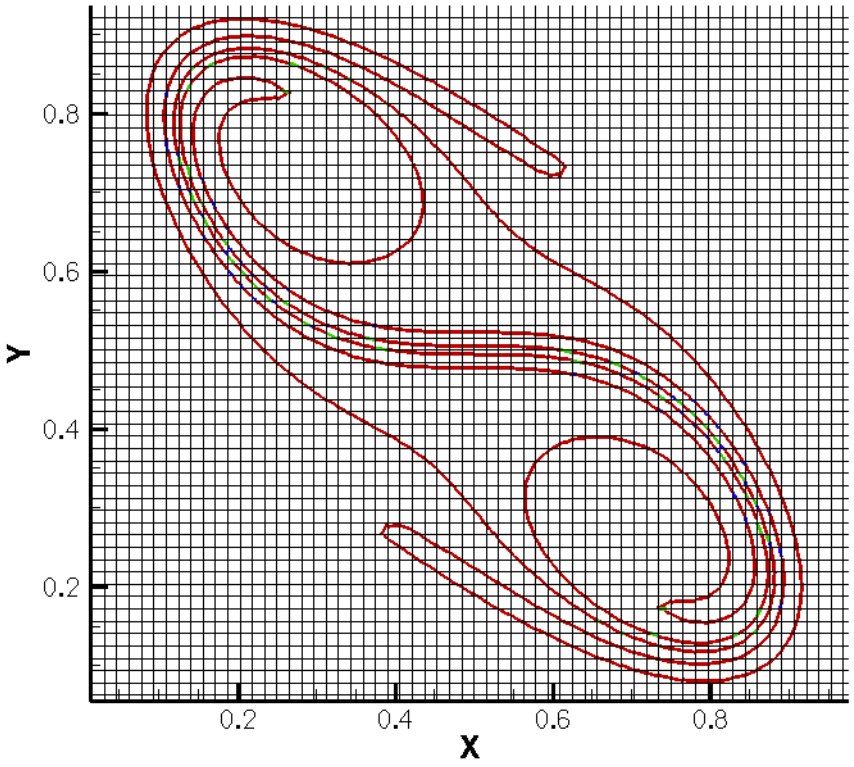

Figure 5.27: The S-shape flow at maximum deformation with starting mesh resolution 32^2 with 1 level of AMR, using filament capturing. Blue lines indicate the introduction of a twin Material $1'$ to capture a filament of Material 2. A green line indicates the introduction of a twin Material $2'$ to capture a filament of Material 1. Note that filaments of both material types are formed.

The densities are taken to be $\rho_1 = 2$ and $\rho_2 = 1$, with the force due to gravity $\boldsymbol{g} = -0.1\ \widehat{e}_y$. This gives an Atwood number $A = 1/3$. Each fluid is assumed to be governed by the ideal gas equation, with $\gamma_1 = \gamma_2 = 1.4$. The perturbation amplitude for (5.36-5.37) is taken to be $a_0 = 1/100$, and $\sigma = 1/4$. An initial, perturbed velocity field is applied inside the domain,

$$\boldsymbol{u}_0 = \begin{pmatrix} 0 \\ a_0 \cos(6\pi x) e^{-\frac{(y-0.5)^2}{\sigma^2}} \end{pmatrix} \tag{5.37}$$

with $\boldsymbol{u}_0 \cdot \widehat{n} = 0$ on the boundary. Initial pressure is taken to be the force per unit area of "column of fluid" above a given point,

$$p_0(x,y) = \begin{cases} 1 + \rho_1 ||\boldsymbol{g}|| (1-y), & y \geq \Gamma_0(x) \\ 1 + \rho_1 ||\boldsymbol{g}|| (1-y) + \rho_2 ||\boldsymbol{g}|| (\Gamma_0(x) - y), & y < \Gamma_0(x) \end{cases}. \tag{5.38}$$

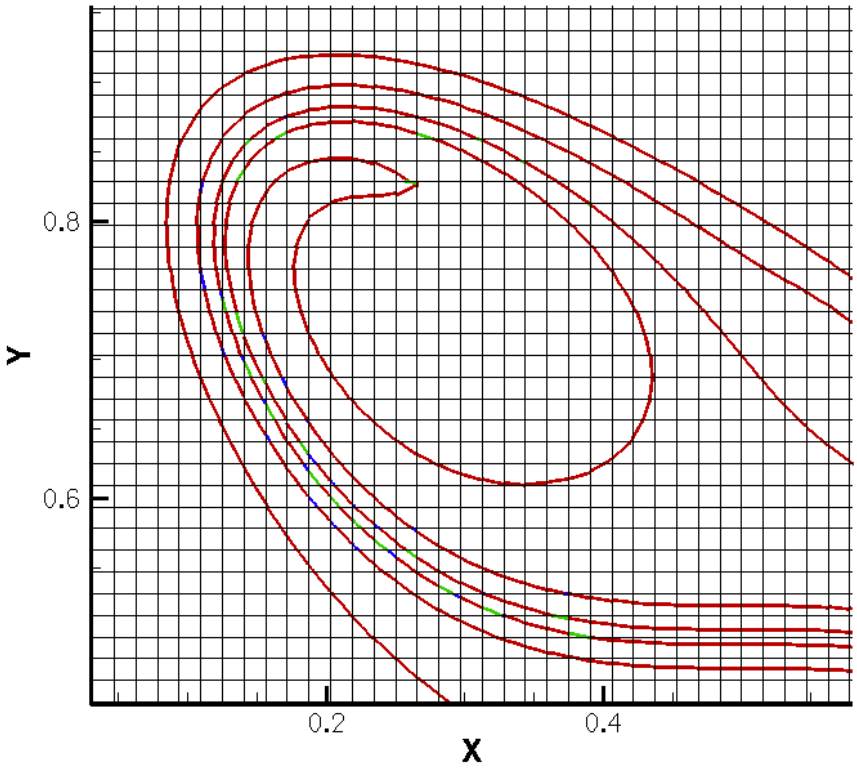

Figure 5.28: The S-shape interface from Fig. 5.27, zoomed in near the vortex center to show captured filaments of each material type. Green lines denote a filament of the material in the initial circle. Blue lines denote a filament of the background material.

Because the materials are taken to be strictly incompressible, the asymptotically-preserving pressure equation simplifies to Poisson's equation with discontinuous coefficients, depending on material density. We wish to prescribe the there is no flow in or out of the domain, i.e. $\boldsymbol{u} \cdot \widehat{n}|_{\partial\Omega} = 0$ for outward unit normal vector $\widehat{n}$. So, the boundary condition for pressure is obtained as

$$\frac{\partial p}{\partial n}|_{\partial\Omega} = \rho \boldsymbol{g} \cdot \widehat{n}. \tag{5.39}$$

The Rayleigh-Taylor Instability test is, by its nature, unstable without viscosity. Small perturbations can lead to large variation in finite time, so nodes are laid out in a logically rectangular fashion and "randomly" triangulated. This limits the formation of secondary Rayleigh-Taylor structures that can form due to interfacial perturbations.

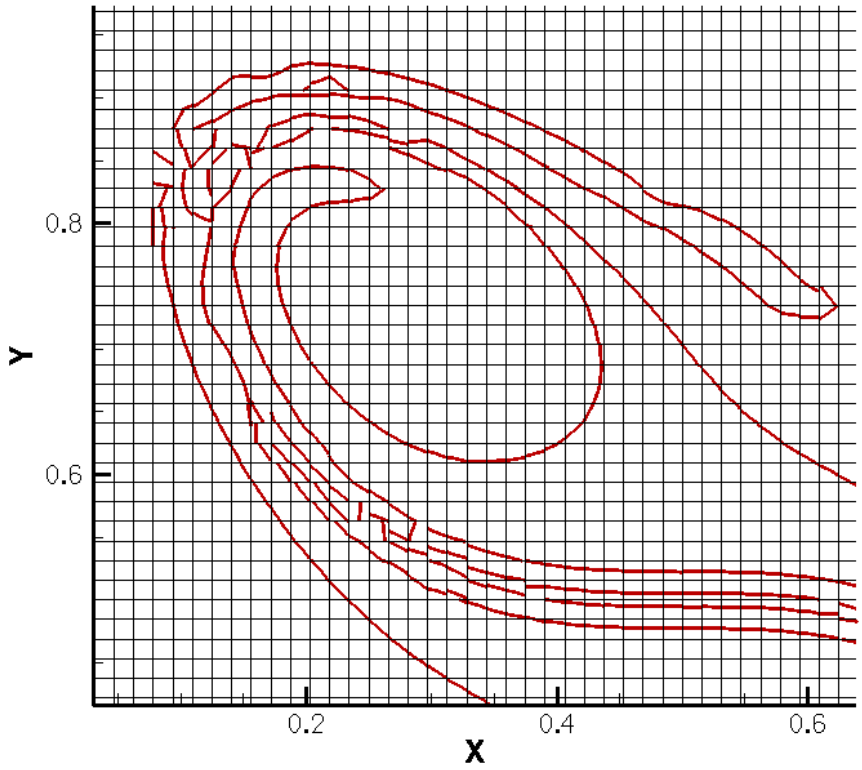

Figure 5.29: The S-shape interface without filament capturing, zoomed in near the vortex center to show break-up due to unresolved thin structures.

Results for the Finite Element method are shown at times $t = 5, 6$, and 7 s (see Fig. 5.32) for Atwood number $A = 1/3$. The domain is tiled with 1600 triangular elements, with nodes distributed uniformly, $\Delta x = 1/96$ and $\Delta y = 1/50$. Results for the two material Finite Element formulation agrees well with the ALE formulation at similar resolution presented in [27], in both spike penetration depth and roll-up width/height. The agreement of this validation test with previously published results indicates that the fluid algorithm is solving Euler's equations correctly.

5.6.2.1 Incompressible CISL Advection. Incompressible flow can be approached from two paradigms with regard to the Cell-Integrated Semi-Lagrangian advection method.

1. Advection of mass and momentum can be performed as in Section 3.2, and density can be masked to the "incompressible density" after advection is complete. Physically, this corre-

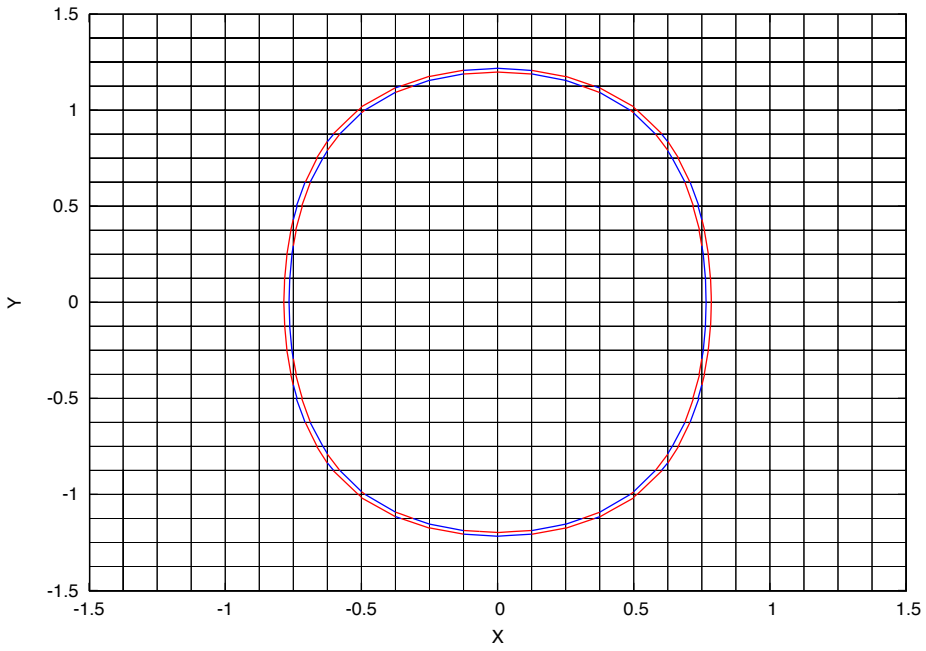

Figure 5.30: Pulsating membrane at time $t = 0.325$.

sponds to allowing material to expand or compress during the Lagrangian mapping phase and updating velocity in a momentum conservative fashion at each step.

2. The density of an incompressible material is updated to be the "incompressible density" if that material exists in a cell, and advective velocity is computed as

$$\boldsymbol{u}_j^a \left(\sum_{m=1}^{M} \rho_j^{I,m} V_j^{m,n+1} \right) = \sum_{i'} \left(\sum_{m=1}^{M} \int_{\Omega_j^D \cap \Omega_{i'}^m} \boldsymbol{u}_{i'}^n \rho_{i'}^{I,m} \, d\Omega \right), \tag{5.40}$$

where $\rho^{I,m}$ is the density of incompressible material m, and $V_j^{m,n+1}$ is the volume of material m in cell j after material is mapped forward in time. Physically, this equates to applying that density is always constant, but velocity is not updated in a momentum conservative fashion.

Both methods were tested, with similar results. As such, Method 1 is used, as it is less intrusive and conserves momentum through the Lagrangian step, which is more consistent with the philosophy of CISL advection.

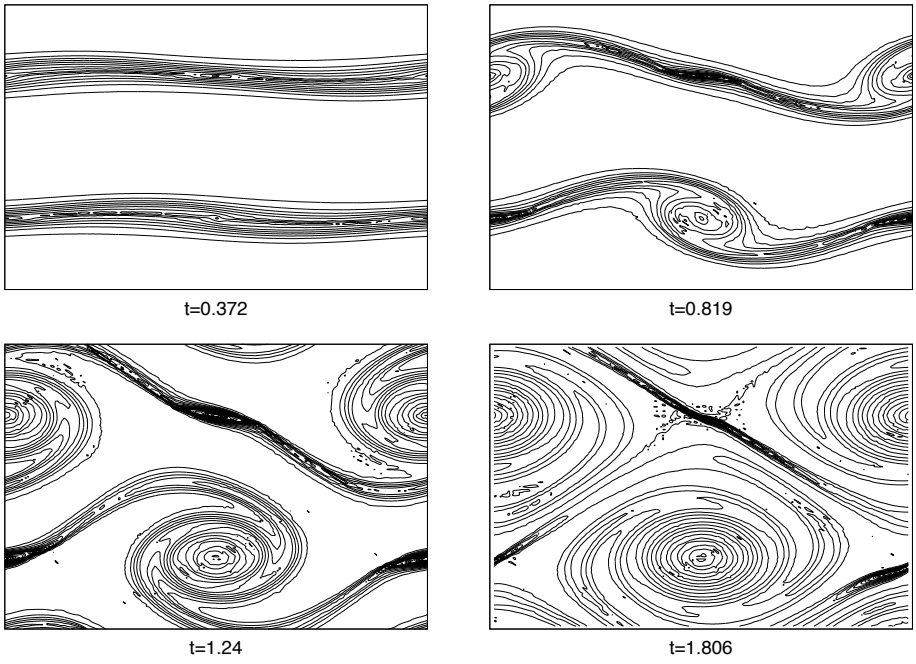

Figure 5.31: Horizontal shear layer vorticity contours at various times.

5.6.3 2D Single-Phase Shock Tube Tests

The Sod and Strong Shock Tube tests from Section 5.1 are cast in 2D to evaluate the unstructured Finite Element and Support Operator Finite Volume fluid algorithms on compressible flow tests. The 1D initial conditions are recast in 2D, with no dependence on y. A uniform mesh on a rectangular domain is used, with $\Delta x = \Delta y = 1/N$. To avoid effects due to deformed elements, the domain for each test is taken to be $\Omega = [0, 1] \times [0, 10\Delta y]$.

A 2D analogue of the acoustic time step is used, where the "maximum propagation" speed in a cell is defined as the maximum cell nodal velocity plus the speed of sound in the cell center. The characteristic length scale of the cell is taken to be the minimum side length. Thus the 2D acoustic time step in each cell is calculated as

$$\Delta t_{u+c} = \frac{\min(|\Delta x|)}{\max(|\boldsymbol{u}_{node}|) + c(\rho, e_{int})}. \tag{5.41}$$

The advective time step is calculated as the time to cell collapse multiplied by some conservative shrink factor (see Sec. 3.2.2). The time step in a cell is the minimum of the acoustic and advective

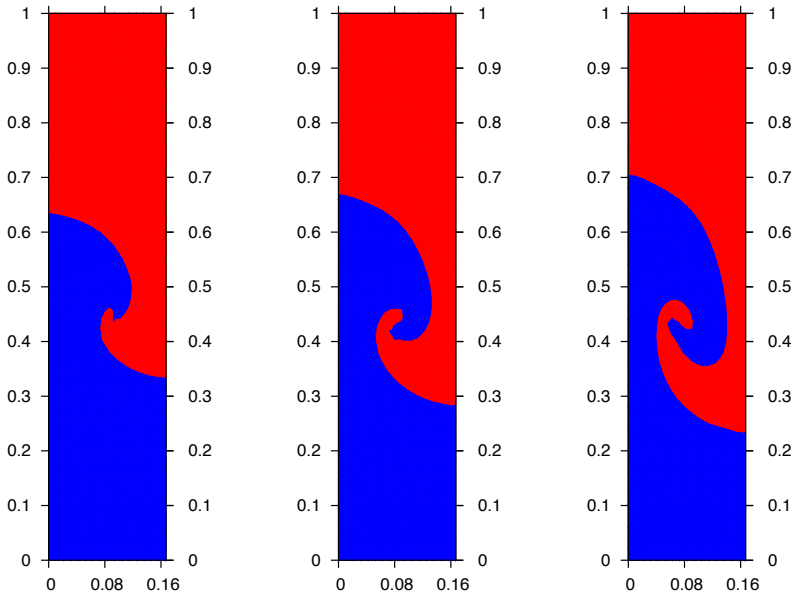

Figure 5.32: Rayleigh-Taylor Instability material configuration with 1600 triangular elements at time $t = 5, 6$, and 7s for Atwood number $A = 1/3$.

time step. The global time step is the minimum time step over all cells in the mesh. Finally, the global time step is allowed to increase by no more than 10% at each iteration.

Sod Shock Tube

$$(\rho(x,y,0), \boldsymbol{u}(x,y,0), p(x,y,0)) = \begin{cases} (1,0,1), & x \leq 0.5 \\ (0.125,0,0.1), & x > 0.5 \end{cases} \tag{5.42}$$

Error and convergence rates for Sod's shock tube problem at time $t = 0.15$ are shown in Table 5.12. The L^1 average nodal errors for pressure are reported, where

$$E_N = \frac{1}{N} \sum_{i=1}^{N} |p_i - p_{exact}|.$$

Convergence rates are approximately $\tau_N \approx 3/4$, similar to those in 1D (Table 5.1). As with the 1D finite volume method, errors increase when a larger CFL number is used in calculating the time step.

$$\tau_N = \log(E_N/E_{N/2})/\log(2).$$

Figure 5.33 shows a comparison of the final pressure for the 2D Finite Element method and Support Operator Finite Volume method for Sod's problem. The Finite Element method captures the shock well, with minimal overshoot and oscillation, and pressure is constant across the contact discontinuity. The Support-Operator Finite Volume method, while designed for problems with discontinuous coefficients, exhibits severe oscillation at the shock. This illustrates the superior performance of the FEM compared to the SO-FVM in the context of the asymptotically-preserving pressure method on unstructured triangular meshes. While Finite Volume methods are traditionally used for problems with discontinuous solutions, the Finite Element method is seen in this work to perform more robustly in the presence of discontinuous coefficients, while also being far cheaper computationally.

Table 5.12: Average nodal L^1 error E_N and convergence rate τ_N for pressure for the Finite Element Method applied to the 2D Sod Shock Tube problem at $t = 0.15$ (5.4).

N	Sod E_N CFL= 0.5	Sod τ_N CFL= 0.5	Sod E_N CFL= 1	Sod τ_N CFL= 1
100	9.951e-3	—	1.325e-2	—
200	6.027e-3	0.723	7.933e-3	0.740
400	3.606e-3	0.741	4.892e-3	0.698

5.7 2D Oscillating Water Column

The 2D Oscillating Water Column test is initialized in the same way as the 1D test (Section 5.2), with the 2D test set up to be uniform in the y-direction. The domain is taken to be $\Omega = [-1, 1] \times [0, 1/4]$. Nodes distributed uniformly with resolution $\Delta x = \Delta y = 1/40$ and "randomly" triangulated. Reflecting wall boundary conditions are used on all edges of the domain. Figure 5.34 shows a "profile" of pressure at the left and right ends of the tube with respect to time. For this test, water is treated to be incompressible, and the sound speed is taken to be infinite in any cell more than half-occupied by water.

$$\begin{cases} c_i = \infty, & \text{if } F_i^{H_2O} >= 1/2 \\ c_i = c(\rho_i^{gas}, e_i^{gas}), & \text{otherwise} \end{cases}$$

It is noted that this test is sensitive to perturbations in the direction orthogonal to the flow, which is exacerbated by the use of triangular elements. As the liquid region enters new cells

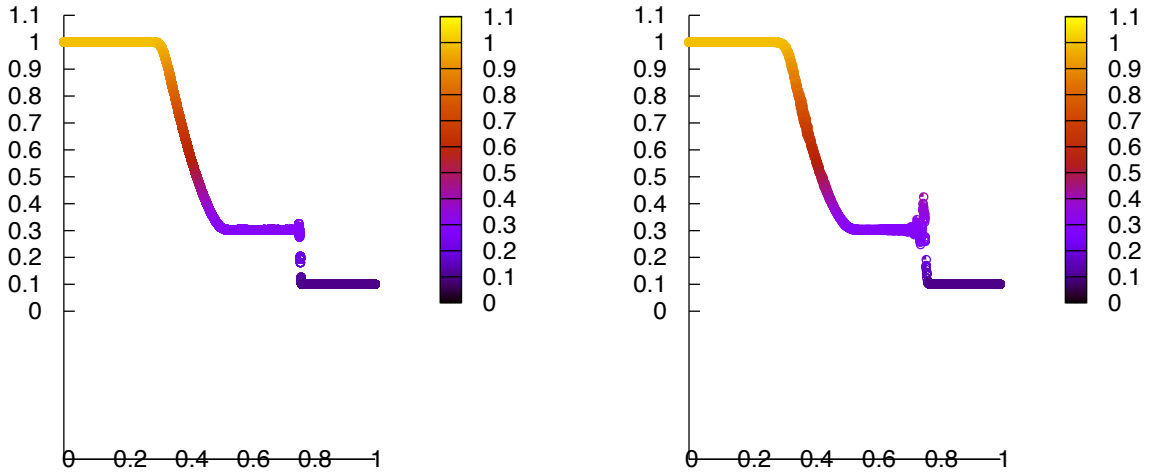

Figure 5.33: Pressure for the 2D Sod Shock Tube at time $t = 0.15\ s$, using two different methods. Left: All-speed Finite Element pressure projection. Note that the shock is well captured, and pressure is constant across the contact discontinuity. Right: All-speed Support-Operator Finite Volume method. Note the spurious oscillation at the shock front and non-uniformity in the direction orthogonal to the flow, near the contact discontinuity.

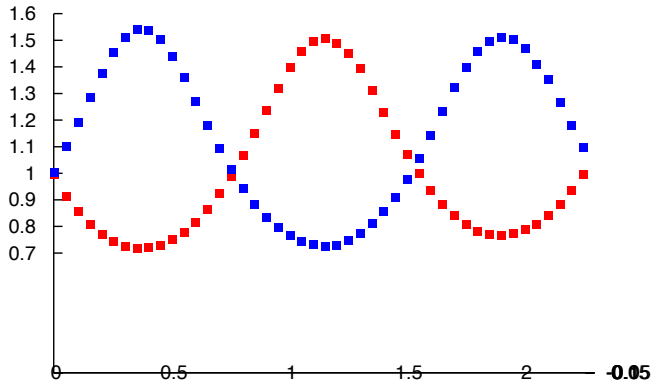

Figure 5.34: Pressure at the right (blue) and left (red) ends of the closed tube. A horizontal "profile" view of pressure is taken, showing uniformity in the y-direction. Note that for early times, period and amplitude of the pressure oscillation agrees well with results in 1D.

and occupies the cell centroid, this will cause a sudden jump in pressure. However, cells will not be covered by the interface in a uniform manner (unlike rectangular meshes). Whether pressure

is determined by the "sharp interface" paradigm or by mass weighting, pressure in neighboring elements will not transition uniformly as the liquid region occupies new cells. This will cause a pressure gradient with a component in the direction orthogonal to the flow that leads to the breakdown of the simulation for large times. However, for short times, it is observed that the oscillation of pressure at the ends of the tube exhibits similar amplitude and period to the results shown in 1D, with good uniformity in the y-direction, for early times.

5.8 Spherical Explosion Shock

An underwater blast is a two-material moving interface test, simulated by initializing a stationary inviscid gas bubble surrounded by inviscid liquid taken to follow the Tait equation of state, as in Wardlaw [83]. The problem is cast in 1D spherical and 2D cylindrical coordinates. A bubble of JWL gas with radius 16 cm. is initialized at the center of the spherical mesh. Initial conditions for density, pressure, and energy are taken to be the same as the JWL shock tube. The acoustic time step (5.1) with $CFL = 1/2$ was used.

$$\begin{cases} \text{JWL Gas: } \rho = 1.63, e_{int} = 4.2814E+10, u = 0, p = 7.81E+10, & r \leq 16 \\ \text{Water: } \rho = 1.0, e_{int} = \text{N/A}, u = 0, p = 1.00E+6, & r > 16 \\ r \in [0, 256] \end{cases} \tag{5.43}$$

The pressure difference at the interface results in a right-moving shock, left-moving rarefaction, and right-moving contact discontinuity. Pressure drops in the bubble, until a critical stage is reached, when the difference in pressure between the interface and the center of the bubble causes a left-going shock to form. This reflects off of the internal wall and reshocks the interface. Euler's equations are cast in spherical coordinates, as in [42].

5.8.1 Spherical Lagrangian Characteristic Tracing

The ODE that governs characteristic backtracing is modified for spherical coordinates. Let α denote the foot of a characteristic, u_f denote a face velocity, and r_f denote radius at the face of a cell. Then, the departure region for a cell is defined by the solution to (5.44) at time $t = \Delta t$.

$$\begin{cases} \dfrac{d\alpha(t)}{dt} = -\dfrac{u_f r_f^2}{\alpha^2} \\ \alpha(0) = r_f \end{cases} \tag{5.44}$$

A linear mapping is applied to map points in the departure region Ω_D to points in the target cell, as in Section 1. The linear mapping coefficient $\mathcal{C}_r$ is computed as

$$\mathcal{C}_r = \frac{\Delta r}{\alpha_{i+1/2} - \alpha_{i-1/2}}. \tag{5.45}$$

$$f(r) = \mathcal{C}_r \left(r - \alpha_{i-1/2}\right) + r_{i-1/2} \tag{5.46}$$

A discretely divergence-free face velocity in spherical coordinates does imply that $\mathcal{C}_r = 1$, i.e. that $|\Omega_D| = |\Omega|$, so a constant field is preserved under a divergence-free face velocity in a single material. This does not hold for multiple materials. Because the mapping from the departure region to the target region is linear, contributions from individual cell may undergo expansion or compression, but the errors cancel in computing the cell-averaged value. If an interface cuts the departure region, the individual liquid and gas regions will feel these effects separately, and the errors will not cancel. Hence, a divergence-free velocity will result in slight compression/expansion

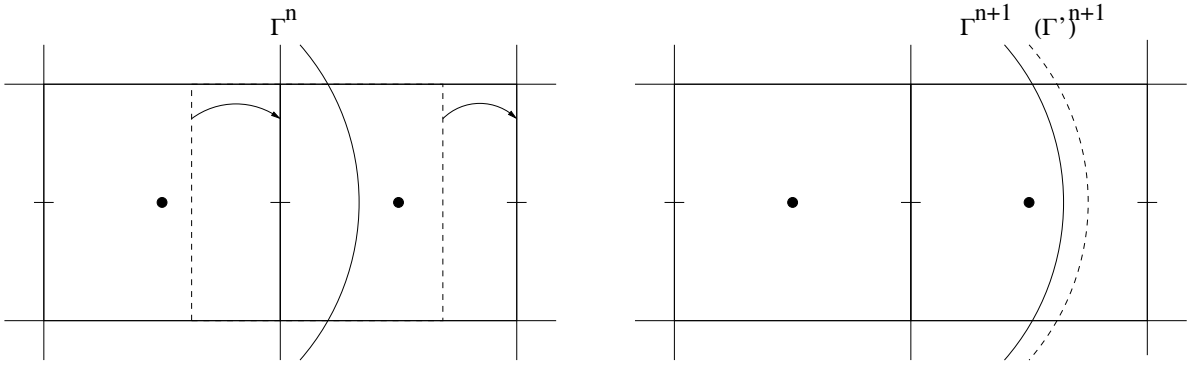

Figure 5.35: Left: Departure region (dotted box) with interface Γ^n is mapped to the target cell under a divergence-free face velocity. Right: Reconstructed interface Γ^{n+1} according to linear mapping (solid curve), and true interface $(\Gamma')^{n+1}$ (dotted curve) mapped with the analytic, divergence-free velocity.

at the interface. See Fig. 5.35. A version of (5.44) can be formulated, with an associated mapping, that has the same departure region and exactly preserves a constant field for a departure region cut by a material interface under a divergence-free face velocity.

$$\frac{d\alpha(r,t)}{dt} = -\frac{Ar + B}{\alpha^2} \tag{5.47}$$

Mapping points from the departure region to the target cell then requires solving for the roots of a depressed cubic polynomial for each mapped point. In 2D, a similar technique in cylindrical coordinates maps a polygonal domain in a departure region to a curved figure in the target region, which introduces errors with the volume and moment computations of the Moment-of-Fluid method. The errors associated with approximating a non-linear velocity field with a linear velocity field, for the purposes of moment calculation, are addressed in the original work on the Moment-of-Fluid method and said to be of third order [20].

Numerical tests for the spherical explosion shock demonstrated very similar behavior in interface location and shock speed between the two methods. As such, the linearity preserving mapping (5.46) is used for problems in spherical coordinates.

5.8.2 Spherical Explosion Shock - 1D Spherical Results

A comparison is made to the benchmark results in Wardlaw [83], with the simulation terminated at time 5.0E-4 seconds. The domain is taken to be of size 256 cm, so that the external boundary does not interfere with the simulation. The internal boundary condition is taken to be a wall, discretized as in (5.12). State variables at the final time are shown in Fig. 5.36, with the interface in red. Resolution is N = 512 cells.

Table 5.13: Results for the Spherical Explosion Shock in 1D Spherical coordinates (Sec. 5.8), with first order, piecewise constant slope reconstruction, two backward sweeps.

Δx	Shock Peak	Bubble Position	Number of Time Steps
1/2	1.6446E+9	41.067	522
1/4	1.7550E+9	41.139	1054
1/8	1.8279E+9	41.181	2123

Table 5.14: Results for the Spherical Explosion Shock in 1D Spherical coordinates (Sec. 5.8), with linear MINMOD density reconstruction in all conserved quantities, and two backward sweeps.

Δx	Shock Peak	Bubble Position	Number of Time Steps
1/2	1.6535e+9	41.086	573
1/4	1.7611e+9	41.150	1106
1/8	1.8328e+9	41.187	2177

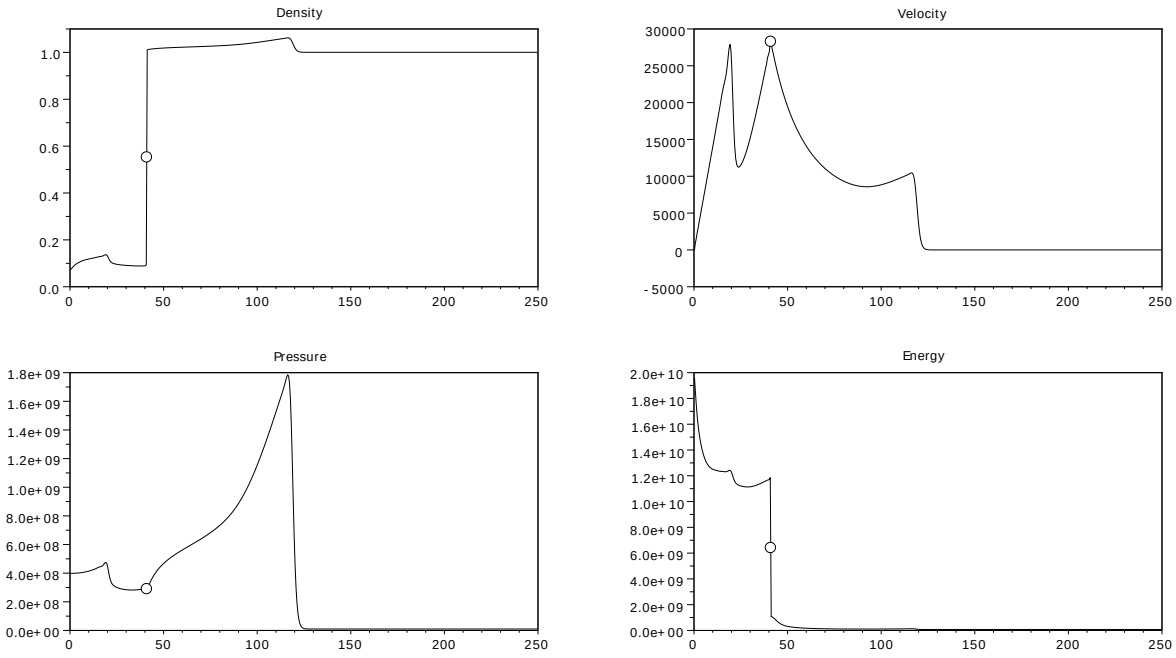

Figure 5.36: Results for the JWL-Tait spherical explosion shock at time $t = 5.0E-4$ sec. The pressure plot shows the left-moving shock after it has reflected off of the origin. The black circle in each image denotes the location of the material interface. Resolution is N = 512 cells. A linear MINMOD reconstruction for each conserved quantity is used.

Convergence of the solution at the shock in the gas and liquid regions is shown in Figure 5.37 and Figure 5.38. Results are shown for density with resolution varying from N=256 to N=4096. Approximate first order convergence is observed in the magnitude and location of the shock in the images. Tables 5.13 - 5.14 additionally show slightly less than first order convergence in the interface location and shock magnitude. This is consistent with expectations of behavior of the method. A comparison is made in Fig. 5.39 - 5.40 between use of the MINMOD and piecewise constant slope reconstructions, for resolution N = 512. As expected, use of the MINMOD slope limiter produces a slightly sharper front at the shock. However, differences are minimal in shock magnitude and bubble location between the use on MINMOD limiting versus constant slope reconstruction for density.

This result is novel in that it uses a conservative, shock-capturing method on a fixed grid to simulate the underwater blast. Unlike Kadioglu [41], the JWL gas is not assumed to be spatially uniform in density or adiabatic. By allowing density to vary in space and time (rather than having

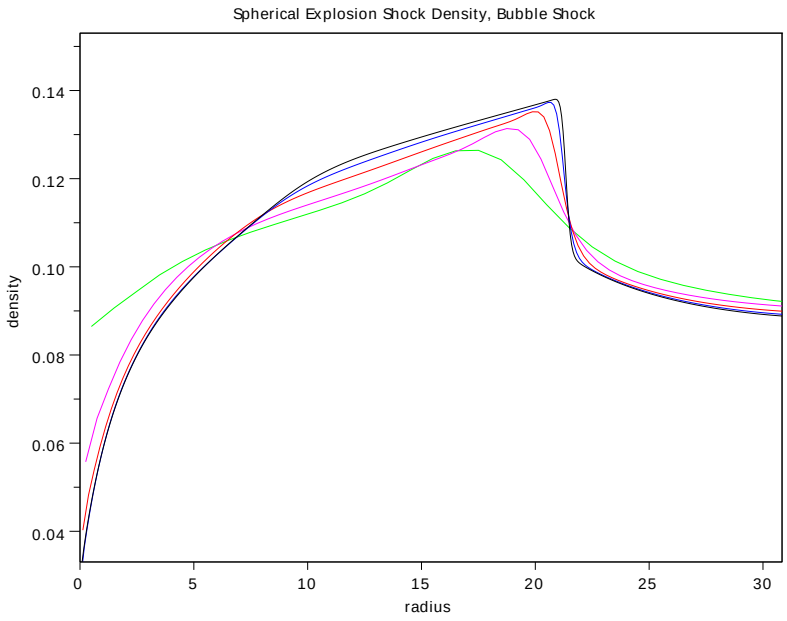

Figure 5.37: Convergence of density with linear MINMOD slope reconstruction at the shock in the JWL bubble region for the Spherical Explosion Shock at time $t = 5.0E-4$ sec. The colors denote the following resolution: N=256 (green), N=512 (magenta), N=1024 (red), N = 2048 (blue), N = 4096 (black).

a spatially uniform gas density that varies with time), it allows for additional physics of the test to be captured. As the bubble expands, density in the gas region drops. After density in the bubble decreases below a threshold, a shock emanates from the material interface, moving from right to left, from the liquid to the gas. The shock reflects off of the origin and reshocks the material interface. The behavior is captured by simulating the gas as compressible and non-uniform. This work uses a cut-cell Moment-of-Fluid method to represent the interface, so the grid is not deformed or fit to the shock or material interface, as in Wardlaw [82]. Agreement in location of the material interface, shock location, and magnitude of pressure at the shock front is observed with the benchmark in Wardlaw. Scaling is performed in the computation of characteristics (5.44) and in integration over control volumes to reduce the severity of round-off errors. Relative mass error is $\mathcal{O}(10^{-13})$ in the JWL gas and $\mathcal{O}(10^{-10})$ in the water. The discrepancy is attributed to round-off error, in that the gas occupies the interior of the domain, where control volumes are much smaller than those toward

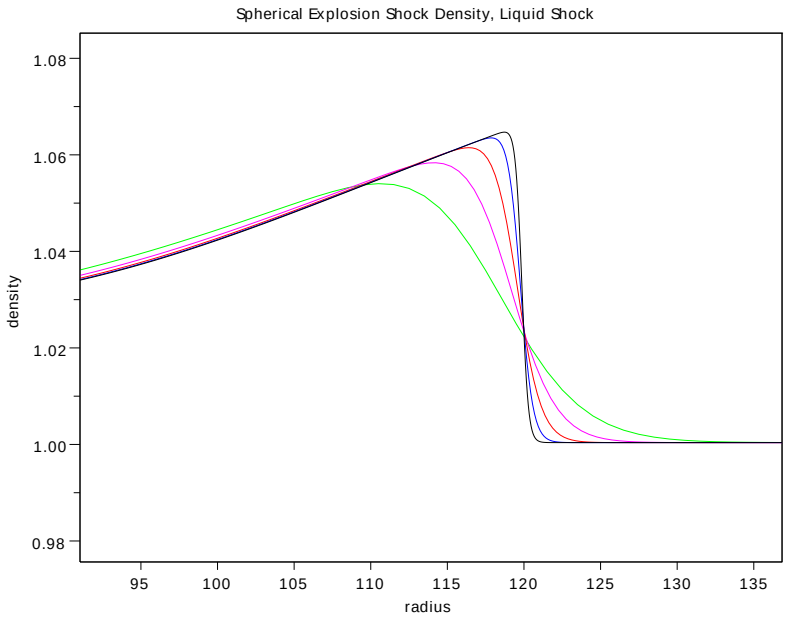

Figure 5.38: Convergence of density with MINMOD linear slope reconstruction at the shock in the Tait liquid region for the Spherical Explosion Shock at time $t = 5.0E-4$ sec. The colors denote the following resolution: N=256 (green), N=512 (magenta), N=1024 (red), N = 2048 (blue), N = 4096 (black).

the edge of the mesh.

Remarks:

- In Figures 5.39 - 5.40, a comparison is made as to using piecewise constant vs. MINMOD linear reconstruction in density in 1D. Very little difference in position and magnitude are observed.
- Approximate first order convergence in shock magnitude and bubble position is observed as the mesh is refined.

5.8.3 Spherical Explosion Shock - 2D Curvilinear Results

The Spherical Explosion Shock test is cast in 2D cylindrical coordinates and simulated using the Finite Element Method. The initial conditions are identical to (5.43), however the domain is reduced to $\Omega = [0, 160]^2$. Using 256^2 nodes, this gives resolution comparable to the coarsest

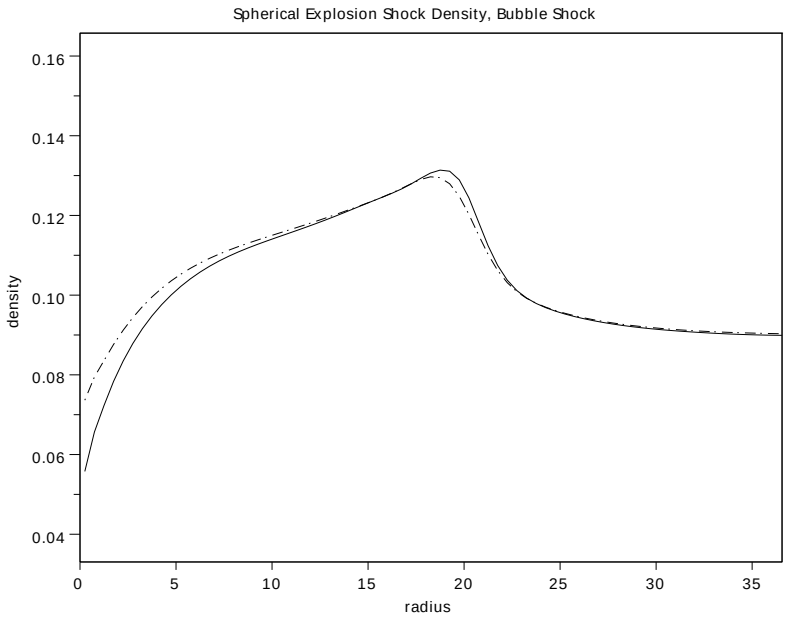

Figure 5.39: Comparison slope reconstructions at the shock in the JWL gas region for the Spherical Explosion Shock at time t = 5.0E-4 sec, with resolution N = 512. The solid black line uses the MINMOD slope reconstruction, the dotted line uses a piecewise constant reconstruction.

resolution in 5.13, and is large enough that the flow does not interact with the boundary before the run is terminated.

Using the 2D Finite Element method, results are not as sharp as the 1D Finite Volume method. This is due to the use of piecewise-linear nodal basis functions, rather than the piecewise-constant cell-centered basis functions used in 1D. However, the solution is noted to be symmetric with respect to angle θ, as seen in Figure 5.41, with some variation at the origin and near the contact discontinuity

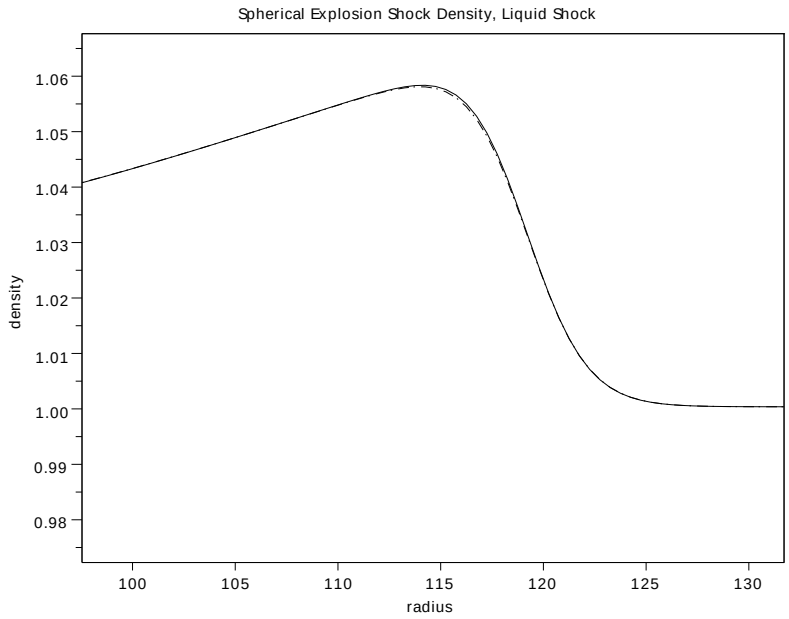

Figure 5.40: Comparison slope reconstructions at the shock in the Tait liquid region for the Spherical Explosion Shock at time t = 5.0E-4 sec, with resolution N = 512. The solid black line uses the MINMOD slope reconstruction, the dotted line uses a piecewise constant reconstruction.

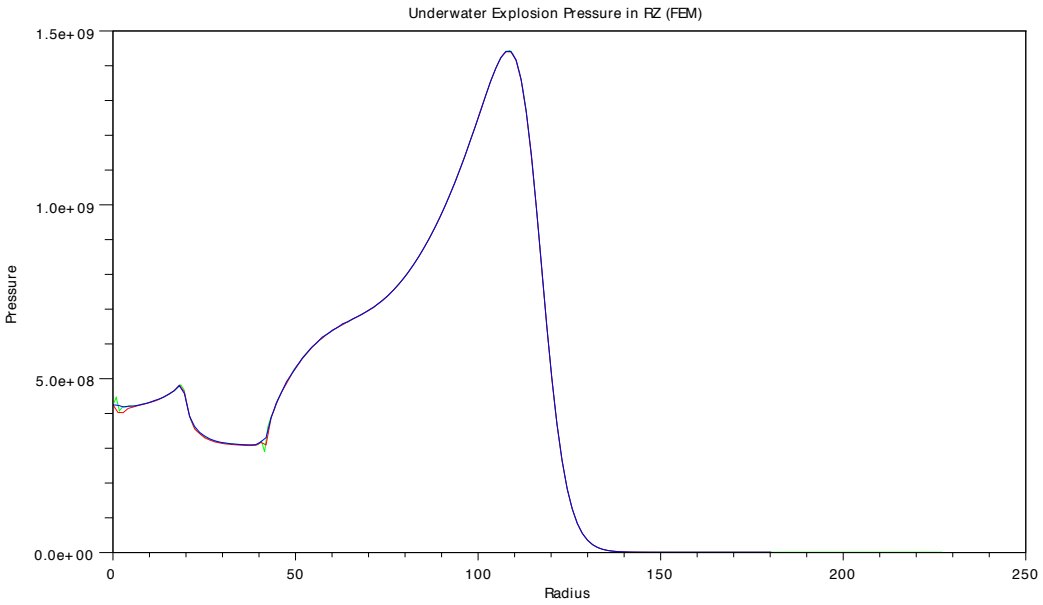

Figure 5.41: Comparison of pressure at $t = 0.5\,ms$ along the line $y/x = 1/2$ (green), 1 (blue), and 2 (red) using the Finite Element method. Symmetry in the solution is observed, along with agreement in the magnitude of the secondary shock in the bubble on the domain $[0, 160]^2$. The shock peak in the liquid is diffused from the 1D results.

CHAPTER 6

CONCLUSIONS

In summary, a first-order, asymptotically-preserving method for simulating multiphase flow in a mass, momentum, and energy conservative fashion is presented. This unified technique is applicable to compressible flow in the presence of strong shocks, while still recovering the dynamics of incompressible flow in the limit of infinite sound speed, without the need for artificial mechanisms to distinguish between regimes. The method is shown to be convergent for a variety of tests while taking longer time steps than those prescribed by the standard Courant-Friedrich-Levy (CFL) condition, due to the use of a conservative Cell-Integrated Semi-Lagrangian advection scheme. In the presence of a material interface with high impedance-mismatch, the method is seen to perform robustly, while limiting the formation of spurious oscillations.

New techniques for capturing deforming material interfaces in 2D are presented. A novel method for capturing filaments using the existing functionality of the Moment-of-Fluid interface reconstruction method is presented. This method is unique in that it uses only local information to detect and capture the formation of a filament, while remaining asymptotically constant in cost as the width of the filament approaches zero. Additionally, a new method for choosing an optimal initial interface for the Moment-of-Fluid optimization procedure is presented for two-dimensional triangular mesh elements.

The one-dimensional fluid algorithm, which generalizes well to multi-dimensional rectangular meshes, is extended to unstructured triangular meshes. Two discretizations of the asymptotically-preserving pressure evolution equation are presented: a Support-Operator Finite Volume method and a Finite Element method. While both methods are second-order accurate for smooth-solutions, the Finite Element method is seen to perform more robustly in the presence of contact discontinuities and shock fronts. This method is applied to a variety of single- and multi-material problems in 2D, and while it is not as robust as a 2D Finite Volume method on a rectangular mesh, it is seen to capture the physics of both compressible and incompressible flow.

BIBLIOGRAPHY

[1] M. Aanjaneya, S. Patkar, and R. Fedkiw. A mass tracking formulation for bubbles in incompressible flow. *J. Comput. Phys.*, 2013. 1, 2, 3

[2] H.T. Ahn and M. Shashkov. Multi-material interface reconstruction on generalized polyhedral meshes. *J. Comput. Phys.*, 226(2):2096–2132, 2007. 19, 24

[3] H.T. Ahn and M. Shashkov. Adaptive moment-of-fluid method. *J. Comput. Phys.*, 228(8):2792–2821, 2009. 4, 19, 109, 111

[4] John C. Anderson, Christoph Garth, Mark A. Duchaineau, and Kenneth I. Joy. Smooth, volume-accurate material interface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 16(5):802–814, 2010. 4

[5] John D. Anderson and John Anderson. *Modern Compressible Flow: With Historical Perspective*. McGraw-Hill Science/Engineering/Math, 3 edition, July 2002. 7, 10, 16, 17

[6] M. Arienti and M. Soteriou. Time-resolved proper orthogonal decomposition of liquid jet dynamics. *Phys. Fluids*, 21, 2009. 1

[7] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, February 2000. 16

[8] J.R. Bates and A. McDonald. Multiply-upstream, semi-lagrangian advective schemes: Analysis and application to a multi-level primitive equation model. *Mo. Wea. Rev.*, 110:1831–1842, Dec. 1982. 35

[9] J. B. Bell and D. L. Marcus. A second-order projection method for variable-density flows. *J. Comput. Phys.*, 101:334–348, 1992. 117

[10] D. J. Benson. Volume of fluid interface reconstruction methods for multi-material problems. *App. Mech. Rev.*, 55:151–165, 2006. 18

[11] W. Bo, X. Liu, J. Glimm, and X. Li. A robust front tracking method: Verification and application to simulation of the primary breakup of a liquid jet. *SIAM J. Sci. Comput.*, 33:1505–1524, 2011. 1, 2, 3

[12] J. Breil and P.H. Maire. A cell-centered diffusion scheme on two-dimensional unstructured meshes. *J. Comput. Phys.*, 224:785–823, 2007. 67, 68, 73, 79, 97, 99

[13] Edip Can and Andrea Prosperetti. A level set method for vapor bubble dynamics. *Journal of Computational Physics*, 231(4):1533 – 1552, 2012. 18

[14] G. Cerne, S. Petelin, and I. Tiselj. Coupling of the interface tracking and the two-fluid models for the simulation of incompressible two-phase flow. *J. Comput. Phys.*, 171:776–804, 2001. 4, 18, 117

[15] C-H Chang, X. Deng, and T. G. Theofanous. Direct numerical simulation of interfacial instabilities: A consistent, conservative, all-speed, sharp-interface method. *J. Comput. Phys.*, 242:946–990, 2013. 1, 2, 3

[16] Y.C. Chang, T.Y. Hou, B. Merriman, and S. Osher. Eulerian capturing methods based on a level set formulation for incompressible fluid interfaces. *J. Comput. Phys.*, 124:449–464, 1996. 18

[17] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comput.*, 22:745–762, October 1968. 3

[18] I. G. Currie. *Fundamental Mechanics of Fluids*. CRC Press, December 2002. 16

[19] P. Degond, H. Liu, D. Savelief, and M.-H. Vignal. Numerical approximation of the euler-poisson-boltzmann model in the quasineutral limit. *J. Sci. Comput.*, 51:59–86, 2012. 1

[20] Vadim Dyadechko and Mikhail Shashkov. Reconstruction of multi-material interfaces from moment data. *J. Comput. Phys.*, 227:5361–5384, 2008. 5, 18, 31, 32, 129

[21] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comp. Phys.*, 183(1):83–116, 2002. 4

[22] Charbel Farhat, Jean-Frédéric Gerbeau, and Arthur Rallu. FIVER: A Finite Volume Method Based on Exact Two-Phase Riemann Problems and Sparse Grids for Multi-Material Flows with Large Density Jumps. *J. Comput. Phys.*, 231:6360–6379, 2012. 1, 2, 3, 83

[23] Ronald P. Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.*, 152:457–492, 1999. 1, 2, 60, 62, 95

[24] C. Fochesato, R. Loubere, R. Motte, and J. Ovadia. Adaptive subdivision piecewise linear interface calculation (ASPLIC) for 2d multi-material hydrodynamic simulation codes. *Int. J. Num. Meth. Fluids*, 2013. 4

[25] C. Fochesato, R. Loubere, R. Motte, and J. Ovadia. A vof extension to live better with filaments and fragments in multimaterial simulations. In *MULTIMAT '13*, San Francisco, CA, USA, September 2-6 2013. 4

[26] M. M. Francois, R. B. Lowrie, and E. D. Dendy. A material interface transition algorithm for multiphase flow. In *Proceedings of the FEDSM 08, ASME Fluids Engineering Division*, Jacksonville, FL, USA, August 10-14 2008. 4

[27] S. Galera, J. Breil, and P.H. Maire. A 2d unstructured multi-material cell-centered arbitrary lagrangianeulerian (ccale) scheme using mof interface reconstruction. *Comp. Fluids*, 46:237–244, 2011. 1, 2, 3, 6, 118, 121

[28] S. Galera, P.H. Maire, and J. Breil. A two-dimensional unstructured cell-centered multi-material ale scheme using vof interface reconstruction. *J. Comput. Phys.*, 229:5755–5787, 2010. 6, 117

[29] R. Garimella, V. Dyadechko, B. Swartz, and M. Shashkov. Interface reconstruction in multi-fluid, multiphase flow simulations. In *Proceedings of the 14th International Meshing Roundtable*, San Diego, CA, USA, September 11-14 2005. 4

[30] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. 82

[31] J. Gretarsson and R. Fedkiw. Fully conservative leak-proof treatment of thin solid structures immersed in compressible fluids. *J. Comput. Phys.*, 2013. 88

[32] F. Harlow and A. Amsden. Numerical calculation of almost incompressible flow. *J. Comput. Phys.*, 3(1):80–93, 1968. 1, 2, 3

[33] Ryan N. Hill and Mikhail Shashkov. The symmetric moment-of-fluid interface reconstruction algorithm. *J. Comput. Phys.*, 249:180–184, 2013. 21

[34] T. Y. Hou and P. G. Le Floch. Why nonconservative schemes converge to wrong solutions: Error analysis. *Math. Comp.*, 62(206):497–530, April 1994. 88

[35] X.Y. Hu, N.A. Adams, and G. Iaccarino. On the hllc riemann solver for interface interaction in compressible multi-fluid flow. *J. Comput. Phys.*, 228:6572–6589, 2009. 35

[36] M.J. Ivings, D.M. Causon, and E.F. Toro. On riemann solvers for compressible liquids. *Int. J. Num. Meth. Fluids*, 28:395–418, 1998. 35

[37] Matthew Jemison, Eva Loch, Mark Sussman, Mikhail Shashkov, Marco Arienti, Mitsuhiro Ohta, and Yaohong Wang. A coupled level set-moment of fluid method for incompressible two-phase flows. *J. Sci. Comput.*, 54(2-3):454–491, February 2013. 44

[38] Matthew Jemison, Mark Sussman, and Marco Arienti. Compressible, Multiphase Semi-Implicit Method with Moment of Fluid Interface Representation. *preprint*, July 2013. 64, 83

[39] J.Gretarsson and R.Fedkiw. Fully conservative leak-proof treatment of thin solid structures immersed in compressible fluids. *J. Comp. Pyhs.*, 245:160–204, 2013. 4

[40] E. Johnsen and T. Colonius. Shock-induced collapse of a gas bubble in shockwave lithotripsy. *J. Acoust. Soc. Am.*, 124:2011–2020, 2008. 1

[41] S. Kadioglu, M. Sussman, S. Osher, J.P. Wright, and M. Kang. A second order primitive preconditioner for solving all speed multi-phase flows. *Journal of Computational Physics*, 209(2):477–503, 2005. 1, 2, 3, 91, 130

[42] Samet Kadioglu. *All Speed Multi-Phase Flow Solvers.* dissertation, Florida State University, 2005. 127

[43] B. Koren, M. R. Lewis, E. H. van Brummelen, and B. van Leer. Riemann-problem and level-set approaches for homentropic two-fluid flow computations. *J. Comput. Phys.*, 181:654–674, 2002. 1, 2, 3, 18, 91

[44] Nipun Kwatra, Jonathan Su, Jón T. Grétarsson, and Ronald Fedkiw. A method for avoiding the acoustic time step restriction in compressible flow. *J. Comput. Phys.*, 228:4146–4161, June 2009. 3, 6, 61, 62, 63, 64, 66

[45] P. Lauritzen. A stability analysis of finite-volume advection schemes permitting long time steps. *Mo. Wea. Rev.*, 2006. 35, 45

[46] V. LeChenadec and H. Pitsch. A 3d unsplit forward/backward volume-of-fluid approach and coupling to the level set method. *J. Comput. Phys.*, 233, 2013. 18, 44, 46

[47] Lentine, J.Gretarsson, and R.Fedkiw. An unconditionally stable fully conservative semi-lagrangian method. *J. Comp. Pyhs.*, 230(8):2857–2879, 2011. 4

[48] S. Li and F.Xiao. Cip/multi-moment finite volume method for euler equations: A semi-lagrangian characteristic formulation. *J. Comp. Phys.*, 222:849–871, 2007. 86, 89

[49] T. G. Liu, B. C. Khoo, and C. W. Wang. The ghost fluid method for compressible gas-water simulation. *J. Comput. Phys.*, 204(1):193–221, March 2005. 1

[50] B. Machenhauer and M. Olk. The implementation of the semi-lmplicit scheme in cell-lntegrated semi-lagrangian models. *Atmosphere-Ocean*, 35:103–126, 1995. 35

[51] R. C. Martineau and R. A. Berry. The pressure-corrected ice finite element method for compressible flows on unstructured meshes. *J. Comput. Phys.*, 198:659–685, 2004. 3

[52] A. McDonald and J.R. Bates. Improving the estimate of the departure point position in a two-time level semi-lagrangian and semi-implicit scheme. *Am. Met. Soc.*, 115:737–739, Sept. 1986. 35

[53] R. Nair, J. Cote, and A. Staniforth. Monotonic cascade interpolation of semi-lagrangian advection. *J. Comput. Phys.*, 125:197–212, 1999. 35, 44, 45

[54] R. Nair, J. Scroggs, and F. Semazzi. A forward-trajectory global semi-lagrangian transport scheme. *J. Comput. Phys.*, 190:275–294, 2003. 35, 45, 46

[55] R. D. Nair and B. Machenhauer. The mass-conservative cell-integrated semi-lagrangian advection scheme on the sphere. *Am. Met. Soc.*, 130, March 2002. 35

[56] J.-C. Nave, R. R. Rosales, and B. Seibold. A gradient-augmented level set method with an optimally local, coherent advection scheme. *J. Comput. Phys.*, 229:3802–3827, 2010. 4, 18

[57] R.R. Nourgaliev, T.N. Dinh, M.-S. Liou, and T.G. Theofanous. The characteristics-based matching method (cbm) for high-speed fluid-fluid flows. *ASME Journal of Fluids Engineering*, 126:586–604, 2004. 1, 2

[58] R.R. Nourgaliev and T.G. Theofanous. High-fidelity interface tracking in compressible flows: Unlimited anchored adaptive level set. *Journal of Computational Physics*, 224:836–866, 2007. 4

[59] R.R. Nourgaliev, S. Wiri, N.T. Dinh, and T.G. Theofanous. On improving mass conservation of level set by reducing spatial discretization errors. *Int. J. Multiphase Flow*, 31:1329–1336, 2005. 4, 18

[60] M. Ohta and M. Sussman. The buoyancy-driven motion of a single skirted bubble or drop rising through a viscous liquid. *Physics of Fluids*, 24, 2012. 18

[61] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A pde based fast local level set method. *J. Comp. Phys.*, 155(2):410–438, Nov 1999. 5

[62] A. Phillipp, M. Delius, C. Scheffczyk, A. Vogel, and W. Lauterborn. Interaction of lithotripter-generated shock waves with air bubbles. *J. Acoust. Soc. Am.*, 93:2496–2509, 1993. 1

[63] J.E. Pilliod and E.G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199(2):465–502, 2004. 18

[64] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics*. Texts in Applied Mathematics. Springer, Paris, FR, 2007. Written for graduate students, researchers. 82

[65] M. Raessi, J. Mostaghimi, and M. Bussmann. Advecting normal vectors: A new method for calculating interface normals and curvatures when modeling two-phase flows. *J. Comput. Phys.*, 226:774–797, 2007. 4

[66] Yuriko Renardy and Michael Renardy. PROST: A Parabolic Reconstruction of Surface Tension for the Volume-of-Fluid Method. *Journal of Computational Physics*, 183(2):400–421, December 2002. 4

[67] T. Ressguier, L. Signor, A. Dragon, and G. Roy. Dynamic fragmentation of laser shock-melted tin: experiment and modelling. *International Journal of Fracture*, 163(1-2):109–119, 2010. 1

[68] R. Saurel and R. Abgrall. A simple method for compressible multifluid flows. *SIAM J. Sci.*

Comput., 21:1115–1145, 1999. 1, 2

[69] S. P. Schofield, M. A. Christon, V. Dyadechko, R. V. Garimella, R. B. Lowrie, and B. K. Swartz. Multi-material incompressible flow simulation using the moment-of-fluid method. *Int. J. Num. Meth. Fluids*, 63:931–952, 2010. 117

[70] Seungwon Shin and Damir Juric. Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. *J. Comp. Phys.*, 180(2):427–470, 2002. 5

[71] S. Singh, J.-K. Choi, and G. Chahine. Optimum configuration of an expanding-contracting-nozzle for thrust enhancement by bubble injection. *J. Fluids Eng.*, 134, 2012. 1

[72] S.Leung and H.Zhao. A grid based particle method for moving interface problems. *J. Comp. Pyhs.*, 228:2993–3204, 2009. 5

[73] S.Leung, J. Lowengrub, and H.Zhao. A grid based particle method for high order geometrical motions and local inextensible flows. *J. Comp. Pyhs.*, 230(7):2540–2561, 2011. 5

[74] A. Staniforth and J. Cote. Semi-lagrangian schemes for atmospheric models - a review. *Mo. Wea. Rev.*, 119:2206–2223, 1990. 34

[75] M. Sussman. A second order coupled levelset and volume of fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187:110–136, 2003. 114, 115

[76] M. Sussman, A. Almgren, J. Bell, P. Colella, L. Howell, and M. Welcome. An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.*, 148:81–124, 1999. 4

[77] M. Sussman and M.Y. Hussaini. A discontinuous spectral element method for the level set equation. *J. Scientific Computing*, 19:479–500, 2003. 4

[78] M. Sussman, P. Smereka, and S.J. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994. 4, 5

[79] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, and R. Zhi-Wei. A sharp interface method for incompressible two-phase flows. *J. Comp. Phys.*, 221(2):469–505, 2007. 18

[80] G. Tryggvason. Numerical simulations of the rayleigh-taylor instability. *J. Comput. Phys.*, 75:253–282, 1988. 117

[81] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100:25–37, 1992. 5

[82] A. Wardlaw and H. Mair. Spherical solutions of an underwater explosion bubble. *Shock and Vibration*, 5:89–102, 1998. 1, 2, 131

[83] A. Wardlaw, Jr. Underwater explosion test cases. Technical Report IHTR 2069, ADB238684, Office of Naval Research, 1998. 1, 66, 93, 95, 127, 129

[84] P. Wesseling, D.R. van der Heul, and C. Vuik. Unified methods for computing compressible and incompressible flows. In *Proceedings of the European Congress on computational methods in applied sciences and engineering*, pages 1–23, Barcelona, Spain, 2000. 3

[85] F. Xiao. Unified formulation for compressible and incompressible flows by using multi-integrated moments i: one-dimensional inviscid compressible flow. *J. Comput. Phys.*, 195:629–654, 2004. 3, 88

[86] F. Xiao, R. Akoh, and I. Satoshi. Unified formulation for compressible and incompressible flows by using multi-integrated moments ii: Multi-dimensional version for compressible and incompressible flows. *J. Comput. Phys.*, 213:31–56, 2006. 3

[87] T. Yabe, Y. Ogata, K. Takizawa, T. Kawai, A. Segawa, and Sakurai K. The next generation cip as a conservative semi-lagrangian solver for solid, liquid and gas. *J. Comp. App. Math.*, 149:267–277, 2002. 1, 2

[88] Takashi Yabe, Feng Xiao, and Takayuki Utsumi. The constrained interpolation profile method for multiphase analysis. *Journal of Computational Physics*, 169(2):556 – 593, 2001. 1

[89] Qinghai Zhang and Philip L.-F. Liu. A new interface tracking method: The polygonal area mapping method. *Journal of Computational Physics*, 227(8):4063 – 4088, 2008. 5, 105

BIOGRAPHICAL SKETCH

Matthew Jemison was born on September 20, 1986 in Ft. McClellan, Alabama. He entered Florida State University in 2005, and quickly decided to study computational mathematics. After working for some time in the area of computer vision and pattern analysis, he decided to pursue an advanced degree with a focus on computational fluid dynamics. He entered the graduate program at Florida State University, and under the supervision of advisor Mark Sussman, graduated with his Ph.D. in Applied and Computational Mathematics in the summer of 2014.

Matthew's research interest include numerical analysis, computational fluid dynamics, multi-phase flow, interface reconstruction, and fluid-structure interaction.