UNIVERSITY OF CAMBRIDGE

Sharp interface schemes for multi-material computational fluid dynamics

Murray Connelly Cutforth

Corpus Christi College

This dissertation is submitted in July 2018 for the degree of Doctor of Philosophy

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Murray Connelly Cutforth
July 2018

UK Ministry of Defence © Crown Owned Copyright 2019/AWE

Abstract

Sharp interface schemes for multi-material computational fluid dynamics

In this thesis we consider the solution of compressible multi-material flow problems, where each material is governed by the Euler equations and the material interface may be considered to be a perfect discontinuity separating macroscopic pure-material regions. Working in the framework of Godunov-type finite volume methods, we develop numerical algorithms for tracking the material interface and evolving fluid states.

For the task of tracking the location of sharp material interfaces, we focus on volume tracking methods due to their ability to conserve mass in highly deformational flows. Three original contributions are presented in this area. First, the accuracy of the volume-of-fluid algorithm is improved through the addition of marker particles. Next, the efficient moment-of-fluid method is presented. This improves the computational efficiency of the moment-of-fluid method by a factor of three by mapping certain quantities during the interface reconstruction step on to a pre-computed data structure. Finally, a general framework for updating volume fractions based on the solution to a quadratic programming problem is presented.

The evolution of fluid states in the full multi-material system is an independent problem. We present developments to two numerical approaches for this problem. The first, the ghost fluid method, is widely used due to the ease in which pure-fluid algorithms can be extended to the multi-material case. We investigate the effect of using a number of different interface tracking methods on the solutions, and find that the conservation errors vary by more than an order of magnitude. The ghost fluid method is then altered such that the ghost state extrapolation step is eliminated from the algorithm, allowing volume fraction-based interface tracking methods to be coupled. In the final chapter, we tackle the same problem using a mixture model-based method. The numerical method presented here is based on work by Miller and Puckett in 1996, in which a six-equation system using the assumption of pressure and velocity equilibrium was used to model two-material flow. We have thoroughly overhauled this method, incorporating Riemann solvers developed for the five-equation system, as well as a robust implicit energy update. We present numerical results on a range of one- and two-dimensional shock-interface interaction test problems which demonstrate the ability of the method to match the solutions from the five-equation model while maintaining a perfectly sharp material interface.

Murray Connelly Cutforth

Acknowledgements

First and foremost I would like to thank my parents, Isobel and Peter, as well as the rest of my family. Without their support none of this would have been possible. Liat also deserves a special mention for bringing her light into a tough final year.

Many thanks are due to my two academic supervisors, Phil Barton and Nikos Nikiforakis, for their guidance over the course of my studies. I have been lucky to benefit from help from the rest of the Laboratory for Scientific Computing, in particular my room-mates in 3.78.

Finally, I am grateful to the many friends I have tied onto a rope with during the last four years for providing some fantastic adventures along the way.

Contents

CHAPTER 1

Introduction

Amongst the first partial differential equations ever to be written down [1] were the set of equations describing the motion of a perfect fluid, described by Leonhard Euler over 250 years ago in [2]. Euler's equations can be obtained by taking the limit of zero heat conduction and zero viscosity in the more general Navier-Stokes equations, and are an accurate model for real-life flows in which fluid motion is dominated by inertial effects. Written in conservation form, the two-dimensional compressible Euler equations describe the evolution of mass, momentum, and energy of a continuous fluid:

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = 0, \tag{1.1}$$

where

$$\mathbf{U}(\mathbf{x}, t) = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{bmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{bmatrix},$$

and the total energy is defined by $E = \rho\left(e + \frac{1}{2}(u^2 + v^2)\right)$. Since the Euler equations contain more variables than equations, we also require a closure condition. The closure condition depends on the material under consideration, and often takes the form of a caloric equation of state:

$$e = e(p, \rho). \tag{1.2}$$

The nonlinear nature of the Euler equations makes the mathematical analysis very difficult, and many of the interesting applications feature complicated boundary conditions and free surfaces. As a consequence, computational study is particularly well-suited to these equations.

1.1 Multi-material flow

It is beyond the scope of this work to give a full introduction to the many different flow regimes involving more than one material; a brief survey is given in [3]. We are interested in problems involving free surfaces– sharp interfaces between macroscopic single material regions whose location is not known *a priori*. This regime is sometimes referred to as *separated flow*, and is distinct from *dispersed flow* in which large numbers of invidual droplets are present throughout a continuous phase [4]. In separated flows, the geometry of the interface may play a critical role in governing the overall flow solution.

The numerical methods developed in this thesis are based on the assumption of a perfectly sharp material interface. This assumption is valid in high-speed flows where the mixing of materials occurs over a longer time scale than the process under investigation. High speed flows are often well-approximated by the Euler equations, since unless the material under question has a particularly large kinematic viscosity, the flow dynamics will be dominated by inertial forces. This concept is conveniently encapsulated by the Reynolds number

$$\mathrm{Re} = \frac{uL}{\nu}, \tag{1.3}$$

where u is fluid velocity, L is the length scale of the flow, and ν is the kinematic viscosity. We consider multi-material flows in which $\mathrm{Re} \gg 1$, where the Euler equations are applicable.

As an example of this, consider the case in which a Mach 1.2 shock wave in air is incident on a helium bubble over a $\mathcal{O}(10^{-1})$m length scale. In this case both the molecular diffusivity and the kinematic viscosity are approximately 0.2cm^{2}s^{-1} [5]. Over the course of a 400μs interaction this translates to both an effective viscous length scale and an mixture zone width of $\mathcal{O}(10^{-4})$m. Direct numerical simulation of the viscous structures and the mixture zone would therefore have to employ a mesh spacing of at least one order of magnitude smaller than this length scale [6]. Such a simulation, which would involve between 500 million and 1 billion computational cells, is only possible on current supercomputers. Therefore, our assumptions of a sharp interface and of inviscid flow are valid on the computational mesh sizes which are accessible to us.

Although the numerical methods in this thesis are developed for the high speed flow case, there is another common scenario in which material interfaces remain sharp: low speed, surface tension-driven flow, such as air-water flows over practical (millimetre to metre) scales. In this case, the interface remains sharp due to inter-molecular forces. The interface tracking methods developed in chapter 3 of this thesis may be applied equally well to these flows.

1.1.1 Applications

The dynamics of many real-life flows are well-satisfied by these assumptions, and numerical methods of the type considered in this thesis have previously been used to investigate a wide range of phenomena, including:

- Astrophysical flows, such as the hydrodynamics of supernovae [7]. Interface tracking methods may be used to track the evolution of the thermonuclear flame front.

- Inertial confinement fusion, where a small pellet containing deuterium and tritium is compressed with a petawatt laser pulse, igniting a nuclear fusion reaction at the point of maximum compression in the centre. A significant challenge is the appearance of hydrodynamic instabilities between the material layers in the pellet which cause premature mixing of hot and cold fuel. Multi-material numerical methods have been used to quantify the dependence of these instabilities on the initial interface perturbations [8].

- The dynamics of high velocity impact problems, such as the impact of liquid droplets travelling at 500ms^{-1} considered in [9], are particularly well-approximated by the Euler equations. These flows are significant in a range of industrial processes. High velocity impact of solids may also be simulated using identical numerical methods, such as the level set ghost fluid approach presented in [10].

- Underwater explosion problems such as [11] are relevant in defence applications, where numerical simulation of the interaction between underwater shock waves and ship hulls are significantly cheaper than real-life trials. In the low-speed flow regime, interface tracking methods are also the preferred approach for incorporating the free surface between air and water in ship wake simulations [12].

- Bubble collapse, such as the investigation of the hydrodynamics underlying sonoluminescence presented in [13].

- In surface tension-driven flows, phenomena such as wave breaking and jet atomisation exhibit elaborate interface geometries. These flows are typically approximated by the incompressible Navier-Stokes equations, for example [14, 15].

1.1.2 Free surface boundary condition

The dynamics of the material interface are determined by two boundary conditions, known as the kinematic and dynamic conditions.

The kinematic condition on a free surface relates the fluid velocity, $\mathbf{u}_f$, to the interface velocity, $\mathbf{u}_i$. Since we are working under the assumption of immiscibility, the free surface is impermeable to each material, hence:

$$\mathbf{u}_f \cdot \mathbf{n} = \mathbf{u}_i \cdot \mathbf{n}, \tag{1.4}$$

where $\mathbf{n}$ is the unit vector normal to the interface. In effect, this states that particles on the interface remain on the interface, although they may move along the interface. This is known as the 'slip' condition. The fluid velocity may be further constrained with the 'no-slip' condition:

$$\mathbf{u}_f = \mathbf{u}_i. \tag{1.5}$$

The dynamic condition describes the balance of forces across the interface. We limit the scope of this work to the dynamic condition in which pressure is continuous across the interface:

$$p_1 - p_2 = 0. \tag{1.6}$$

We refer the reader to [16] for an introduction to incorporating another common dynamic condition, surface tension, into multi-material numerical methods.

1.2 Eulerian numerical methods

In the simulation of multi-material flows where material distortions are large it is most convenient to employ Eulerian (fixed) computational grids, allowing fluid to move past each element. The converse approach, in which a Lagrangian (moving) grid follows the fluid motion, is impractical due to the frequent re-meshing operations which would be required. It should however be noted that for multi-material problems involving small deformations, Lagrangian methods offer superior accuracy. Since the introduction of early Lagrangian methods such as [17], progress has been made in the form of the free-Lagrange method [18], and arbitrary-Lagrangian-Eulerian methods [19]. Nevertheless, due to the anticipated future application of the numerical methods developed in this thesis to problems featuring extreme interface deformations, we adopt the Eulerian approach for all numerical methods considered.

Eulerian numerical methods for the solution of the hyperbolic conservation laws (of which the Euler equations are one example) may be classified into three broad families. The simplest, and most intuitive, is the finite difference approach, where the solution value is stored at a set of discrete points. The second approach, the finite element method, is generally better-suited to structural analysis, and approximates the solution by some function over each discretised volume. However, the most common approach in computational fluid dynamics is the finite volume approach, in which the domain is divided into volume elements, and the governing conservation laws are solved separately in each element. These methods have the advantage of being inherently conservative, an important property to ensure the correct treatment of solution discontinuities such as shocks. The finite volume methods which we use in single-fluid regions are fully described in chapter 2.

1.3 Multi-material flow solvers

In Eulerian numerical methods for multi-material problems the material interface must be represented as an internal boundary. This general problem has yet to find a universally satisfactory solution, and a vast range of multi-material numerical methods continue to see development.

In this section we present a brief survey of the most significant approaches for the extension of finite volume methods to multi-material flow. Two of these approaches, the ghost fluid method, and the VOF-based mixture method, will be investigated further in this thesis, and a full literature review of these two fields can be found in chapters 4 and 5 respectively.

At this point we take note of several fundamentally different approaches which have been shown to extend well to the multi-material case. The first, smoothed particle hydrodynamics (SPH), is a meshless method where the fluid is discretised into a set of Lagrangian particles and fluid quantities are obtained by weighting the contribution of each particle according to some kernel function [20]. SPH methods have the advantage of not requiring any special treatment of free surfaces, although they are not as accurate as modern Riemann solver-based finite volume methods [21]. Another family of meshless methods has been developed from the peridynamic theory of continuum dynamics [22], which is particularly well-suited for problems involving fracture since spatial derivatives (which do not exist on crack surfaces) are not evaluated. The final approach which we mention here is the lattice Boltzmann method, which evolves particles in discretised space. See [23] for an overview of this intruiging recent idea.

1.3.1 Diffuse interface methods

In the class of diffuse interface methods the governing equations of the system are modified in order to account for the multi-material nature of the problem. For finite volume methods, this results in the mixture of both materials in cells on the interface, which then requires the definition of a mixture equation of state which describes the material properties of this mixture. Early approaches augmented the Euler equations with one or more advection equations for equation of state parameters (e.g. [24, 25]), although this requires the same equation of state to be used for both fluids, and is prone to pressure oscillations at the interface.

The most general two-material model consists of equations for the evolution of density, momentum, and energy of each material, plus an equation for material volume fraction [26]. This seven-equation system (known as the Baer-Nunziato model) is numerically complex to solve, and where possible simplifying assumptions are brought to bear. A range of reduced models exist, ranging from three to six equations depending on the physics of the problem at hand. In particular, if the velocities and pressures of each material are assumed to reach equilibrium faster than other characteristic timescales of the flow, a five-equation model may be obtained [27–29]. The five-equation model is attractively simple and has been used extensively, for example [30, 31]. However, the material interface is represented by a contact discontinuity in the characteristic function which will continuously diffuse over the course of a simulation. Many approaches to reducing this unphysical interface diffusion have already been published [32–34], and it remains an active area of research.

Overall, diffuse interface methods are a simple and efficient approach to multi-material flow, since the same equations are solved with the same numerical methods throughout the domain. Their most significant shortcoming is poor interface resolution, an issue which is addressed in chapter 5 of this thesis.

1.3.2 VOF-based mixture methods

A related approach to diffuse interface methods is the sharp interface volume-of-fluid (VOF) approach of Miller and Puckett [35], which may be thought of as lying half-way between a cut-cell method and a diffuse interface method. An interface tracking method is used to maintain a sharp interface, but material in interfacial cells is allowed to mix in order to compute the intercell fluxes. Miller and Puckett's approach was introduced over 20 years ago, and despite the fact that this type of method has been widely implemented in multi-material codes developed by the various national laboratories (for example CTH [36]), the original method of Miller and Puckett has not experienced further direct development. Further details of the algorithms commonly used in the Eulerian codes developed by national laboratories are provided by Benson in [37], where it is speculated that national security considerations have limited the development of these methods in the literature.

1.3.3 Ghost fluid methods

Ghost fluid methods, originating from [38], are the least complicated approach to capturing internal boundaries, both in the theory and implementation. They maintain a separate grid for each material, and usually employ a level set interface tracking method to identify the real material in any given grid cell. The effect of the material interface is modelled by setting suitable states in the ghost cells, and then the conserved variables in both grids are evolved according to the unmodified Euler equations using any single-material flow solver. Contemporary ghost fluid methods such as [39, 40] use the solution of a mixed Riemann problem across the interface to set these states. The ghost fluid method can suffer from significant loss of conservation, due to the multi-valued flux at the interface as well as due to regularisation of the level set field. Nevertheless, they remain a popular approach due to their simplicity of implementation.

1.3.4 Cut-cell methods

In cut-cell methods a Cartesian mesh is used, with some volume elements divided into two subvolumes along the material interface. The subvolumes may be arbitrarily small, which presents a problem as the stable time step in a finite volume scheme is proportional to the cell size. This 'small-cell' problem has been solved through a number of different approaches, including cell merging [41], the h-box method [42], and flux stabilisation [43]. At present, cut-cell methods are most commonly used for motionless solid boundaries. The extension to moving solid boundaries has been achieved, for example in [44], and multi-fluid problems in [45], but the implementation of the method is complicated in this case. Cut-cell methods also typically employ an interface tracking method to locate the interface.

Cut-cell methods share many similarities with the family of immersed boundary methods [46], which were originally designed for biological fluid-structure interaction problems. Unlike cut-cell methods, in the immersed boundary method the underlying Cartesian mesh is not modified, and the effect of the boundary is instead imposed on the solution through source terms in the governing equations.

1.4 Interface tracking methods

Tracking the position of the material interface is a related but independent problem to determining the dynamics of a multi-material system. Of the multi-material flow solvers mentioned in the preceding section, all but the diffuse interface approach employ an interface tracking method. The problems of computing the fluid dynamics and tracking the interface are tightly coupled, since the interface evolves according to the fluid velocity, and the fluid states depend on the position of the interface.

Over previous decades a large family of numerical methods has emerged to represent and evolve the location of a sharp interface between multiple immiscible materials within a fluid dynamics simulation. Interface tracking methods can be divided into two categories depending on whether the interface is represented explicitly or implicitly.

Explicit methods place marker particles or fit a moving mesh – of codimension one – on the interface. Foremost amongst these is the front tracking method [47, 48]. This offers very high accuracy in tracking interface deformations, but requires the implementation of special procedures to handle changes in topology, such as break-up and coalescence, and to re-normalise the distance between interface markers – these procedures introduce significant programming complexity, particularly when extended to three dimensions.

The great advantage of implicit interface representation is that these topology changes emerge automatically. Two major types of implicit interface representations exist.

1. The volume-of-fluid (VOF) method uses a piecewise-constant indicator function to denote material type at each point in space. The term 'VOF method' has come to describe a large family of methods derived from the original description by Hirt and Nichols in 1981 [49]; other pioneering work includes the SLIC method [50]. A useful overview is given in [51].

2. The level set method, proposed by Osher and Sethian in [52], is the principal alternative implicit interface tracking method. Rather than the piecewise-constant indicator function of the VOF method, in the level set method the interface is defined as the set of all points on the zero level contour of a scalar distance function which is arbitrarily assigned opposite signs to indicate the interior and exterior regions of the tracked material.

These two approaches have a complementary set of advantages and disadvantages. While the VOF method possesses local mass conservation, the level set method suffers from mass change caused by the need to periodically reinitialise the level set field to a signed distance function. This problem is particularly severe in highly deformational flows in which interface features are under-resolved, although progress has been made in reducing the problem [53]. On the other hand, unlike the piecewise-constant indicator function in the VOF method, the smooth level set function is differentiable on the interface, making it straightforward to calculate geometric properties such as the normal vector and interface curvature. This also means that the evolution equation can be discretised with high-order finite differences [54], which in general lend themselves to more straightforward implementation than the geometric subroutines associated with VOF methods. The deficiencies in each interface tracking method have led to the development of a number of hybrid methods, such as the level set-VOF method [55], and the particle level set method [56]. A thorough review of the VOF literature is presented in chapter 3.

1.5 Thesis outline

This thesis consists of work in three related areas of study, unified by the common aim of improving the computational algorithms used in multi-material fluid dynamics simulations. In the pursuit of this aim, we have explored enhanced interface tracking methods, internal boundary representations in flow solvers, and novel couplings between interface tracking methods and flow solvers.

1. We begin in chapter 2 with an introduction to Godunov-type finite volume methods, and a description of the single-fluid numerical methods which the fluid simulations in chapters 4 and 5 are based upon.

2. In chapter 3 we focus on interface tracking methods. This chapter aims to contribute towards the development of increasingly accurate and efficient interface tracking methods. We focus on methods which use a volume fraction interface representation (rather than a level set field) due to their mass conservation of under-resolved features. Three original contributions are presented in this chapter.

    (a) First, a method for improving the accuracy of the VOF method by introducing marker particles on the interface is described. This method avoids the usual disadvantages associated with marker particles by re-seeding every time step.

    (b) The current state-of-the-art in volume tracking, the moment-of-fluid method, is reformulated such that the efficiency of the method is significantly improved, becoming comparable with other less accurate VOF methods. Previous implementations of the moment-of-fluid method are computationally expensive due to the repeated solution of a numerical optimisation problem. We make each objective function evaluation in this optimisation problem trivially cheap by pre-computing certain values and mapping the objective function domain onto these pre-computed values.

    (c) Our final contribution to interface tracking methods is the development of a mathematical framework for updating volume fractions in a fully-conservative manner which does not require a conservative reconstruction. In this approach, we present the volume fraction update as a quadratic programming problem, and derive one possible solution using the active set method. This framework permits volume fractions to be coupled with other interface representations, and as an example we describe one such coupling with a level set field.

3. In chapter 4 the ghost fluid approach to multi-material flow is surveyed and developed. We begin with a literature review, and a quantitative comparison of the various types of ghost fluid method. After demonstrating that the choice of interface tracking method can have a dramatic effect on the conservation properties of the ghost fluid method, we present a novel formulation of the ghost fluid method in which the extrapolation step is eliminated from the algorithm. This allows the ghost fluid method to be coupled to any volume fraction-based interface tracking

method, rather than being limited to level set interface tracking methods. We present numerical results on several gas-gas problems, including shocked bubbles and Richtmyer-Meshkov instability test cases, comparing conservation and interface resolution properties of both level set and VOF-based ghost fluid methods.

4. In chapter 5 a novel sharp-interface numerical scheme is developed for compressible two-material flow. This scheme follows from work done by Miller and Puckett over 20 years ago [35]. As well as developing a significantly more robust update formula for mixed cells, we simplify the operation of the method by introducing modern numerical techniques developed for five-equation diffuse interface methods. Specifically, the concept of an "effective single phase" in the homogenisation method is eliminated through the use of the mixture model introduced by Massoni et al. [57], and it is shown how approximate Riemann solvers developed for the five-equation model can be used in place of the original Hugoniot approximation. The method employs a volume fraction-based interface tracking method, and maintains a perfectly sharp interface indefinitely. Numerical results are presented on various two-material gas-gas and gas-water problems which demonstrate the potential of this neglected approach.

1.6 Code Development

All numerical results presented in this thesis were obtained with various C++11 codes written from scratch by the author. The following three external libraries were used in this software. Eigen [58], for vector manipulation and linear algebra. OpenMP [59] for multi-core parallelisation. NLopt [60] for nonlinear optimisation algorithms.

CHAPTER 2

Godunov-type finite volume methods

The numerical solution of the Euler equations (and other hyperbolic conservation laws) is made particularly interesting by the phenomenon of shock waves- discontinuous solutions which may arise even from smooth initial data. Capturing these solutions accurately and without resorting to special procedures motivated the development of Godunov-type finite volume methods, originating from [61].

The underlying concept of finite volume methods is to divide the domain into a number of cells, and store the integral average value of the conserved variables in each cell. In one spatial dimension we have the conservation law

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0. \tag{2.1}$$

By integrating the conserved variable $\mathbf{U}$ over the cell $[x_{i-1/2}, x_{i+1/2}]$ at time level t^n, we store:

$$\mathbf{U}_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}(x, t^n)\ \mathrm{d}x, \tag{2.2}$$

where $\Delta x = x_{i+1/2} - x_{i-1/2}$. Godunov-type numerical methods are constructed by integrating the conservation law, equation 2.1, in space over cell i and then in time from t^n to t^{n+1}:

$$\int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}(x,t)_t\ \mathrm{d}x\ \mathrm{d}t = -\int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{F}(\mathbf{U}(x,t))_x\ \mathrm{d}x\ \mathrm{d}t. \tag{2.3}$$

This simplifies to the conservative update formula over the time step of length Δt:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-1/2} - \mathbf{F}_{i+1/2}\right), \tag{2.4}$$

where

$$\mathbf{F}_{i-1/2} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U}(x_{i-1/2}, t)) \, \mathrm{d}t. \tag{2.5}$$

It is through approximations to this flux function, $\mathbf{F}_{i-1/2}$, that Godunov methods are defined.

2.1 The Riemann problem

The Riemann problem is integral to Godunov-type methods. It is the simplest possible non-trivial initial value problem for the conservation laws under investigation, and is solved at every cell edge at every time step in the Godunov method. The Riemann problem consists of the solution of equation 2.1 with initial data consisting of two constant states separated by a discontinuity:

$$\mathbf{U}(x, t=0) = \begin{cases} \mathbf{U}_L & x < 0 \\ \mathbf{U}_R & x > 0 \end{cases} . \tag{2.6}$$

It is possible to find an exact solution to the Riemann problem for the Euler equations, but at the cost of an iterative procedure. Consequently, a number of approximate Riemann solvers have been developed, which in the context of the entire numerical method do not affect solutions, but offer better computational efficiency. Foremost amongst these are the HLLC [62] and Roe [63] families of schemes. The exact solution to the Riemann problem for the Euler equations with the stiffened gas equation of state is derived in appendix A.

2.2 The Godunov method

The original first-order Godunov method assumes a piecewise-constant profile of the conserved variables, and estimates the value of $\mathbf{F}_{i-1/2}$ from the solution to the Riemann problem between the states $\mathbf{U}^n_{i-1}$ and $\mathbf{U}^n_i$ on the $\frac{x}{t} = 0$ characteristic. Since the solution is the Riemann problem is self-similar (constant along each line $\frac{x}{t} = \text{const}$), the value of $\mathbf{F}_{i-1/2}$ will remain constant across the time step, which must be set small enough that the waves emerging from each cell edge do not cross adjacent cell edges. This vastly simplifies the integral in equation 2.5 used in the conservative update.

Today, Godunov methods are a mature numerical technology. Their accuracy has been continually improved, initially through second-order accurate approaches such as the piecewise-parabolic method [64] and MUSCL-Hancock method [65], and more recently through sophisticated sub-cell reconstructions such as WENO [66]. For more information on Godunov-type methods and the Riemann problem we refer the reader to the comprehensive books from Toro [67] and Leveque [68].

2.3 The MUSCL-Hancock method

The multi-material algorithms presented in this thesis are based on the MUSCL-Hancock method for the solution of a system of hyperbolic conservation laws. The method belongs to the reconstruct-evolve-average family of schemes, and exhibits second-order accuracy in space and time.

2.3.1 One dimensional scheme

The one dimensional MUSCL-Hancock method may be applied to any hyperbolic conservation law of the form given in equation 2.1. It utilises the explicit, fully conservative update procedure:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-\frac{1}{2}} - \mathbf{F}_{i+\frac{1}{2}}\right), \tag{2.7}$$

where $\mathbf{F}_{i-\frac{1}{2}}$ is the flux of the state on the $\frac{x}{t} = 0$ characteristic of the cell edge, where the initial data is obtained by a reconstruction and evolution method. We describe the primitive variable variant of the scheme, which is necessary when applying it to the five equation system considered in chapter 5 of this thesis. In this variant the conservation law is re-written in quasi-linear form as

$$\mathbf{W}_t + \mathbf{A}(\mathbf{W})\mathbf{W}_x = 0, \tag{2.8}$$

where

$$\mathbf{W} = \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}, \qquad \mathbf{A}(\mathbf{W}) = \begin{pmatrix} u & \rho & 0 \\ 0 & u & \frac{1}{\rho} \\ 0 & \rho c^2 & u \end{pmatrix}. \tag{2.9}$$

Following the procedure in [67], the first step is the extrapolation of the primitive variables on either side of the current cell edge under consideration, under the assumption of a piecewise-linear reconstruction:

$$\mathbf{W}_{i+1}^L = \mathbf{W}_{i+1} - \frac{1}{2}\Delta_{i+1}, \quad \mathbf{W}_i^R = \mathbf{W}_i + \frac{1}{2}\Delta_i \tag{2.10}$$

where $\Delta_i = \Delta_i(\mathbf{W}_i - \mathbf{W}_{i-1}, \mathbf{W}_{i+1} - \mathbf{W}_i)$ is a slope difference vector with the minmod limiter applied to every component:

$$\text{minmod}(x, y) = \begin{cases} \max(0, \min(x, y)) & \text{if } x > 0, \\ \min(0, \max(x, y)) & \text{if } x \leq 0. \end{cases} \tag{2.11}$$

These extrapolated boundary vectors are then evolved by half a time step:

$$\bar{\mathbf{W}}_{i+1}^{L} = \mathbf{W}_{i+1}^{L} - \frac{\Delta t}{2\Delta x}\left(\mathbf{A}(\mathbf{W}_{i+1})\Delta_{i+1}\right), \tag{2.12}$$

$$\bar{\mathbf{W}}_{i}^{R} = \mathbf{W}_{i}^{R} - \frac{\Delta t}{2\Delta x}\left(\mathbf{A}(\mathbf{W}_{i})\Delta_{i}\right). \tag{2.13}$$

Finally, $\bar{\mathbf{W}}_i^R$ and $\bar{\mathbf{W}}_{i+1}^L$ are used as initial states in the Riemann problem, which computes the flux across edge $i + 1/2$.

The size of the time step Δt is limited by stability considerations. We use

$$\Delta t = \frac{C_{\text{cfl}}\Delta x}{S_{\max}}, \tag{2.14}$$

where $C_{\text{cfl}} \in [0, 1]$ and the largest wave speed is estimated as $S_{\max} = \max_i(|u_i| + c_i)$, where u_i and c_i are the fluid velocity and sound speed in each cell.

2.3.1.1 Verification

Since all codes used for the numerical results presented in this thesis were developed from scratch, a brief verification of the one dimensional MUSCL-Hancock method with an exact Riemann solver is now presented. The order of convergence of the method on two test problems is measured and compared to the Godunov method. The error in a scalar variable $q(x,t)$ is measured as

$$E_i^n = q_i^n - q_{exact}(x_i, t^n) \tag{2.15}$$

where q_i^n is the integral average of the quantity computed by our scheme and $q_{exact}(x_i, t^n)$ is the value of the exact solution at the cell centre. Since the cell centre solution agrees with the integral average value to within $\mathcal{O}(\Delta x^2)$ when the solution is smooth, this comparison is suitable for the second-order MUSCL-Hancock method. At a given time the error is measured using the 1-norm:

$$\|E\|_1 = \Delta x \sum_{i=1}^{N} |E_i|. \tag{2.16}$$

The first test case is the advection of a Gaussian density profile with constant pressure and velocity. The initial conditions are

$$\begin{cases} \rho(x) = 1 + Ae^{-\frac{(x-\mu)^2}{2\sigma^2}} \\ u = 1 \\ p = 0.0001 \\ \gamma = 1.4 \end{cases} \tag{2.17}$$

where $A = 1000$, $\mu = 0.5$, $\sigma = 0.1$. Periodic boundary conditions are used, the domain is $[0, 1]$, $C_{CFL} = 0.8$, and the error is measured at time $t = 10$ when the profile has completed 10 full translations of the domain. The convergence of the density errors are shown in figure 2.1. As expected the MUSCL-Hancock converges with second order accuracy on this smooth problem.

The second test case validates the performance of the solvers on solutions involving shock waves. We present results from shock tube test case 1 from Toro [67] in figure 2.2. Since this problem involves shock waves both methods are expected to exhibit first order convergence, and the MUSCL-Hancock method outperforms the Godunov method as expected. This concludes the validation of the one-dimensional flow solver.

2.3.2 Two dimensional scheme

In two spatial dimensions, a hyperbolic conservation law of the following form is solved:

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = 0. \tag{2.18}$$

There are two approaches to the solution of this problem.

2.3.2.1 Strang splitting

Under the dimensional splitting approach, two one-dimensional problems are solved alternately:

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0, \tag{2.19}$$

followed by

$$\mathbf{U}_t + \mathbf{G}(\mathbf{U})_y = 0, \tag{2.20}$$

in which the momentum component orthogonal to each sweep direction is treated as a passively advected quantity. Denoting the operator for advancing in the x-direction across time interval Δt (equation 2.19) as $\mathcal{X}(\Delta t)$, and in the y-direction (equation 2.20) as $\mathcal{Y}(\Delta t)$, Strang suggested the following formally second-order scheme [69]:

$$\mathbf{U}^n = \mathcal{X}(\frac{\Delta t}{2})\left(\mathcal{Y}(\Delta t)\mathcal{X}(\Delta t)\right)^{n-1}\mathcal{Y}(\Delta t)\mathcal{X}(\frac{\Delta t}{2})\mathbf{U}^0, \tag{2.21}$$

which we apply in the dimensionally-split flow solver presented in chapter 5.

2.3.2.2 Unsplit MUSCL-Hancock

In the second approach to extending Godunov methods to two-dimensional flow, the numerical method is reformulated so that the fluxes in both dimensions are considered

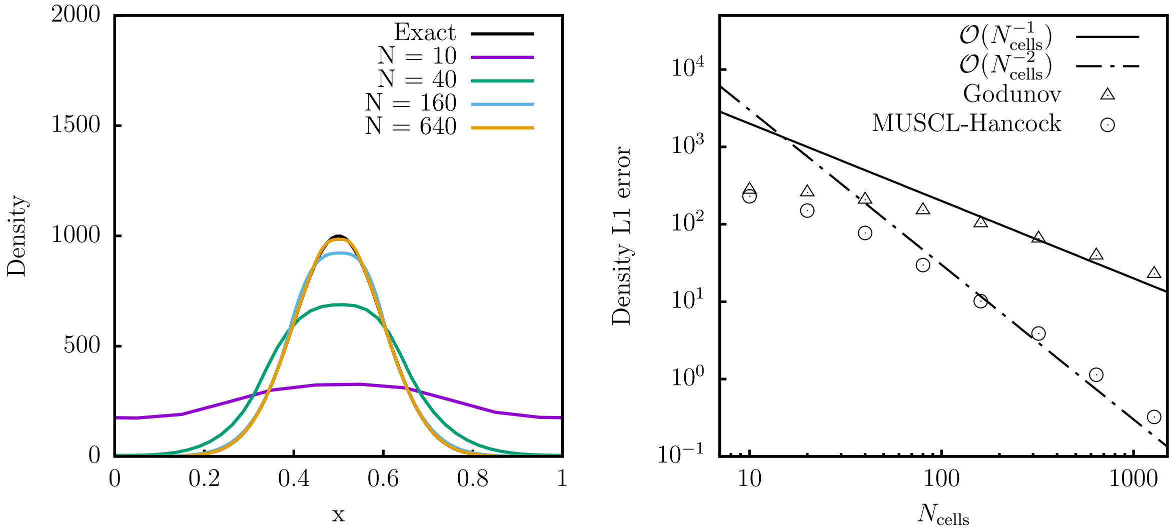


Figure 2.1: Convergence of the 1-norm of the density error for flow solvers in the Gaussian density profile advection test. Both solvers use the exact solution to the Riemann problem. The plot on the left shows the density profile at time $t = 10$ for various resolutions of the MUSCL-Hancock method.

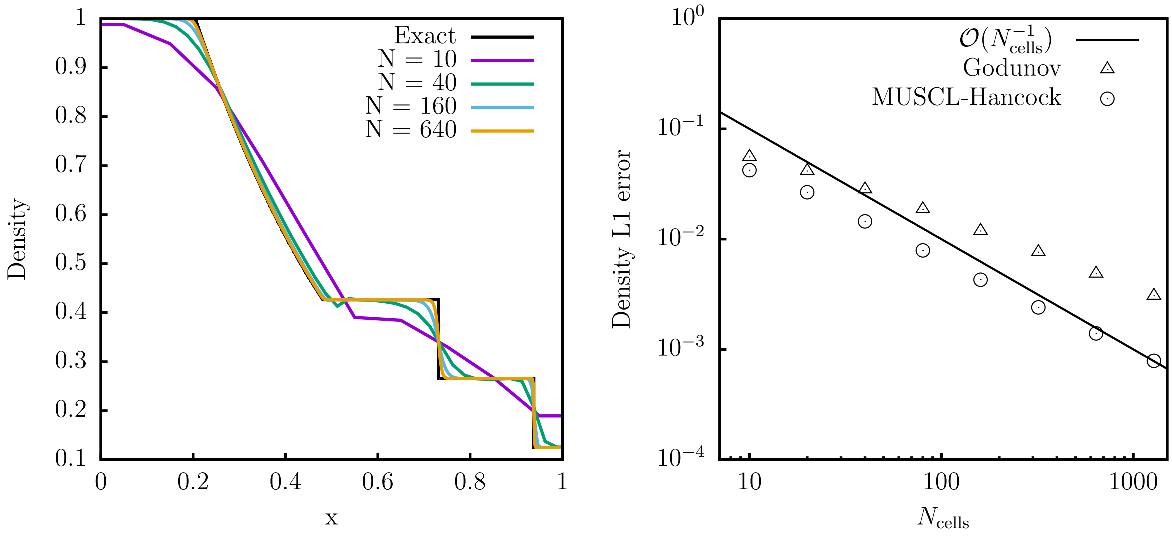


Figure 2.2: Convergence results from test case 1 of Toro in the validation of the 1D flow solver. Both solvers use the exact solution to the Riemann problem. The left-hand plot shows the density profile at time $t = 0.25$ in the MUSCL-Hancock solution.

simultaneously. Once again we use the primitive-variable variation of the method, which uses the quasi-linear form of equation 2.18:

$$\mathbf{W}_t + \mathbf{A}(\mathbf{W})\mathbf{W}_x + \mathbf{B}(\mathbf{W})\mathbf{W}_y = 0. \tag{2.22}$$

The numerical method is similar to the one-dimensional scheme described earlier. The first step is the piecewise-linear reconstruction of the primitive variables, which are extrapolated to the left and right cell edges (equation 2.10), as well as the top and bottom:

$$\mathbf{W}^T_{i,j} = \mathbf{W}_{i,j} + \frac{1}{2}\Delta^{(y)}_{i,j}, \quad \mathbf{W}^B_{i,j} = \mathbf{W}_{i,j} - \frac{1}{2}\Delta^{(y)}_{i,j} \tag{2.23}$$

where $\Delta^{(y)}_{i,j} = \Delta^{(y)}_{i,j}\left(\mathbf{W}_{i,j} - \mathbf{W}_{i,j-1}, \mathbf{W}_{i,j+1} - \mathbf{W}_{i,j}\right)$ is a slope difference vector with the minmod limiter (equation 2.11) applied to every component. In the same way as for the one dimensional scheme, all four boundary extrapolated values are approximately evolved by a half time step, for example:

$$\tilde{\mathbf{W}}^T_{i,j} = \mathbf{W}^T_{i,j} + \frac{\Delta t}{2\Delta x}\left(\mathbf{A}(\mathbf{W}_{i,j})\Delta^{(x)}_{i,j}\right) + \frac{\Delta t}{2\Delta y}\left(\mathbf{B}(\mathbf{W}_{i,j})\Delta^{(y)}_{i,j}\right). \tag{2.24}$$

The conservative update formula then utilises these extrapolated states as the initial conditions in four Riemann problems:

$$\mathbf{U}^{n+1}_{i,j} = \mathbf{U}^n_{i,j} + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-1/2,j} - \mathbf{F}_{i+1/2,j}\right) + \frac{\Delta t}{\Delta y}\left(\mathbf{G}_{i,j-1/2} - \mathbf{G}_{i,j+1/2}\right), \tag{2.25}$$

where for example:

$$\mathbf{G}_{i,j+1/2} = \mathbf{G}_{i,j+1/2}\left(\tilde{\mathbf{W}}^T_{i,j}, \tilde{\mathbf{W}}^B_{i,j-1}\right). \tag{2.26}$$

This unsplit MUSCL-Hancock method is applied in the ghost fluid method simulations presented in chapter 4.

2.3.2.3 Verification

The two-dimensional MUSCL-Hancock method is verified on a similar Gaussian advection test problem. In this case the initial conditions are:

$$\begin{cases} \rho(\mathbf{x}) = 1 + Ae^{-\frac{|\mathbf{x}-\mu|^2}{2\sigma^2}} \\ u = 1 \\ v = 1 \\ p = 0.0001 \\ \gamma = 1.4 \end{cases}, \tag{2.27}$$

where $\mu = (0.5, 0.5)^T$, and A and σ take the same values as previously. The computational domain is $[0, 1] \times [0, 1]$. We measure the error at time $t = 10$, after the profile has traversed the domain 10 times. With the dimensionally-split approach, we use $C_{CFL} = 0.8$, while the unsplit method is applied with $C_{CFL} = 0.4$. The convergence of the L1 density errors are shown in figure 2.3, and a sample of the final density profiles are shown in figure 2.4. The expected second-order convergence rate is achieved by both approaches, although the split methods exhibit smaller errors at a given resolution.

The two-dimensional solvers were validated on problems involving shock waves by applying the five shock tube problems from Toro in each coordinate direction. For brevity, these results are not presented here as the expected behaviour was observed.

2.4 Boundary conditions

In this work we apply boundary conditions to the edge of the domain by creating several fictitious cells, and populating them with appropriate states so that the resulting numerical flux across the edge of the domain corresponds to the desired boundary condition. Three boundary conditions are applied.

The first, the transmissive boundary condition, allows waves to leave the domain without reflection. This boundary condition is implemented by setting each fictitious state to the real fluid state reflected in the domain edge. For example, in one dimension the real states are denoted by the subscripts $[1, N]$, and the fictitious states on the right hand side are set according to:

$$\mathbf{U}_{N+k} = \mathbf{U}_{N+1-k}, \quad k = 1, 2, ... \tag{2.28}$$

The reflective boundary condition corresponds to a solid wall along the edge of the domain. All waves are fully reflected back into the domain. Once again using the right hand side of a one-dimensional domain as an example, this boundary condition is applied as:

$$\mathbf{U}_{N+k} = \mathcal{R}\left(\mathbf{U}_{N+1-k}\right), \quad k = 1, 2, ..., \tag{2.29}$$

where

$$\mathcal{R}\left(\mathbf{U}\right) = \begin{pmatrix} \rho \\ -\rho u \\ E \end{pmatrix}. \tag{2.30}$$

The final boundary condition we have implemented is the periodic boundary condition. This wraps the spatial domain around, so that:

$$\mathbf{U}_{N+k} = \mathbf{U}_k, \quad k = 1, 2, ... \tag{2.31}$$

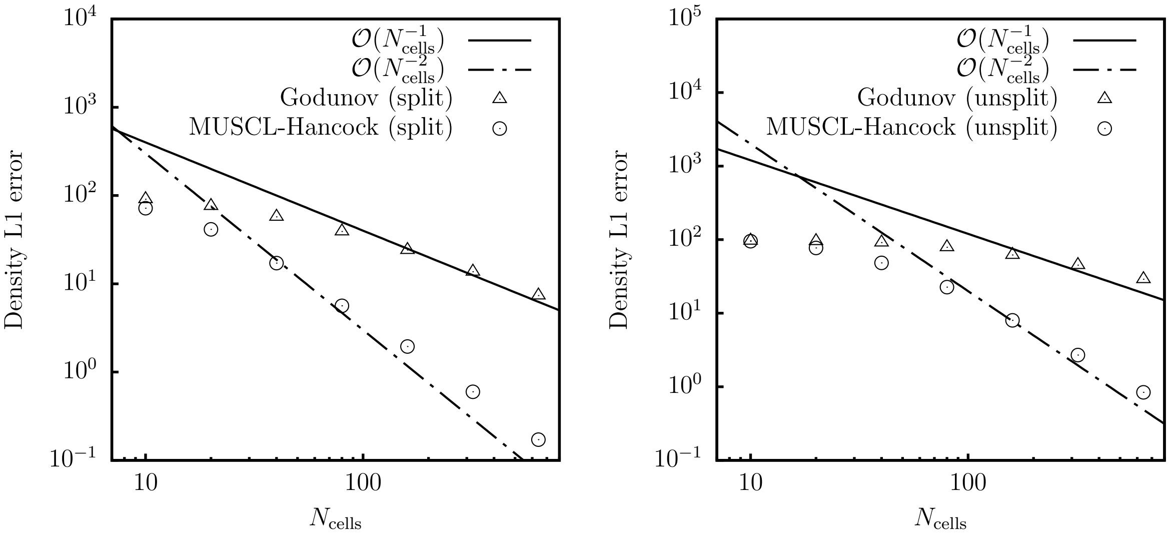


Figure 2.3: Convergence of the 1-norm of the density error for flow solvers in the two-dimensional Gaussian density profile advection test. Both solvers use the exact solution to the Riemann problem. N_{cells} denotes the number of grid cells in each coordinate direction.

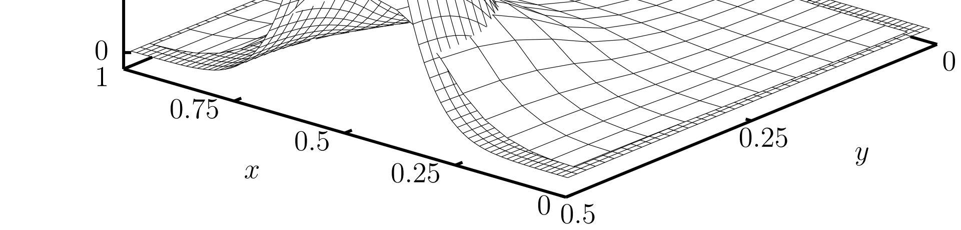


Figure 2.4: Density solutions at time $t = 10$ in the region $[0,1] \times [0,0.5]$ computed using the dimensionally-split MUSCL-Hancock method. The coarse solution uses a 20^2 grid, and the fine solution uses a 80^2 grid.

CHAPTER 3

Volume tracking methods

For representing and tracking the location of material interfaces we now consider implicit representations which track volumes of each material, primarily the volume-of-fluid (VOF) method, and the recent extension known as the moment-of-fluid (MOF) method. These methods perform better than level set approaches on the shock-interface interaction problems we are interested in, due to their inherent ability to conserve mass in spite of extreme interface deformations and topology changes.

In this chapter, three novel developments in this area are presented. The particle-VOF method increases the accuracy of the piecewise-linear reconstruction via the introduction of marker particles on the interface. The efficient-MOF method significantly improves the speed of the MOF method without affecting accuracy through the pre-computation of certain quantities. Thirdly, a novel volume fraction update is presented which permits a more accurate coupling of the VOF and level set methods. Since the same numerical tests apply to all three novel interface tracking methods, a combined results section is delivered at the end of the chapter. Before presenting these methods, we review existing approaches and describe the equations underlying the VOF method.

3.1 Governing equations

The indicator function used in VOF methods is the simplest possible implicit interface representation. This is a piecewise-constant function which takes one of two possible values throughout the domain:

$$z(\mathbf{x}, t) = \begin{cases} 1 & \mathbf{x} \in \mathcal{M} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where $\mathcal{M} \subset \mathbb{R}^d$ is the d-dimensional region containing the tracked material. From the assumption that material type is constant along streamlines, the integral form of the evolution equation of the indicator function under a velocity field $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is:

$$\frac{\partial}{\partial t}\int_{\Omega} z(\mathbf{x}, t)\ \mathrm{dV} + \int_{\partial\Omega} z(\mathbf{x}, t)\mathbf{u}\cdot\mathbf{ds} = \int_{\Omega} z(\mathbf{x}, t)\nabla\cdot\mathbf{u}\ \mathrm{dV}, \tag{3.2}$$

where Ω is some fixed control volume, and $\mathbf{ds}$ is an element on the surface of Ω. This equation can intuitively be understood as describing changes in the indicator function due to the flux of material across the boundary of the control volume, and due to compression or expansion of material in the interior of the control volume. In the discretisation of this approach, the average value of the indicator function over the computational cell $\Omega_{i,j}$ is stored, which is called the volume fraction:

$$z_{i,j} = \frac{1}{|\Omega_{i,j}|}\int_{\Omega_{i,j}} z(\mathbf{x}, t)\ \mathrm{dV}. \tag{3.3}$$

The evolution equation for the volume fractions is then:

$$|\Omega_{i,j}|\frac{\partial z_{i,j}}{\partial t} + \int_{\partial\Omega_{i,j}} z(\mathbf{x}, t)\mathbf{u}\cdot\mathbf{dl} = \int_{\Omega_{i,j}} z(\mathbf{x}, t)\nabla\cdot\mathbf{u}\ \mathrm{dV}. \tag{3.4}$$

Since $\frac{\partial z_{i,j}}{\partial t}$ depends on $z(\mathbf{x}, t)$, we require some estimate of the full indicator function. This is obtained through an explicit reconstruction of the interface in each mixed cell. Note that this advection equation is not solved using standard numerical methods, as this would smear out the interface discontinuity.

3.2 A review of volume tracking methods

In order to maintain the piecewise-constant property as it is advected by a velocity field, the indicator function in VOF methods is advanced in time using a special two-stage procedure:

1. **Reconstruction:** Explicitly reconstruct the interface (and hence the indicator function) in each mixed cell subject to a volume conservation constraint.

2. **Advection:** Advance the volume fractions to the next time step using the reconstruction from the current time step.

Each substep may be solved independently with a large variety of different schemes. In this section we survey prominent contributions to the solution of these two problems, although this list is far from exhaustive. The combination of a reconstruction scheme and an advection scheme defines a full VOF method.

3.2.1 Reconstruction schemes

Early reconstruction schemes were based on interfaces aligned with the grid [70]. Although simple to implement and cheap to run, this approach results in significant amounts of 'flotsam'– small regions which erroneously break off. The vast majority of contemporary VOF reconstruction schemes place a linear interface through each mixed cell. This interface is placed in such a way that the enclosed volume fraction of the cell is conserved, with the orientation estimated using adjacent volume fractions. The following paragraphs summarise published VOF reconstruction schemes in chronological order.

Puckett (1991) [71] The least-squares volume-of-fluid interface reconstruction algorithm (LVIRA) extends the linear interface into the surrounding 3×3 stencil of cells and minimises the error in all nine volume fractions, subject to the volume fractions being exactly equal in the central cell. The LVIRA algorithm exactly reproduces linear interfaces, and has been shown to give second order geometric accuracy. A more efficient version of LVIRA which avoids a nonlinear optimisation is presented in [72].

Parker and Youngs (1992) [73] The normal vector to the interface is estimated as the gradient of the indicator function, which is approximated by a finite difference approximation. This is a cheap and very widely used method for estimating the normal direction. However, taking finite differences of a discontinuous function is not mathematically well-defined, as discussed in [74], and the method introduces significant numerical surface tension into the solution.

Price (2000) [75] A piecewise parabolic interface was implemented with preimage-based advection, and then with edgewise advection in [76]. Despite the improved accuracy of parabolic interface representation, subsequent work has returned to piecewise-linear interface reconstruction due to the massive implementation complexity of parabolic interfaces.

Weymouth and Yue (2010) [74] A height function is constructed using the mean value theorem in the surrounding 3×3 stencil of cells. The interface normal is then estimated from finite differences of the height function. The accuracy of height function schemes is comparable to that of LVIRA, and they have the advantage of allowing for interface curvature information to be easily extracted.

3.2.2 Advection schemes

The second task in volume-of-fluid methods is calculating the volume fractions at the next time level, using the reconstructed indicator function, according to the dynamics described by equation 3.2. The update can be performed either by calculating fluxes through each cell edge, or through an equivalent cellwise update. The following paragraphs summarise significant VOF advection methods in chronological order.

Harvie and Fletcher (2001) [77] The defined donating region (DDR) scheme is an unsplit method for performing an edgewise update of the volume fractions. Under this approach the donating region (DR) of each edge is defined as a trapezium, which is non-overlapping with other DRs to conserve mass. The volume fraction fluxed across each edge is given by the intersection between the donating region, and the region containing fluid. This intersection is calculated geometrically. The DDR scheme has served as a baseline Eulerian advection method, and has been incrementally improved on by the EI-LE [78] and EMFPA [79] methods.

Weymouth and Yue (2010) [74] A simple split edgewise advection method is presented by Weymouth and Yue. As with previous methods, the flux term is calculated geometrically using a first- order accurate estimate of the donating region of each edge. In order to conserve volume in incompressible flows, the divergence term in each dimensional step is estimated using the cell-centre value of the characteristic function.

Zhang (2013) [80] Zhang takes a particularly rigorous approach to interface tracking, analysing existing methods in terms of Boolean algebra, and proposing a new advection scheme based on cubic spline intersections called DRACS (donating region approximated by cubic splines). The DRACS method is an edgewise update with fourth order accuracy, although the implementation is extremely complex.

Comminal et al. (2015) [81] The most recent VOF advection method to be proposed in the literature, called the cellwise conservative unsplit (CCU) method, this method performs a cellwise update of the volume fraction, as opposed to evaluating the flux across each edge separately. In the CCU scheme, each cell vertex is traced backwards using the fourth order Runge-Kutta method, resulting in a quadrilateral preimage. The volume fractions are updated by computing the intersection of the preimage with the material region. This method has no stability restrictions on the size of the time step. This advection method is similar to the centroid advection method used with the moment-of-fluid method in [82].

3.2.3 Alternative approaches

While the above reconstruction and advection schemes can generally be used interchangeably in a VOF method, the following methods take a different approach.

Sussman and Puckett (2000) [55] The coupled level set and VOF (CLSVOF) method uses the level set field to choose the interface normal used to reconstruct the indicator function. In turn, the reconstructed interface is used to reinitialise the level set. This hybrid scheme exhibits advantages of both methods- material conservation and easy access to geometric information such as interface normals. The principal drawback is the complexity of implementation, and the accuracy is limited by the piecewise-linear VOF interface reconstruction.

Xiao et al. (2005) [83] The THINC (tangent of hyperbola for interface capturing) scheme is a particularly novel approach in which the indicator function is assumed to be a hyperbolic tangent function. This allows the indicator function to be updated without the expensive reconstruction step normally required. This approach has been shown to work well as an interface sharpening method in multi-material flow models [34].

Dyadechko and Shashkov (2006) [82] The moment-of-fluid (MOF) method represented a significant step forward relative to existing VOF approaches. The basic idea is to track material centroids as well as volumes. The reconstruction problem can then be framed as a nonlinear optimisation problem in which the centroid error in the reconstructed indicator function is minimised. Although the MOF method is computationally expensive, it provides much better interface resolution– particularly around sharp corners and thin filaments.

Marek et al. (2008) [84] The simplified VOF (SVOF) method is a fairly recent attempt to reduce the complexity inherent in VOF methods, particularly in 3D. This approach interpolates between fluxes calculated from interfaces reconstructed parallel to a Cartesian grid, allowing for volume fractions to be evolved without the need to solve reconstruction problems. The method is simple to implement and efficient to run, but the accuracy is lower than contemporary approaches.

Jemison et al. (2013) [85] The coupled level set and MOF (CLSMOF) method presented here works similarly to the CLSVOF method [55] mentioned above, and carries similar advantages and disadvatages. In this case the level set field is used to accelerate convergence in the nonlinear optimisation step solved as part of the MOF reconstruction problem.

Kawano (2016) [86] The APPLIC (approximated piecewise linear interface calculation) method eliminates many geometric steps, including the VOF reconstruction problem, from the update procedure. Instead, simple formulae are used to solve an approximate problem. The method manages to maintain similar accuracy to PLIC methods, while running a factor of two quicker. The method is currently limited to Cartesian grids of square (or cubic) cells.

3.2.4 Remarks

The field of VOF reconstruction schemes has not moved particularly quickly over past decades. Two schemes from the early 1990's (Parker and Youngs, and LVIRA) are still standard approaches. Indeed, the maximum order of accuracy possible for piecewise-linear interfaces is two, and there are a range of different reconstruction methods which achieve this. It was noted earlier in this report that a couple of VOF schemes have been presented which use piecewise quadratic interfaces [75, 76]. However, these schemes are extremely complex to implement, and as a result they have not seen much uptake from other researchers. In general, the practical considerations of implementing geometric intersection routines seems to be the largest factor stopping the field from moving beyond piecewise–linear interface representations. This explains the recent push towards purely algebraic updates which eliminate the geometric calculations from the algorithm, such as THINC and APPLIC. There also appears to be a trend towards using Lagrangian preimage schemes for the advection substep in recent publications [81, 87] rather than the edgewise flux-based schemes developed in the early 2000s [77–79]. The cellwise approach has a number of advantages: it is more straightforward to achieve fluid conservation (just ensure that the cell preimages form a tesselation), and the update involves fewer costly polygon intersection computations.

A significant number of recent developments in interface tracking have come in the form of hybrid methods which aim to combine the best features of different algorithms, usually at the cost of simplicity. These methods tend to suffer from significant implementation complexity, but are good at rectifying particular flaws in a given approach.

Overall, a roughly equal amount of effort is being dedicated to improving the accuracy (e.g. MOF) as to decreasing the complexity (e.g. SVOF) of VOF methods. A recent comparison in [88] found these simplified methods to be suprisingly effective, given that the volume fraction flux calculation can be implemented in $\mathcal{O}(100)$ lines of FORTRAN code (in 3D a VOF reconstruction implementation will run to many thousands of lines). Nevertheless, conventional piecewise-linear VOF methods still exhibit superior accuracy. A further point to bear in mind is that when used in the context of a fully-coupled multi-material simulation, the computational cost may be dominated by the flow solver. In this case a more computationally-expensive VOF method will not affect the overall simulation cost.

From this literature review we can see that the accuracy, simplicity, and computational efficiency of VOF methods all remain under active research. In the remainder of this chapter we present developments towards all three aims, with three different original numerical schemes. Results from all three schemes are presented together at the end of this chapter.

3.3 The particle-VOF method

Following several recent successful hybrid interface tracking methods we now propose a simple but effective modification to Parker and Youngs' method which improves the accuracy of the interface representation through the introduction of marker particles lying on the interface. Being a type of explicit interface representation, marker particles offer very high accuracy for small interface deformations, and have been used in several successful varieties of front-tracking method [47, 48]. We couple this approach with a VOF method in such a way that the disadvatanges inherent to explicit interface representations are nullified, and the additional implementation complexity and computational cost are small. The important advantages of the VOF method, including mass conservation, automatic handling of topology changes, and a small direct numerical domain of dependence, are all preserved in the new method.

The basic idea behind our approach is to discard and regenerate all marker particles on the reconstructed interface every single time step. The new marker particles are then advected in the velocity field across one time level, when they are used to perform the next interface reconstruction. By regenerating all marker particles every time step, no issues with topology change or marker redistribution need to be addressed, which vastly simplifies the method. The marker particles do not add significantly to the computational expense of the VOF method since particles are only placed on the reconstructed surface, rather than in a finite-width band around the surface.

Marker particles have previously been used to improve the VOF method in [89]. Under this approach the interface in each cell is described by a chain of particles, and the interface advection occurs through geometric operations on line segments between particles. Marker particles have also been used in [90], where they enable two linear interfaces to be placed per cell, providing much greater accuracy in the representation of thin filaments. The numerical method presented here takes a different approach to using marker particles, and is designed such that existing VOF implementations may be easily retro-fitted with the marker particle coupling.

3.3.1 Interface reconstruction with marker particles

The goal of the reconstruction step is to recover the exact interface geometry in each mixed cell, subject to a volume conservation constraint and the assumption of a single interface per cell. As with other contemporary VOF methods, we use a cellwise-linear approximation to the true interface. A consequence of this is that the reconstruction problem reduces to finding the normal direction to the interface– once this has been chosen the position of the interface in the cell is uniquely determined by the cell volume fraction. For the sake of simplicity the method is described in two spatial dimensions,

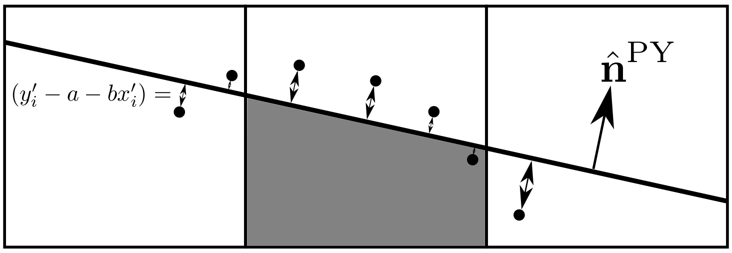


Figure 3.1: The particle-VOF reconstruction procedure viewed from the global frame. The interface normal is found through a least-squares fitting to local marker particles, performed in the frame where $\hat{\mathbf{n}}^{\mathrm{PY}}$ is aligned with the positive y-axis.

but the idea extends naturally to three dimensions.

Assume that we possess a set of marker particles, $\mathcal{S}$, which are distributed with an average density of N particles per distance Δx along the interface. The first step in the particle-VOF reconstruction process in the mixed cell (i,j) is to compute a cheap estimate for the interface normal using Parker and Youngs' method:

$$\mathbf{n}^{\mathrm{PY}} = \begin{pmatrix} \frac{-1}{8\Delta x}\left(z_{i+1,j+1} + 2z_{i+1,j} + z_{i+1,j-1} - z_{i-1,j+1} - 2z_{i-1,j} - z_{i-1,j-1}\right) \\ \frac{-1}{8\Delta y}\left(z_{i+1,j+1} + 2z_{i,j+1} + z_{i-1,j+1} - z_{i-1,j-1} - 2z_{i,j-1} - z_{i+1,j-1}\right) \end{pmatrix}. \tag{3.5}$$

This vector is normalised to obtain the unit vector $\hat{\mathbf{n}}^{\mathrm{PY}}$. Reconstructing in cell (i,j), with a cell centre located at $\mathbf{x}_{cc}$, we collect the following set of markers local to this cell:

$$\mathcal{S}_{i,j} = \{\mathbf{p} \in \mathcal{S} : \|\mathbf{p} - \mathbf{x}_{cc}\| \leq 2\Delta x\}, \tag{3.6}$$

and then limit $\mathcal{S}_{i,j}$ to the nearest $N+2$ markers to the cell centre. In the case that $\mathcal{S}_{i,j}$ is empty, simply reconstruct using $\hat{\mathbf{n}}^{\mathrm{PY}}$. This default case prevents the reconstruction process from ever failing.

Each particle in $\mathcal{S}_{i,j}$ is first transformed into the local frame. For the following linear least squares procedure to work well, the y-axis of the local frame should be normal to the interface. As such, it is reasonable to choose the y-axis of the local frame to be in the direction of $\hat{\mathbf{n}}^{\mathrm{PY}}$, with the origin on the current cell centre. Call the axial directions in this transformed frame x' and y'. A linear fit is then performed in this frame such that the squares of the vertical (y') offsets between the fitted line:

$$y' = bx' + a \tag{3.7}$$

and the y'-coordinate of each marker particle are minimised. Since we are only interested in the normal to the fitted line, there is no need to calculate the axis intercept a. The problem can be stated as follows:

$$\arg\min\ \left(R^2(a,b)\right), \tag{3.8}$$

where

$$R^2(a,b) = \sum_{i=1}^{N+2} \left(y_i' - a - bx_i'\right)^2 . \tag{3.9}$$

This has the analytic solution for b of

$$b = \frac{(N+2)\sum x_i' y_i' - \sum x_i' \sum y_i'}{(N+2)\sum x_i'^2 - \left(\sum x_i'\right)^2}. \tag{3.10}$$

This procedure is illustrated in Figure 3.1. In the transformed frame, the magnitude of b is related to how close the agreement between the Parker and Youngs normal and the marker particle normal is. If $b = 0$, they are exactly in agreement, whereas as they become perpendicular, we see that $b \to \pm\infty$. Note that the marker particle normal carries no concept of inside or outside– we rely on the Parker and Youngs normal to determine this.

Before transforming the normal of this fitted line back into the global frame, there is one important step. To improve the robustness of the normal estimation, we apply the following step to b to limit the extent to which it is allowed to differ with the Parker and Youngs normal:

$$\textbf{if } \ |b| > M \textbf{ then } b \leftarrow 0. \tag{3.11}$$

This means that if the marker particle normal makes an angle of greater than $\tan^{-1}(M)$ with $\hat{\mathbf{n}}^{\mathrm{PY}}$, it is assumed to be unreliable and discarded in favour of $\hat{\mathbf{n}}^{\mathrm{PY}}$. Finally, the interface normal in the current cell in the global frame in this cell is given by:

$$\mathbf{n}_{i,j} = \begin{pmatrix} -b \cdot \hat{n}_y^{\mathrm{PY}} + \hat{n}_x^{\mathrm{PY}} \\ b \cdot \hat{n}_x^{\mathrm{PY}} + \hat{n}_y^{\mathrm{PY}} \end{pmatrix}. \tag{3.12}$$

In the derivation of the particle-VOF interface normal, two parameters of the method have appeared. The parameter N controls the density of the marker particles. The second parameter, M, controls the extent to which the particle-VOF normal is allowed to differ to the Parker and Youngs normal. The effect of both parameters is investigated in Section 3.3.3, where optimal values are found to be $N = 9.07, \ M = 0.555$.

3.3.2 Marker re-seeding and advection

After the interface has been reconstructed as a linear function in every mixed cell, the current set of marker particles are discarded. The re-seeding procedure is simple: starting from the midpoint of the line segment which represents the interface, place marker particles every $\frac{\Delta x}{N}$ in both directions until the end points of the segment are reached. Re-seeding every marker after every time step is crucial for the simplicity and robustness of the method- this means that topology changes are incorporated automatically by the marker particles.

The particles are then advected in a velocity field $\mathbf{u}(\mathbf{x}, t)$ according to:

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{u}(\mathbf{x}_i, t), \tag{3.13}$$

where $\mathbf{x}_i$ is the position of the i-th marker, and the velocity field may be either predefined, or obtained from a flow solver. This ODE is integrated from the current time until the next time step using the RK-4 scheme. Note that the $\mathcal{O}(\Delta t^4)$ truncation error of this scheme is significantly smaller than the $\mathcal{O}(\Delta x^2)$ error associated with the approximation of the preimage as a polygon, since from the CFL criterion $\Delta t \sim \Delta x$. The volume fractions may be advected independently using any classic VOF advection scheme. We have implemented an unsplit Lagrangian preimage method, as described in [81].

3.3.3 Parameter optimisation

We have freedom in choosing values for the two parameters N and M. Both parameters must lie in the interval $[0, \infty]$, and if either parameter is equal to zero the scheme reduces to Parker and Youngs' VOF method. The computational cost of the method is independent of M, but increases with N as the number of particles increases.

In order to apply an optimisation algorithm to set the optimal value of these parameters, we first need to design a suitable objective function which reflects the performance of the method. We choose to run the single vortex test on 64^2 and 128^2 grids, and the Zalesak disk test on 100^2 and 200^2 grids. The initial conditions of both tests are fully described in section 3.6. The final error on the volume fractions from each of the four tests is normalised (so they all lie in [0,1]) and summed to give the value of our objective function $f = f(N, M)$. These two tests were chosen since they assess the performance of the method for both sharp corners and thin filaments.

The choice of optimisation algorithm is constrained by the many local minima in the objective function, and the lack of gradient information. In order to find the optimal values of these parameters, the locally biased variant of the dividing rectangles algorithm [91] for global optimisation was applied to this objective function, with the search space bounded to the region $(N, M) \in [0, 10] \times [0, 2]$. The bound of $N \leq 10$ was motivated by computational efficiency considerations. We used the implementation of this algorithm from the NLopt C++ library [60]. The algorithm converged to the optimal parameter values: $N = 9.07, \quad M = 0.555$. Note that this value of M corresponds to a maximum allowable difference between the Parker and Young's normal and particle-VOF normal of 29 degrees before the particle-VOF normal is discarded in favour of the more robust Parker and Youngs normal.

3.4 The efficient moment-of-fluid method

The introduction of the moment-of-fluid (MOF) method of Dyadechko and Shashkov [82] was a milestone in the development of indicator function-based interface tracking methods. Under this approach, both the material volume (zero-th moment) and the material centroid (first moment) are tracked, and utilised to reconstruct a piecewise-linear interface. This formulation carries two major advantages over previous VOF methods:

1. The accuracy of interface reconstruction is improved, and the method outperforms both VOF and level set methods [85]. In particular, the method is able to better evolve sharp corners and thin filaments.

2. The direct numerical domain of dependence of the MOF interface reconstruction problem is only the single cell under consideration. This makes parallelisation very straightforward.

Since the introduction of the original MOF method in 2006, significant further work has been presented. The MOF reconstruction algorithm was applied in 3D for a mesh of generalised polyhedra containing an arbitrary number of different materials in [92], where it was shown to be more accurate than various VOF reconstruction algorithms. The method has also successfully been applied in the modelling of various fluid phenomena [93, 94], including an approach designed to capture thin, under-resolved filaments [95]. Most recently, a dimensionally-split coupled level set and MOF method has been described in [85], and the accuracy and efficiency of interface reconstruction for large (≥ 3) numbers of material types has been improved in the symmetric MOF method [96].

The principal disadvantage of the moment-of-fluid method is the computational cost. Every interface reconstruction requires the solution of a nonlinear optimisation problem. Although the search space of this optimisation problem is only one-dimensional, the objective function is not trivial to evaluate, and the problem must be solved repeatedly for every separate cell reconstruction. In this section we propose a novel development to the MOF method which significantly improves the computational cost, with a negligible change in the accuracy of the method. This is achieved by making each objective function evaluation in this optimisation problem trivially cheap through an approach in which certain values are pre-computed, and then mapped on to the objective function.

3.4.1 A recap of the moment-of-fluid method

While previous VOF methods only tracked the volume fractions of the material under consideration, the MOF method also tracks the first moment of area. For an arbitrary region $R \in \mathbb{R}^2$ this is defined as:

$$\mathbf{m} = \int_R \mathbf{x} \; \mathrm{d}x_1 \mathrm{d}x_2. \tag{3.14}$$

The first moment of area is related to the material centroid $\mathbf{c}$ by

$$\mathbf{c} = \frac{\mathbf{m}}{\int_R \mathrm{d}x_1 \mathrm{d}x_2}. \tag{3.15}$$

It follows trivially that for an indicator function $z(\mathbf{x}, t)$ defined inside the cell Ω, the material centroid corresponding to the volume fraction given by Equation 3.3 is:

$$\mathbf{c} = \frac{1}{\int_\Omega z(\mathbf{x}, t) \; \mathrm{d}x_1 \mathrm{d}x_2} \int_\Omega z(\mathbf{x}, t) \mathbf{x} \; \mathrm{d}x_1 \mathrm{d}x_2. \tag{3.16}$$

The moment-of-fluid method reconstructs a piecewise-linear interface inside a cell given the volume fraction and centroid of the material under consideration inside that cell. Denoting the entire cell by Ω, let $\omega_k(\theta)$ be the region of material k which has been reconstructed inside a linear interface within Ω, where the interface normal has polar angle θ. Define the reference volume fraction and centroid for this material as z_k and $\mathbf{c}_k^{\mathrm{ref}}$. Then the moment-of-fluid interface reconstruction problem can be written as:

$$\arg\min_\theta \left\| \mathcal{T}(\omega_k(\theta)) - \mathbf{c}_k^{\mathrm{ref}} \right\|_2 \tag{3.17}$$

subject to

$$\frac{|\omega_k(\theta)|}{|\Omega|} = z_k, \tag{3.18}$$

where the function $\mathcal{T}(x)$ returns the centroid of the region $x \subset \mathbb{R}^2$.

The reference centroids are computed using a preimage-based method, as described in [82]. This approach involves computing the intersection of the Lagrangian preimage of each computational cell with the reconstructed interface geometry at the previous time step. The centroids of these intersections are then advected forwards over one time step. All advection problems are solved using the fourth-order Runge-Kutta scheme, and polygon intersections are computed using our own implementation of the Sutherland-Hodgman algorithm.

In our implementation of the original moment-of-fluid method, we apply the COBYLA (Constrained Optimization BY Linear Approximations) algorithm from the NLopt C++ library [60, 97]. This algorithm is well suited to the problem since it does not require derivatives of the objective function, and is tolerant of the gradient discontinuities in the objective function which develop as $z_k \to 0$ or $z_k \to 1$. We stop the optimisation when any one of the following termination conditions are satisfied:

- The objective function has been evaluated at least 20 times,
- The distance between the reference and reconstructed centroids falls below $10^{-10}\sqrt{\Delta x \Delta y}$. The constant $\sqrt{\Delta x \Delta y}$ is a length scale representing the size of the computational cell.

In evaluating the objective function, the linear subcell $\omega_k(\theta)$ is stored as an ordered list of vertices connected $(\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N)$ by straight lines. We make use of the formula for the centroid of a polygon:

$$\mathcal{T}(\omega_k(\theta)) = \frac{1}{6A} \sum_{i=1}^{N} (\mathbf{v}_i + \mathbf{v}_{i+1})\ \mathbf{v}_i^T\ \mathbf{R}\ \mathbf{v}_{i+1} \tag{3.19}$$

where

$$\mathbf{R} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \tag{3.20}$$

and A is the area of the polygon. The initial guess θ_0 which is provided to the optimisation routine is chosen as the polar angle of the vector between the reference centroid, and the centroid of the entire cell:

$$\theta_0 = \tan^{-1}\left(\frac{(\mathbf{c}_k^{\text{ref}} - \mathcal{T}(\Omega))_2}{(\mathbf{c}_k^{\text{ref}} - \mathcal{T}(\Omega))_1}\right). \tag{3.21}$$

We make the numerical optimisation even more robust to bad initial guesses following the procedure in [98]. If the optimisation terminates with the centroid error greater than the chosen tolerance, we increment the initial guess by $\frac{2}{3}\pi$ and repeat the optimisation. This is repeated for a total of at most three optimisations, and upon exit the value of θ which corresponds to the smallest centroid error is accepted.

3.4.2 The efficient moment-of-fluid method

The efficient moment-of-fluid method proposed here is based on the observation that the set of possible arguments for the original moment-of-fluid interface reconstruction problem (Equation 3.17) is small. Consider the moment-of-fluid reconstruction problem applied to a unit square cell. This problem accepts the following three arguments:

$$\begin{aligned} z &\in [0,1] \\ c_1^{\text{ref}} &\in [0,1] \\ c_2^{\text{ref}} &\in [0,1] \end{aligned} \tag{3.22}$$

and returns the optimal interface normal polar angle $\theta \in [0, 2\pi)$. Note we have dropped the subscript 'k' for clarity. We can describe the moment-of-fluid reconstruction by the function $\mathcal{F}_{\text{unit}}$:

$$\mathcal{F}_{\text{unit}}\colon [0,1]^3 \to [0,2\pi), \tag{3.23}$$

$$\mathcal{F}_{\text{unit}}(c_1^{\text{ref}}, c_2^{\text{ref}}, z) = \begin{cases} \arg\min_\theta \left\| \mathcal{T}(\omega(\theta)) - \mathbf{c}^{\text{ref}} \right\|_2, \\ \text{subject to } |\omega(\theta)| = z. \end{cases} \tag{3.24}$$

We now investigate how the function $\mathcal{F}_{\text{unit}}$ can be applied to more complicated cell geometries. To begin with, consider a square cell, Ω_{square}, of side Δ, with reference centroid $\mathbf{d}^{\text{ref}} \in \Omega_{\text{square}}$. Then arguments based on the scale invariance of the reconstruction problem lead to:

$$\mathcal{F}_{\text{square}}\left(d_1^{\text{ref}}, d_2^{\text{ref}}, z\right) = \mathcal{F}_{\text{unit}}\left(\frac{d_1^{\text{ref}}}{\Delta}, \frac{d_2^{\text{ref}}}{\Delta}, z\right). \tag{3.25}$$

Next consider all possible linear transformations of the unit square cell. The invertible linear transform $\mathcal{S}$ maps the cell Ω on to the unit square cell, as illustrated in Figure 3.2. Consider the effect of applying $\mathcal{S}^{-1}$ to a linear interface reconstruction inside Ω_{unit} which has an interface normal polar angle of θ, and the centroid of the region inside the interface is $\mathbf{c}$. Let α denote the interface normal polar angle following transformation of this geometry by $\mathcal{S}^{-1}$. The centroid location in the transformed space is given by

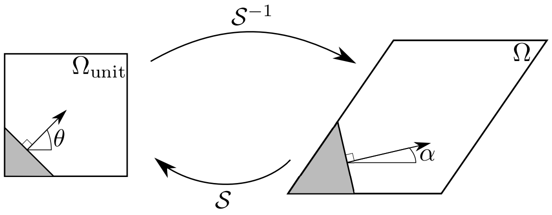


Figure 3.2: The linear transformation $\mathcal{S}$ maps the arbitrary cell Ω to the unit square cell Ω_{unit}.

$$\mathbf{d} = \mathcal{S}^{-1}\mathbf{c}. \tag{3.26}$$

Since normal vectors are transformed by the transpose of the inverse of the transformation matrix, in this case the interface normal vector will be transformed by $\mathcal{S}^{\mathrm{T}}$. This allows us to write down the relation between α and θ:

$$\begin{aligned} \alpha = \operatorname{atan2}(&\mathcal{S}_{12}\cos(\theta) + \mathcal{S}_{22}\sin(\theta), \\ &\mathcal{S}_{11}\cos(\theta) + \mathcal{S}_{21}\sin(\theta)). \end{aligned} \tag{3.27}$$

In the inverse direction, the relationship is:

$$\begin{aligned}\theta = \operatorname{atan2}\big(&\mathcal{U}_{21}\cos(\alpha) + \mathcal{U}_{22}\sin(\alpha),\\ &\mathcal{U}_{11}\cos(\alpha) + \mathcal{U}_{12}\sin(\alpha)\big)\end{aligned} \tag{3.28}$$

where $\mathcal{U} = \left(\mathcal{S}^{-1}\right)^{\mathrm{T}}$. Denote the MOF reconstruction problem in cell Ω by $\mathcal{F}_\Omega$:

$$\mathcal{F}_\Omega(\mathbf{c}^{\text{ref}}, z) = \begin{cases} \arg\min_\alpha \left\|\mathcal{T}(\omega(\alpha)) - c^{\text{ref}}\right\|_2, \\ \text{subject to } |\omega(\alpha)| = z. \end{cases} \tag{3.29}$$

Now for any given z and linear interface normal angle α, consider the position of the reconstructed material centroid $\in \Omega$. From Equation 3.15, we see that centroid position must vary smoothly as a function of θ, and for arbitrary fixed z, the position of the material centroid traces out a closed curve in $\mathbb{R}^2$ as θ varies from 0 to 2π. Call this closed curve ζ_Ω, parametrised by θ so $\zeta_\Omega(0) = \zeta_\Omega(2\pi)$. Formally, we have

$$\zeta_\Omega(\alpha) = \mathcal{T}(\omega(\alpha)). \tag{3.30}$$

Three examples of this curve for various z are shown in Figure 3.3. The reconstruction problem (Equation 3.29) can be written in terms of $\zeta_\Omega(\theta)$:

$$\mathcal{F}_\Omega\left(\mathbf{c}^{\text{ref}}, z\right) = \arg\min_\alpha \left\|\zeta_\Omega(\alpha) - \mathbf{c}^{\text{ref}}\right\|. \tag{3.31}$$

Equipped with this definition of the curve ζ_Ω, the following lemma tackles the problem of whether $\mathcal{F}_{\text{unit}}$ can be directly related to $\mathcal{F}_\Omega$ in a similar way to the square cell case.

Lemma 3.1. If the invertible linear transformation $\mathcal{S}$ maps the arbitrarily-shaped cell Ω to the unit square cell Ω_{unit} then in general:

$$\mathcal{F}_\Omega\left(\mathbf{c}^{\text{ref}}, z\right) \neq \mathcal{F}_{\text{unit}}\left(\mathcal{S}\mathbf{c}^{\text{ref}}, z\right), \tag{3.32}$$

where $\mathcal{F}_\Omega \colon \Omega \times [0,1] \to [0, 2\pi)$ solves the moment-of-fluid reconstruction problem in cell Ω.

Proof. With reference to Equation 3.30, denote the corresponding closed curve on the unit square cell by $\mathcal{S}\zeta_{\Omega_{\text{unit}}}$. Since centroid positions are also transformed by $\mathcal{S}$, the two curves can be directly related through:

$$\zeta_{\Omega_{\text{unit}}}(\alpha) = \mathcal{S}\zeta_\Omega(\theta), \tag{3.33}$$

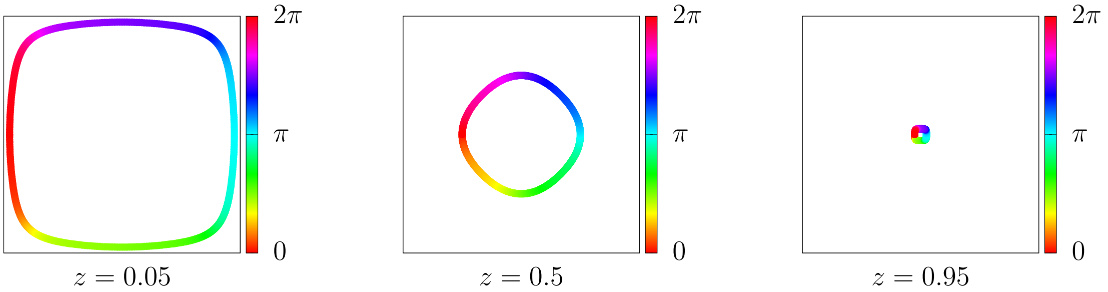


Figure 3.3: Three examples of the closed curve $\zeta_{\Omega_{\text{unit}}}(\theta)$ inside the unit square cell for various z. The colour denotes the value of θ yielding the corresponding centroid position.

where the relations between θ and α are given by Equations 3.28 and 3.27. Therefore

$$\mathcal{F}_{\text{unit}}\left(\mathcal{S}\mathbf{c}^{\text{ref}}, z\right) = \arg\min_{\theta} \left\| \mathcal{S}(\zeta_{\Omega}(\theta) - \mathbf{c}^{\text{ref}}) \right\|, \tag{3.34}$$

and unless the linear transformation $\mathcal{S}$ has all eigenvalues equal (i.e. it is an isotropic scaling, a multiple of the identity matrix) this is not equal to Equation 3.31. □

Define the surface Z_{Ω} as the union of all curves ζ_{Ω} with $z \in [0, 1]$. Any point on Z_{Ω} simply corresponds to the centroid position of the linear reconstruction inside Ω for some z and α. In this framework the reconstruction problem can be solved for a given z and reference centroid by finding the point on Z_{Ω} on the given z-isosurface closest to the reference centroid.

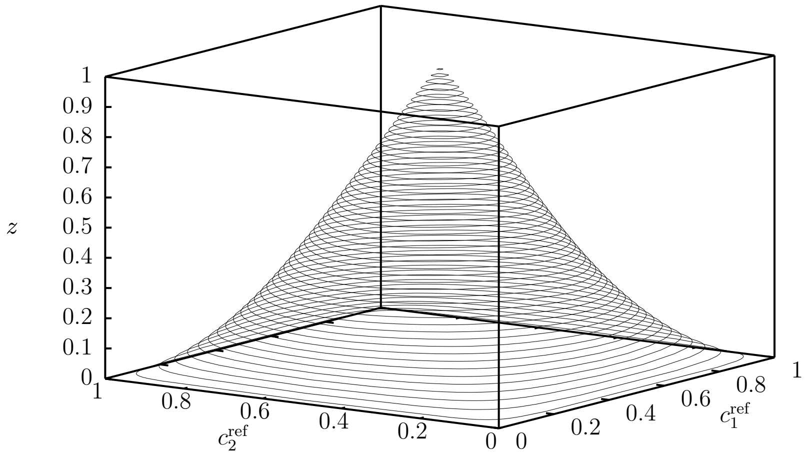


Figure 3.4: The curves $\zeta_{\Omega_{\text{unit}}}$ for 50 different samples of z. The surface $Z_{\Omega_{\text{unit}}}$ consists of the union of these curves for $z \in [0, 1]$.

The efficient MOF method is based on pre-computing the surface Z for a base cell geometry, and relating this to the reconstruction problem inside any cell which can be reached through a linear transformation of the base cell. For instance, applying the pre-computation to the unit triangle allows for reconstruction on any triangular cell, and the pre-computation for the unit square allows for reconstruction on any rectangular cell.

3.4.3 The efficient moment-of-fluid algorithm

Consider the following algorithm for solving the reconstruction problem in the cell Ω with volume fraction z, and reference centroid $\mathbf{d}^{\text{ref}}$. The cell Ω results from a linear transformation of some base cell, which is chosen to be the unit square cell Ω_{unit} in this case: $\Omega = \mathcal{S}^{-1}\Omega_{\text{unit}}$. In this algorithm, it is assumed that the surface $Z_{\Omega_{\text{unit}}}$ has been pre-computed and stored.

1. Choose the initial guess for the interface normal polar angle, α, in the usual way.

2. Solve the minimisation problem

$$\arg\min_{\alpha} \left\| \zeta_{\Omega}(\alpha) - \mathbf{d}^{\text{ref}} \right\| \tag{3.35}$$

   using the following procedure to rapidly evaluate $\zeta_{\Omega}(\alpha)$:

   (a) Compute θ using Equation 3.27.

   (b) Look up an approximation to the value of $\zeta_{\Omega_{\text{unit}}}(\theta)$ from a pre-computed data structure.

   (c) Return $\mathcal{S}^{-1}\zeta_{\Omega_{\text{unit}}}(\theta)$.

This approach does not eliminate the costly nonlinear optimisation from the interface reconstruction; instead an evaluation of the objective function in the optimisation is made trivially cheap. Data is only stored on the two-dimensional surface shown in Figure 3.4, which will be shown later to be smooth enough for interpolation schemes to work effectively. This approach is applicable on any grid in which all cell shapes can be expressed as a linear transformation of one base shape. This covers all rectilinear and unstructured triangular grids, but not curvilinear grids.

This also explains why we limit our consideration to cell shapes which can be reached by a linear transformation from the base case – otherwise the new interface normal polar angle may not be uniquely defined. Note that the volume fraction is unchanged, since both the area of the cell and the area of the reconstructed region are modified by a factor of $\det(\mathcal{S})$.

Implementation remark 1 The centroid position vector is stored on a two-dimensional grid which discretises z and θ throughout $[0, 1] \times [0, 2\pi)$. This allows for $\mathcal{O}(1)$ retrieval of $\zeta_{\Omega_{\text{unit}}}(\theta)$. Each of z and θ are discretised at N evenly distributed points. The x- and y-components of the centroid in each grid cell are both stored as 4 byte (single precision) floating point numbers. By storing all N^2 grid points in one contiguous block of memory, it is possible to write this pre-computed data structure to a file using exactly $8N^2$ bytes of memory.

Implementation remark 2 Generating the pre-computed approximation to $\zeta_{\Omega_{\text{unit}}}(\theta)$ is very fast. Computing the centroid location in the unit square cell given the normal angle and volume fraction requires reconstructing the given linear subcell and then computing the centroid of this polygon.

3.4.3.1 A note on the naive approach

It is possible to take a naive approach towards improving the efficiency of the MOF reconstruction problem by simply pre-computing the value of $\mathcal{F}_{\text{unit}}$ throughout its domain (the unit cube), and applying Equation 3.25 to use these function values to solve the reconstruction problem in any cell which is an isotropic scaling of the unit square cell. This approach is exceptionally fast, since solving the interface reconstruction problem reduces to just looking up a value in a data structure. Unforunately, every time that a simulation is run with a new cell shape, the entire look-up table must be regenerated. For large simulations in which every cell is the same shape, it is still advantageous to regenerate the look-up table once at the beginning of the simulation, but if a grid is used which features many different cell geometries, this approach is not applicable. Furthermore, in our numerical experiments the function $\mathcal{F}_{\text{unit}}$ has not proven smooth enough to be well-approximated by grid interpolation schemes.

3.4.4 Objective function approximation error

The efficient MOF (denoted as EMOF in this section) method works by making an evaluation of the objective function in the reconstruction problem trivially cheap. We now investigate the error associated with this approximation by measuring

$$\int_0^{2\pi} |f_{\text{exact}}(\theta) - f_{\text{EMOF}}(\theta)| \, \mathrm{d}\theta, \tag{3.36}$$

where $f_{\text{exact}}(\theta)$ denotes the true objective function used in moment of fluid reconstruction, and $f_{\text{EMOF}}(\theta)$ denotes the approximated objective function. This integral is estimated using a Monte Carlo approach:

$$E_f = \frac{2\pi}{N} \sum_{i=1}^{N} |f_{\text{exact}}(\theta_i) - f_{\text{EMOF}}(\theta_i)|, \tag{3.37}$$

where $\theta_i \sim \text{unif}(0, 2\pi)$ and $N = 10^6$. The setup for the reconstruction problem is shown in Figure 3.5, and is designed so that the exact objective function will have a value of zero at $\theta = \pi$. The error was measured for 7 logarithmically-spaced pre-computation grid sizes, from 200^2 to $12,800^2$. These data structures occupied from 320 kB to 1.3 GB of disk space.

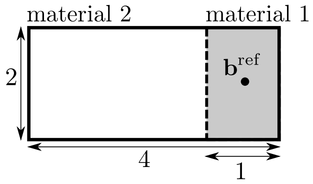


Figure 3.5: The initial set up of the objective function approximation test for the EMOF algorithm. The origin is set at the bottom-left corner of the cell, and the reference material centroid $\mathbf{b}^{\text{ref}}$ is located at $(3.5, 1)$.

Accuracy for two interpolation approaches are displayed in Figure 3.6. The expected convergence order is achieved for both interpolation methods, which reflects the fact that the surface Z_Ω is smooth. Using the finest grid, the objective function error is approaching the value of machine epsilon for a 32-bit floating point number, and this requires storing a 1.3 GB look-up table in memory– a quantity which is easily achieved on modern machines. The efficient MOF approach is able to speed up the evaluation of the objective function in the MOF reconstruction problem by approximately a factor of 10.

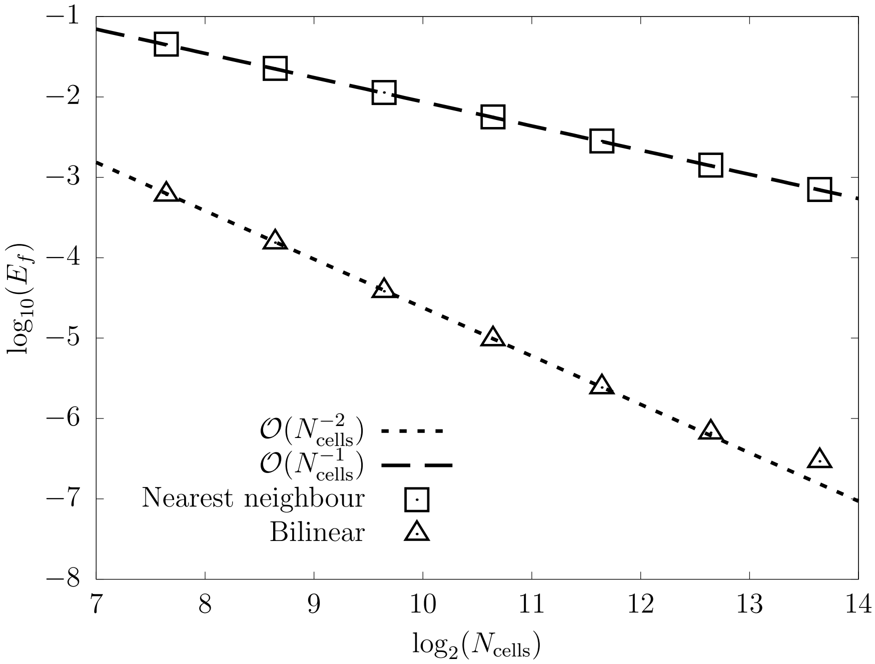


Figure 3.6: Comparison of error on the approximation to the MOF objective function for various sizes of the pre-computed grid and the interpolation strategy in the efficient MOF method. N_{cells} refers to the number of grid points which z and θ are each discretised over. Good agreement with the theoretical error convergence rate for a smooth function is obtained.

3.5 Reconstruction-free VOF

Almost all existing VOF methods follow an identical reconstruction-advection sequence, in which the interface is explicitly located in each mixed cell (subject to a volume conservation constraint) prior to advection of the indicator function over one time step. Prominent reconstruction and advection methods are reviewed in section 3.2. The process of reconstructing the interface in each mixed cell at each time step adds significant computational expense to the VOF method, particularly reconstruction methods which involve a nonlinear optimisation solve. The approximations involved in reconstruction also limit the accuracy of the VOF method – almost all current reconstruction schemes assume a piecewise-linear interface.

This section describes a novel approach to evolving the indicator function in a VOF method which eliminates the reconstruction step, and opens the possibility of coupling the VOF method to other interface tracking methodologies. The approach proceeds by describing the volume fraction update in terms of a quadratic programming problem, and a general solution to this system is derived using the active set method. In section 3.5.1 the mathematical framework is described, and then in section 3.5.2 one possible application of this framework is developed in a novel coupling of the level set and VOF methods.

3.5.1 A constrained optimisation framework for evolving volume fractions

In a nutshell, the framework is based on a set of physical constraints on the volume fraction update which, if satisfied, guarantee that the interface tracking scheme conserves the mass of each material while keeping all volume fractions appropriately bounded. Given some (possibly non-physical) initial guess for the volume fraction update, the framework returns the closest possible update which obeys the constraints. The source of the initial guess is left unspecified, so this framework opens an alternative approach towards combining volume tracking with other interface tracking methodologies.

3.5.1.1 Physical constraints on a general volume fraction update

To begin with we write down analytic formulae for the volume fraction update using the idea of the Lagrangian preimage [99]. If the area occupied by the tracked material at time t^n is denoted by $\mathcal{M}(t^n)$, where $\mathcal{M}(t^n) \subset \mathbb{R}^2$, then equation 3.3 can be re-written as

$$z_{i,j}^n = \frac{|\mathcal{M}(t^n) \cap \Omega_{i,j}|}{|\Omega_{i,j}|}. \tag{3.38}$$

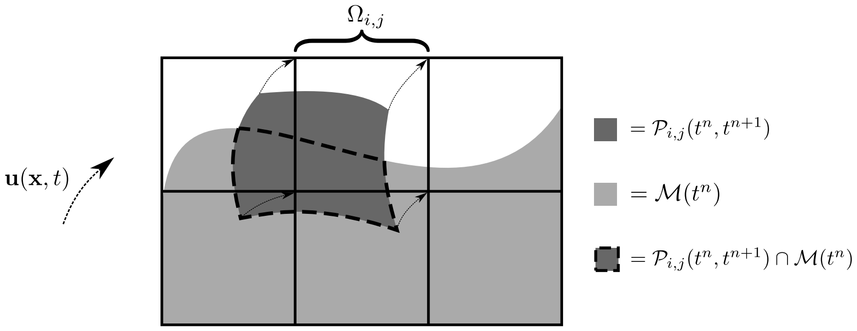

$\Omega_{i,j}$
$\mathbf{u}(\mathbf{x},t)$
$= \mathcal{P}_{i,j}(t^n, t^{n+1})$
$= \mathcal{M}(t^n)$
$= \mathcal{P}_{i,j}(t^n, t^{n+1}) \cap \mathcal{M}(t^n)$

Figure 3.7: An illustration of the Lagrangian preimage of cell $\Omega_{i,j}$. The volume fractions at time t^{n+1} can be found using the intersection of the preimage with the region occupied by the tracked material at time t^n as described by equation 3.40.

The Lagrangian preimage of $\Omega_{i,j}$ (denoted by $\mathcal{P}_{i,j}(t^n, t^{n+1}) \subset \mathbb{R}^2$) is the set of points which are advected into $\Omega_{i,j}$ by the velocity field $\mathbf{u}(\mathbf{x}, t)$ over the time interval $[t^n, t^{n+1}]$:

$$\mathcal{P}_{i,j}(t^n, t^{n+1}) = \left\{ \mathbf{p}(t) \in \mathbb{R}^2 \;\middle|\; \mathbf{p}(t^n) + \int_{t^n}^{t^{n+1}} \mathbf{u}(\mathbf{p}(t), t)\ \mathrm{d}t \ \in \ \Omega_{i,j} \right\}. \tag{3.39}$$

This is illustrated in figure 3.7. There are now two possible solutions for the volume fractions at time t^{n+1} depending on the divergence of the velocity field:

1. If $\nabla \cdot \mathbf{u} = 0$
$$z_{i,j}^{n+1} = \frac{|\mathcal{P}_{i,j}(t^n, t^{n+1}) \cap \mathcal{M}(t^n)|}{|\mathcal{P}_{i,j}(t^n, t^{n+1})|}, \tag{3.40}$$
because the area of the preimages are invariant with respect to time.

2. If $\nabla \cdot \mathbf{u} \neq 0$, the solution is more complicated because it is necessary to account for the non-uniform compression or stretching which may occur inside the preimage:
$$z_{i,j}^{n+1} = \frac{|\mathcal{R}_{i,j}(t^n)| + \int_{t^n}^{t^{n+1}} \left(\int_{\partial \mathcal{R}_{i,j}(t)} \mathbf{u} \cdot \mathbf{dl} \right) \ \mathrm{d}t}{|\Omega_{i,j}|}, \tag{3.41}$$
where the region $\mathcal{R}_{i,j}(t)$ is defined as the intersection of the preimage and the tracked material at time $t < t^{n+1}$:
$$\mathcal{R}_{i,j}(t) = \mathcal{P}_{i,j}(t, t^{n+1}) \cap \mathcal{M}(t). \tag{3.42}$$
In the case of a divergence-free velocity field, (3.41) reduces to (3.40), since the surface integral over $\partial\mathcal{R}_{i,j}(t)$ can be shown to be zero using the divergence theorem. This form of the volume fraction update could also be applied to the case of non-uniform compression or stretching of materials with different bulk moduli, although this is not considered in the current chapter.

The unknown quantity in these formulae is of course the region $\mathcal{M}(t)$. The following sections are devoted to the development of a scheme for estimating $|\mathcal{P}_{i,j}(t^n, t^{n+1}) \cap \mathcal{M}(t^n)|$ such that certain constraints on $z_{i,j}^{n+1}$ are obeyed.

There is relatively little mention in the literature of the extension of VOF advection schemes to the nonzero divergence case, since many schemes were designed with incompressible flow applications in mind [55, 85, 100, 101]. In practice, the solution given by equation 3.40 has been applied to the nonzero divergence case, for example in the dimensionally-split advection scheme in [87]. This approximation is equivalent to making the assumption of uniform compression or expansion over the preimage, which degrades accuracy somewhat but does not affect the conservation properties of the scheme. Our numerical scheme is also based on the solution (3.40), with the extension to (3.41) anticipated in future work.

In deriving conservative solutions for equation 3.40, two definitions are used extensively. As illustrated in Figures 3.8 and 3.9, consider a single grid cell with the preimages of all local cells overlain. The region $\mathcal{G}_i \subset \mathbb{R}^2$ is defined by the intersection of the grid cell Ω, and the i-th local preimage $\mathcal{P}_i$:

$$\mathcal{G}_i = \Omega \cap \mathcal{P}_i. \tag{3.43}$$

Since each cell is completely covered by preimages, $\bigcup_i \mathcal{G}_i = \Omega$. The volume of tracked material inside each $\mathcal{G}_i$ is denoted by $a_i \in \mathbb{R}$. The value of a_i gives the quantity of tracked material which is moving from the current cell, into the cell from which the preimage $\mathcal{P}_i$ originated:

$$a_i = |\mathcal{G}_i \cap \mathcal{M}(t^n)|\,. \tag{3.44}$$

Two physical requirements may now be introduced. Firstly, the volume fraction update should not introduce any spurious change in the quantity of tracked material. Secondly, the volume fractions should remain bounded in the interval $[0, 1]$. Thus, the following two constraints on the a_i used for the volume fraction update are derived:

1. The total volume of material leaving (or being redistributed within) the cell (the sum of the a_i) must equal the total volume of fluid currently in the cell which is known from the cell volume fraction, and defined equal to b:

   $$\sum_i a_i = \Delta x \Delta y z_{i,j} \equiv b. \tag{3.45}$$

   Note that from $z_{i,j} \in [0, 1]$ and $\sum_i |\mathcal{G}_i| = \Delta x \Delta y$ we have

   $$0 \le b \le \sum_i |\mathcal{G}_i|\,. \tag{3.46}$$

   This will prove useful later when we come to proving consistency of the constraints.

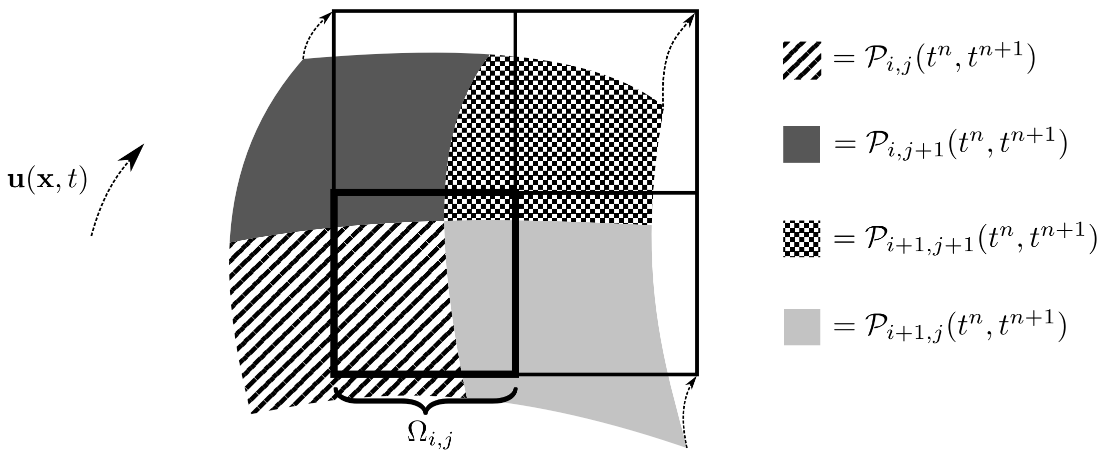

$\mathbf{u}(\mathbf{x},t)$
$\Omega_{i,j}$
▨ $= \mathcal{P}_{i,j}(t^n, t^{n+1})$
■ $= \mathcal{P}_{i,j+1}(t^n, t^{n+1})$
▩ $= \mathcal{P}_{i+1,j+1}(t^n, t^{n+1})$
□ $= \mathcal{P}_{i+1,j}(t^n, t^{n+1})$

Figure 3.8: An illustration of the regions $\mathcal{G}_i$ which are useful in the derivation of the physical constraints. For the cell under consideration ($\Omega_{i,j}$), there are four nonzero regions $\mathcal{G}_i$ as defined by equation 3.43.

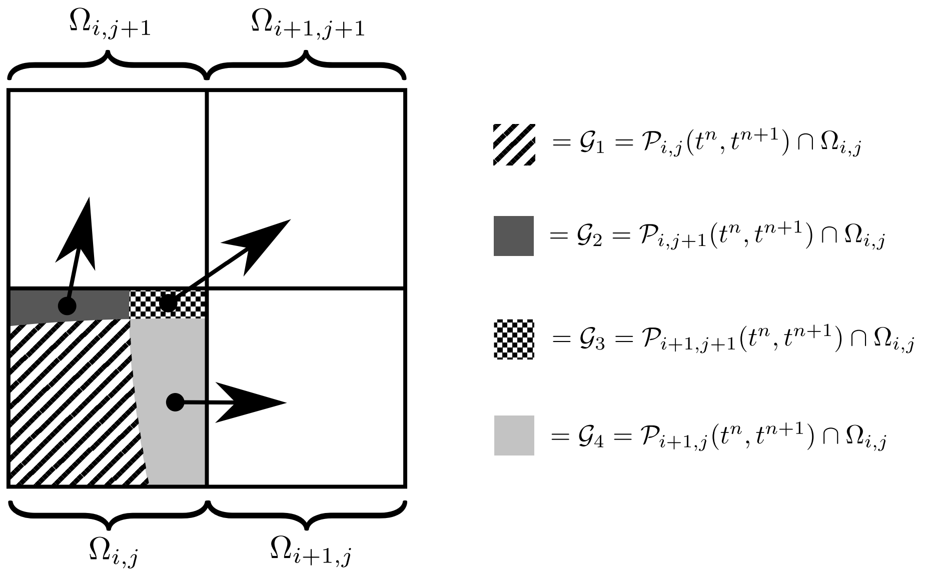

$\Omega_{i,j+1}$
$\Omega_{i+1,j+1}$
$\Omega_{i,j}$
$\Omega_{i+1,j}$
▨ $= \mathcal{G}_1 = \mathcal{P}_{i,j}(t^n, t^{n+1}) \cap \Omega_{i,j}$
■ $= \mathcal{G}_2 = \mathcal{P}_{i,j+1}(t^n, t^{n+1}) \cap \Omega_{i,j}$
▩ $= \mathcal{G}_3 = \mathcal{P}_{i+1,j+1}(t^n, t^{n+1}) \cap \Omega_{i,j}$
□ $= \mathcal{G}_4 = \mathcal{P}_{i+1,j}(t^n, t^{n+1}) \cap \Omega_{i,j}$

Figure 3.9: The intersections between $\Omega_{i,j}$ and the four preimages overlapping $\Omega_{i,j}$ are indicated by the four coloured regions. See Figure 3.8 for an illustration of the full preimage areas. Arrows indicate the cell in which each $\mathcal{G}_i$ ends up at the end of the time step at t^{n+1}. The constraints (3.45) and (3.47) are derived by considering the limits on the amount of tracked material which can fit inside each $\mathcal{G}_i$.

2. The quantity of material a_i leaving each region $\mathcal{G}_i$ cannot be negative, or greater than the size of that region:
$$0 \leq a_i \leq |\mathcal{G}_i|. \tag{3.47}$$

If the constraints on a_i are satisfied, the resulting volume fraction update is guaranteed to be bounded in [0,1]. These constraints must be applied separately to every single mixed cell. This procedure is distinct to all previous VOF schemes which achieve material conservation by explicitly reconstructing the interface within each mixed cell, and then computing a_i geometrically.

3.5.1.2 Formulation as a constrained optimisation problem

Assume that a set of initial guesses for a_i, the quantity of material moving from the current cell over to cell i this time step, are given – there is no restriction on how the initial guesses are obtained, and a method will be proposed later. We then search for the closest possible set of a_i which obey the constraints defined in the previous section. Let us denote the initial guess for a_i by k_i. We also define $\mathbf{a}$ and $\mathbf{k}$ as the vectors whose components are each of the a_i and k_i. The dimension of these vectors is then equal to the number of distinct computational cells (including itself) which material is moving into from the current cell over the current time step. We denote this number by M, so both $\mathbf{a}$ and $\mathbf{k}$ are vectors in M-space. Assuming a CFL restriction of $C_{\text{CFL}} \leq 1$, in a d-dimensional simulation we have $0 \leq M \leq 3^d$. The problem can now be stated as follows :

$$\begin{cases} \min_{\mathbf{a}} \|\mathbf{a} - \mathbf{k}\|^2, \\ \sum\limits_i a_i = b, \\ 0 \leq a_i \leq |\mathcal{G}_i| \ \forall\ i. \end{cases} \tag{3.48}$$

It is useful to think of this minimisation problem in a geometric sense. We are searching for the point $\mathbf{a}$ which minimises the Euclidean distance in M-space to $\mathbf{k}$, subject to the two constraints. The equality constraint is equivalent to searching for $\mathbf{a}$ on the hyperplane $\mathcal{H}$:

$$\mathcal{H} = \left\{ \mathbf{a} \in \mathbb{R}^M \;\middle|\; \mathbf{c}^{\mathrm{T}}\mathbf{a} = b \right\}, \quad \mathbf{c} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{3.49}$$

The inequality constraints are equivalent to searching for $\mathbf{a}$ inside the orthotope (M-dimensional generalisation of the rectangle) $\mathcal{R}$:

$$\mathcal{R} = \left\{ \mathbf{a} \in \mathbb{R}^M \;\middle|\; 0 \leq a_i \leq |\mathcal{G}_i|,\ i = 1, 2, \ldots, M \right\}. \tag{3.50}$$

Problem consistency: The existence of a solution to the constrained optimisation problem is now demonstrated by showing that the constraints are consistent with each other. In geometric terms it must be shown that the hyperplane $\mathcal{H}$ intersects the orthotope $\mathcal{R}$. To begin with, note that the vertex of $\mathcal{R}$ which is furthest from the origin lies at the position $\mathbf{g}$, where $g_i = |\mathcal{G}_i|$ for $i = 1, 2, \ldots, M$. From the definition of $\mathcal{H}$, if $\mathbf{g}$ lies within the half space defined by $\mathbf{c}^{\mathrm{T}}\mathbf{a} \geq b$ then $\mathcal{H}$ must intersect $\mathcal{R}$. The inner product of the normal vector to $\mathcal{H}$ with $\mathbf{g}$ is

$$\mathbf{c}^{\mathrm{T}}\mathbf{g} = \sum_{i=1}^{M} |\mathcal{G}_i| \,, \tag{3.51}$$

and by comparison to equation 3.46 which assumes only that the volume fractions lie in $[0, 1]$:

$$\mathbf{c}^{\mathrm{T}}\mathbf{g} \geq b. \tag{3.52}$$

Therefore $\mathbf{g}$ is indeed contained in this half space.

3.5.1.3 Quadratic programming solution

Since the objective function is convex quadratic, and we have linear equality and inequality constraints, this is a quadratic programming problem (for which a unique solution exists). A number of algorithms exist for the solution of these problems. Due to the small number of dimensions in our search space, and the fact that the feasible region is convex, the active set method is particularly suited. Written in standard form, our problem is:

$$\min_{\mathbf{a}} \; (\mathbf{a} - \mathbf{k})^{\mathrm{T}}(\mathbf{a} - \mathbf{k}), \tag{3.53}$$

subject to

$$\mathbf{h}_i^T \mathbf{a} = b_i \quad i \in \mathcal{E}, \tag{3.54}$$

$$\mathbf{h}_i^T \mathbf{a} \geq b_i \quad i \in \mathcal{I}. \tag{3.55}$$

Under this notation, $\mathcal{E}$ is the set of indices of equality constraints, and $\mathcal{I}$ is the set of indices of inequality constraints:

$$\mathcal{E} = \{0\}, \tag{3.56}$$

$$\mathcal{I} = \{1, 2, ..., 2M\}. \tag{3.57}$$

Furthermore, the constraint vectors and values are:

$$\mathbf{h}_i = \begin{cases} (1\ 1\ ...\ 1)^T & i = 0 \\ \hat{\mathbf{e}}_i & 1 \leq i \leq M \\ -\hat{\mathbf{e}}_{i-M} & M+1 \leq i \leq 2M, \end{cases} \tag{3.58}$$

$$b_i = \begin{cases} b & \text{if } i = 0 \\ 0 & \text{if } 1 \leq i \leq M \\ -|\mathcal{G}_{i-M}| & \text{if } M+1 \leq i \leq 2M. \end{cases} \tag{3.59}$$

The application of the active set method to this problem is summarised in algorithm 1, and further details can be found in [102]. The active set method iterates towards the optimal solution by solving intermediate equality-constrained sub-problems, in which some set of the constraints are imposed as equalities. This set of constraints is called the working set, $\mathcal{W}$, and consists of all the equality constraints plus some of the inequality constraints. For this particular case, it is possible to derive analytic solutions for the equality-constrained sub-problem, as well as for the initial feasible point.

Initial feasible point This is required to start the active set method. A feasible point is guaranteed to be located at the intersection of the hyperplane $(1\ 1\ ...\ 1)\mathbf{a} = b$ and the line $\mathbf{a} = k(|\mathcal{G}_1|\ |\mathcal{G}_2|\ ...|\mathcal{G}_M|)^T$ for some $k \in \mathbb{R}$. This is easily shown to be

$$\mathbf{a}_0 = \frac{b}{\sum_{i=1}^{M} |\mathcal{G}_i|}(|\mathcal{G}_1|\ |\mathcal{G}_2|\ ...|\mathcal{G}_M|)^T. \tag{3.60}$$

Equality-constrained sub-problem It is possible to derive an analytic solution for the equality-constrained sub-problem. The solution of this sub-problem is used to find the next step in the active set method, where the current optimal solution is $\mathbf{a}_c$. The sub-problem can be stated as:

$$\min_{\mathbf{p}} \frac{1}{2}\mathbf{p}^T\mathbf{p} + (\mathbf{a}_c - \mathbf{k})^T\mathbf{p} \tag{3.61}$$

subject to

$$\mathbf{h}_i^T\mathbf{p} = 0 \quad i \in \mathcal{W}_c, \tag{3.62}$$

where $\mathcal{W}_c \subset \{0, 1, ..., 2M\}$. The Lagrangian of this system is

$$\mathcal{L}(\mathbf{p}, \lambda) = \frac{1}{2}\mathbf{p}^T\mathbf{p} + (\mathbf{a}_c - \mathbf{k})^T\mathbf{p} - \lambda_0\mathbf{h}_0^T\mathbf{p} - \sum_{i \in \mathcal{W}_c \cap \mathcal{I}} \lambda_i\mathbf{e}_i^T\mathbf{p}. \tag{3.63}$$

Setting the gradient of the Lagrangian equal to zero gives:

$$0 = \mathbf{p} + (\mathbf{a}_c - \mathbf{k}) - \lambda_0 - \sum_{i \in \mathcal{W}_c \cap \mathcal{I}} \lambda_i \mathbf{e}_i, \tag{3.64}$$

and summing each component of this equation results in:

$$0 = \sum_{j=1}^{M} ((a_c)_j - k_j) - M\lambda_0 - \sum_{i \in \mathcal{W}_c \cap \mathcal{I}} \lambda_i. \tag{3.65}$$

At this point we define the sets $\mathcal{B}$ and $\mathcal{T}$. We have $i \in \mathcal{B}$ if $0 < i \leq M$ and $i \in \mathcal{W}_c$, while $i \in \mathcal{T}$ if $i + M \in \mathcal{W}_c$. Then clearly $\mathcal{B} \cap \mathcal{T} = \emptyset$, and for $i \in \mathcal{B} \cup \mathcal{T}$ we know that $p_i = 0$. Therefore, from equation 3.64, for any component i which also lies in $\mathcal{B} \cup \mathcal{T}$, we obtain

$$\lambda_i = (a_c)_i - k_i - \lambda_0. \tag{3.66}$$

Using this notation, equation 3.65 becomes

$$0 = \sum_{j=1}^{M} ((a_c)_j - k_j) - M\lambda_0 - \sum_{i \in \mathcal{B} \cup \mathcal{T}} ((a_c)_i - k_i - \lambda_0). \tag{3.67}$$

Solving for λ_0 gives

$$\lambda_0 = \frac{\sum_{i \notin \mathcal{B} \cup \mathcal{T}} ((a_c)_i - k_i)}{M - |\mathcal{B} \cup \mathcal{T}|}, \tag{3.68}$$

which makes it possible to write down the full solution for $\mathbf{p}$:

$$p_i = \begin{cases} 0 & i \in \mathcal{B} \cup \mathcal{T} \\ k_i - (a_c)_i + \frac{\sum_{i \notin \mathcal{B} \cup \mathcal{T}} ((a_c)_i - k_i)}{M - |\mathcal{B} \cup \mathcal{T}|} & i \notin \mathcal{B} \cup \mathcal{T} \end{cases}. \tag{3.69}$$

3.5.1.4 Implementation

The following steps summarize the conservative update of a grid of volume fractions without solving the VOF reconstruction problem:

1. Estimate the Lagrangian preimage of all cells in a band around the interface. We solve this step by tracing backwards along streamlines from each cell vertex and approximating the preimage as a polygon.

2. In each cell close to the interface, compute the intersection with all surrounding preimages using the Sutherland-Hodgman algorithm.

3. Using some as-yet unspecified method, estimate the amount of fluid inside each cell-preimage intersection ($\mathbf{k}$).

Algorithm 1: The active set method for the solution of the convex quadratic programming problem specified by equations 3.53 and 3.54 [102].

```
Compute feasible initial point $\mathbf{a}_0$ according to equation 3.60;
Set $\mathcal{W}_0$ to be a subset of the active constraints at $\mathbf{a}_0$;
for $c = 0, 1, 2, ...$ do
    Solve equality-constrained problem with solution given by equation 3.69;
    if $\boldsymbol{p}_c = 0$ then
        Compute lagrange multipliers $\hat{\lambda}_i$ as the solution of the linear system
          $\sum_{i \in \mathcal{W}_c} \mathbf{h}_i \hat{\lambda}_i = \mathbf{a}_c - \mathbf{k}$;
        if $\hat{\lambda}_i \geq 0,\ \forall\ i \in \mathcal{W}_c \cap \mathcal{I}$ then
            return $\boldsymbol{a}_c$
        else
            $j \leftarrow \arg\min_{j \in \mathcal{W}_c \cap \mathcal{I}} \hat{\lambda}_j$;
            $\mathbf{a}_{k+1} \leftarrow \mathbf{a}_c$;
            $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_c \setminus \{j\}$;
        end
    else
        Compute: $\alpha_c = \min\left(1, \min_{i \notin \mathcal{W}_c,\ \mathbf{h}_i^T \mathbf{p}_c < 0} \frac{b_i - \mathbf{h}_i^T \mathbf{a}_c}{\mathbf{h}_i^T \mathbf{p}_c}\right)$;
        $\mathbf{a}_{k+1} \leftarrow \mathbf{a}_c + \alpha_c \mathbf{p}_c$;
        if There are blocking constraints then
            Obtain $\mathcal{W}_{k+1}$ by adding one blocking constraint to the working set
        else
            $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k$
        end
    end
end
```

4. Set up and solve the quadratic programming problem in each cell as described by algorithm 1.

5. Now update volume fractions by summing the area of material inside each cell-preimage intersection.

3.5.2 A novel coupling of the VOF and level set methods

In the volume fraction update described in the preceeding section the source of the initial estimate $\mathbf{k}$ in Equation 3.53 was deliberately left open. In this section one possible method for estimating $\mathbf{k}$ directly from a level set field is described. This two dimensional scheme stores two variables: a finite volume grid of cell volume fractions, and a finite difference grid of level set function values. These variables are advanced through time concurrently. The method has little in common with previous hybrid level-set-VOF methods based on the CLSVOF method of Sussman and Puckett [55] since the VOF reconstruction problem is completely eliminated from the algorithm.

As will be described later, there is still a serious shortcoming in this method. However, it is a useful demonstration of the potential of the novel update framework developed in the preceeding section.

3.5.2.1 Updating volume fractions from a level set field

As described by equation 3.40, the estimated area of material inside each region $\mathcal{G}_i$ (see figure 3.9) is required as input to the quadratic programming framework for the volume fraction update. The problem can be stated as:

$$k_i = |\mathcal{G}_i \cap \mathcal{M}(t^n)| = \int_{\mathcal{G}_i} H(-\phi(\mathbf{x}, t_n)) \, \mathrm{d}A, \tag{3.70}$$

where $H(x)$ is the Heaviside step function. To evaluate the integral, an algorithm with similarities to the marching squares algorithm is proposed – unfortunately standard quadrature-based numerical integration schemes do not apply due to the Heaviside step function. The algorithm, which we refer to as *recursive triangles*, repeatedly subdivides the domain of integration into tesselating triangles, and can also be used to approximate the zero contour of ϕ inside $\mathcal{G}_i$ by a piecewise-linear line. Details of this algorithm are provided in algorithm 2. A significant advantage of this approach is that it is completely independent of topology, while the accuracy can be controlled by the parameter L_{max}. By using this estimate for k_i, the volume fraction update procedure summarised in section 3.5.1.4 is complete. The full hybrid scheme is finished by defining an update scheme for ϕ, which is described in the following section.

Algorithm 2: The recursive triangles algorithm for estimating Equation (3.70). The domain of integration is repeatedly subdivided into triangles, and the parameter L_{max} defines the maximum number of subdivisons before the base case is called. For efficiency the domain is only subdivided close to the zero level set of ϕ. The algorithm can be used to approximate the zero contour of ϕ if each segment of the zero level set which cuts the triangle in the base case is stored.

```
Data: A level set field $\phi$, a polygonal region $\mathcal{G}_i$, an integer $L_{max}$
Triangulate $\mathcal{G}_i$ and apply RecursiveTri() to each one with $L = 0$:

area ← 0
Algorithm RecursiveTri(triangle, L)
    if $L = L_{max}$ then
        area ← area + BaseCase(triangle)
    else
        $d$ ← value of $\phi$ at centroid of triangle
        if $|d| >$ max centroid-vertex distance of triangle        // Assumes $|\nabla\phi| = 1$
         then
            if $d < 0$ then
                area ← area + area of triangle
            end
        else
            Divide triangle across largest edge into tri1 and tri2
            RecursiveTri(tri1, L+1)
            RecursiveTri(tri2, L+1)
        end
    end
    return

Procedure BaseCase(triangle)
    Compute $\phi(\mathbf{x})$ at each of the three triangle vertices
    if All three points have $\phi < 0$ then
        return Area of triangle
    else if All three points have $\phi > 0$ then
        return 0
    else
        Approximate level set field as the plane passing through the three
          coordinates in $\mathbb{R}^3$: $(x_i, y_i, \phi_i)$ for $i = 1, 2, 3$, where $(x_i, y_i)$ are the
          coordinates in $\mathbb{R}^2$ of the $i$-th triangle vertex
        Compute linear equation of zero level set of this plane
        Cut triangle by zero level set
        return Area of cut triangle
    end
```

3.5.2.2 Level set field evolution

For each material spanning the domain $\mathcal{M} \subset \mathbb{R}^d$, a level-set function $\phi(\mathbf{x})$ is defined at time $t = 0$ to be a signed distance such that the Eikonal condition

$$|\nabla\phi| = 1, \tag{3.71}$$

is satisfied and the zero contour coincides with the free-surfaces: $\phi(\mathbf{x}) = 0, \forall\mathbf{x} \in \partial\mathcal{M}$. For any point away from the interface, the sign of the level-set thus indicates whether $\mathbf{x} \in \mathcal{M}$ or otherwise. Here it is assumed $\phi(\mathbf{x}) < 0, \forall\mathbf{x} \in \mathcal{M}$. The level set evolution is given by

$$\frac{\partial\phi}{\partial t} = \mathcal{L}(\phi, t) \qquad \mathcal{L}(\phi, t) = -\mathbf{u} \cdot \nabla\phi \ . \tag{3.72}$$

where $\mathbf{u}(\mathbf{x})$ is a given velocity field. The spatial operator $\mathcal{L}(\phi, t)$ in the level set evolution equation is discretised using the ninth order upstream-central (denoted as HOUC-9) scheme from [54], while the temporal discretisation is performed using the three level, third order SSP-Runge-Kutta scheme [103]. For efficiency purposes, only points local to the interface are updated, following the narrow band approach first described in [104]. For the sake of completeness these schemes are briefly summarised. In the computation of the spatial operator at cell (i, j) (equation 3.72), we have

$$[\phi_x]_{i,j} = \begin{cases} \dfrac{\mathcal{D}^+_{i,j}}{\Delta x} & \text{if } u_x < 0 \\ \dfrac{\mathcal{D}^-_{i,j}}{\Delta x} & \text{if } u_x > 0 \\ 0 & \text{if } u_x = 0 \end{cases} , \tag{3.73}$$

where the ninth order accurate left hand difference is:

$$\begin{aligned} \mathcal{D}^-_{i,j} = &- \frac{1}{630}\phi_{i-5,j} + \frac{1}{56}\phi_{i-4,j} - \frac{2}{21}\phi_{i-3,j} + \frac{1}{3}\phi_{i-2,j} \\ &- \phi_{i-1,j} + \frac{1}{5}\phi_{i,j} + \frac{2}{3}\phi_{i+1,j} - \frac{1}{7}\phi_{i+2,j} + \frac{1}{42}\phi_{i+3,j} \\ &- \frac{1}{504}\phi_{i+4,j} + \mathcal{O}\left(\Delta x^9\right), \end{aligned} \tag{3.74}$$

and the right hand difference can be obtained by reversing the signs on the coefficients and on the x-index offset. The upwind estimate of $[\phi_y]_{i,j}$ follows similarly. The third order SSP-Runge-Kutta scheme computes two intermediate solutions in order to advance

ϕ^n to ϕ^{n+1} over a time step of length Δt:

$$\begin{aligned}
\phi^{(1)} &= \phi^n + \Delta t \mathcal{L}\left(\phi^n,\ t\right) \\
\phi^{(2)} &= \frac{3}{4}\phi^n + \frac{1}{4}\phi^{(1)} + \frac{1}{4}\Delta t \mathcal{L}\left(\phi^{(1)},\ t + \Delta t\right) \\
\phi^{n+1} &= \frac{1}{3}\phi^n + \frac{2}{3}\phi^{(2)} + \frac{2}{3}\Delta t \mathcal{L}\left(\phi^{(2)},\ t + \frac{1}{2}\Delta t\right).
\end{aligned} \tag{3.75}$$

3.5.2.3 Level set correction

As it stands, this scheme is capable of updating a grid of volume fractions directly off a level set function. However, no information flows in the other direction – the level set function is not corrected so that it remains consistent with the volume fractions. This represents a serious obstacle in using the method for highly deformational flows where the level set function may suffer spurious mass loss. Two approaches have been investigated, neither of which worked satisfactorily.

- Linearise a function describing the inconsistency between the volume fractions and level set field. This results in a sparse linear system of size N^2, where N is the number of mixed cells. The system must be solved several times per time step, and was found to add massive computational cost, and suffered from problems in robustly estimating derivatives of the inconsistency function.

- Explicitly locate the zero level set contour, and march this contour outwards/inwards as appropriate in each cell, so that the area contained is equal to the cell volume fractions. The level set field was then reset from the new contour location using the fast sweeping method [105]. This was found to adversely affect the accuracy of the method, although good results were obtained by discretising the level set function on a much higher-resolution grid than the volume fractions.

Instead, the method is presented without any level set correction procedure. As such, it will not work well in cases where the level set exhibits significant mass loss. Despite this limitation, results from advection test cases are presented which show the potential of this novel approach.

3.6 Numerical Results

The performance of an interface tracking method is typically quantified using an advection only problem with a pre-defined velocity field. A number of these test cases have become standard in the literature, allowing for straightforward comparison between methods. In the case of volume-tracking methods, knowledge of the exact solution allows for the L_1 error in the volume fractions to be computed as:

$$E_1 = \Delta x \Delta y \sum_{i,j} |z_{i,j}^{\text{computed}} - z_{i,j}^{\text{exact}}|. \tag{3.76}$$

In this section, the performance a large number of different interface tracking methods are evaluated on four different advection tests. These four tests are formally described in table 3.1, and they each aim to test different characteristics of an interface tracking method. The time step in all tests is set using a CFL condition of 0.8.

3.6.0.1 Aims of the numerical tests

Three original interface tracking methods have been developed in the preceeding sections of this chapter, and we now briefly set out the questions which these numerical tests should answer. In order to make accuracy and efficiency comparisons as fair as possible, all methods are implemented in the same piece of software and share as many underlying subroutines as possible.

- The design goal of the particle-VOF method was to improve the accuracy of the VOF reconstruction. As such, it will be compared against two existing VOF methods. Parker and Youngs' method [73] is a very common approach, which despite no longer providing state-of-the-art accuracy, is still commonly used due to ease of implementation. The LVIRA method [72] is more accurate, and is an example of a second-order VOF method (meaning that it reconstructs linear interface exactly).

- The efficient MOF (EMOF) method was designed to carry a lower computational cost than the MOF method, without sacrificing accuracy. This will be investigated by comparing accuracy, computational cost, and qualitatively examining the reconstructed interface of these two methods on each test case. The EMOF method is run with a 328MB (6400^2) look-up table, and the nonlinear optimisation problem in the MOF and EMOF interface reconstructions is solved using the same algorithm with identical termination conditions in order to make the comparison as fair as possible.

- The quadratic programming volume fraction update was used in a novel coupling of the level set and VOF methods. This method, denoted by QP-LSVOF, is compared

Test name	Velocity field	Spatial domain	Time	Initial condition
Zalesak disk	$u = \frac{\pi}{314}(50-y)$ $v = \frac{\pi}{314}(x-50)$	$[0,100]\times[0,100]$	$[0,628]$	Notched disk with centre $(50,75)$ radius 15 notch width 5 notch height 25
Single vortex	$u = -\sin^2(\pi x)\ \sin(2\pi y)$ $\cdot\cos\left(\frac{\pi t}{T}\right)$ $v = \sin^2(\pi y)\ \sin(2\pi x)$ $\cdot\cos\left(\frac{\pi t}{T}\right)$	$[0,1]\times[0,1]$	$[0,8]$	Disk with centre (0.5,0.75) radius 0.15
Frontogenesis	$u = -y\ \omega(r)$ $v = x\ \omega(r)$ where $\omega(r) = \frac{1}{r}U_T(r)$, $U_T(r) = 2.5980762\cdot$ $\text{sech}^2(r)\text{tanh}(r)$, $r = \sqrt{x^2+y^2}$.	$[-5,5]\times[-5,5]$	$[0,20]$	Half space $y \leq 0$
Multi-vortex	$u = -\sin(4\pi(x+0.5))$ $\cdot\sin(4\pi(y+0.5))$ $\cdot\cos\left(\frac{\pi t}{T}\right)$ $v = -\cos(4\pi(y+0.5))$ $\cdot\cos(4\pi(x+0.5))$ $\cdot\cos\left(\frac{\pi t}{T}\right)$	$[0,1]\times[0,1]$	$[0,4]$	Disk with centre (0.5,0.5) radius 0.15

Table 3.1: Description of the four interface advection test cases with pre-defined velocity fields.

against the existing coupled level set and VOF method [55], which uses a completely different coupling mechanism.

3.6.1 Single vortex test

The single vortex test [106] examines the ability of the method to represent thin filaments. Due to the time-reversed velocity field the initial conditions are recovered at time $t = 8$. The error convergence of all methods is shown in figure 3.10, and a comparison of the reconstructed interface are shown in figures 3.11 and 3.12. Note that the reconstructed interface from the QP-LSVOF method is not shown since this is a reconstruction-free method.

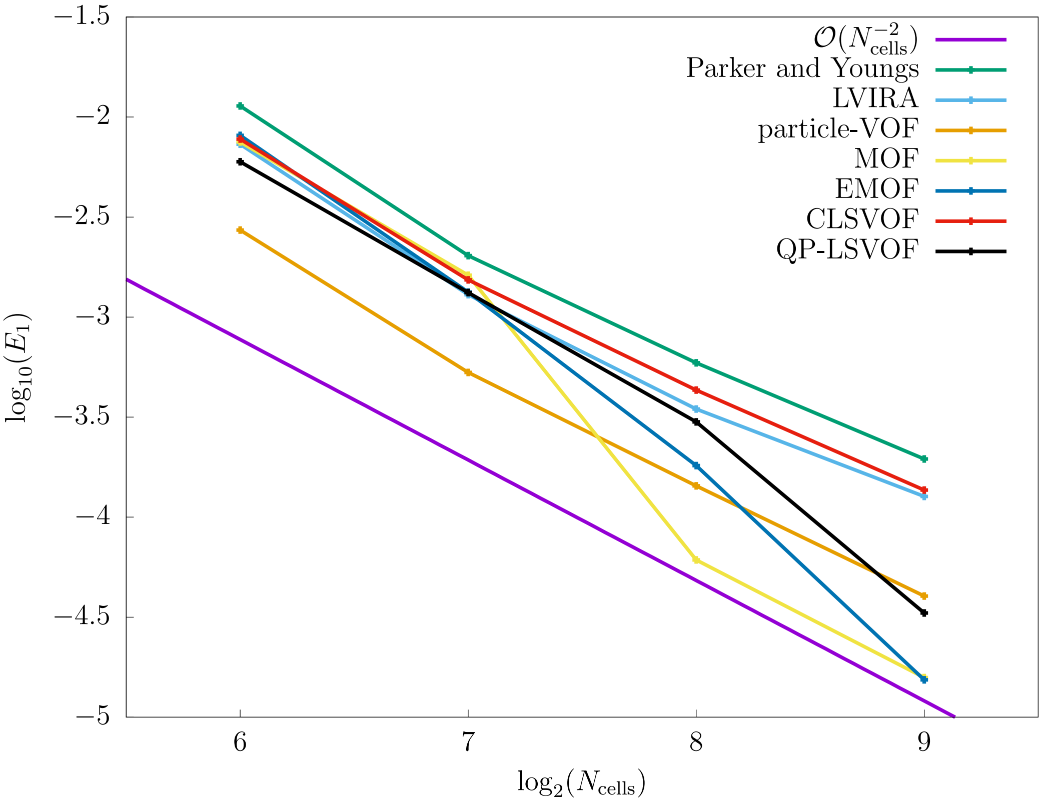


Figure 3.10: Comparison of geometric error between interface tracking methods on the single vortex test. N_{cells} refers to the number of computational cells in each spatial dimension.

The particle-VOF method significantly outperforms Parker and Youngs and LVIRA on this test case. Indeed, the error on the particle-VOF method is roughly equal to the error exhibited by the Parker and Youngs method using twice as many grid cells in each spatial dimension. The comparison of reconstructed interfaces in figure 3.11 shows that the particle-VOF method is less prone to spurious filament break-up than Parker and Youngs. The MOF and EMOF methods both exhibit better convergence than the VOF methods, and provide very similar accuracy on three out of four resolutions used. However, there is a large difference between MOF and EMOF on the 256^2 grid. We speculate that this is simply due to the significant nonlinearities introduced by the time-reversal of the velocity field, since the two methods exhibit very similar performance on all other resolutions. The QP-LSVOF method performs better, and even seems to converge faster than the traditional CLSVOF method.

3.6.2 Zalesak disk test

The Zalesak disk test [107] applies a velocity field corresponding to a solid-body rotation around the centre of the domain. The error is computed after completion of one full rotation, when the initial conditions are recovered in the exact solution.

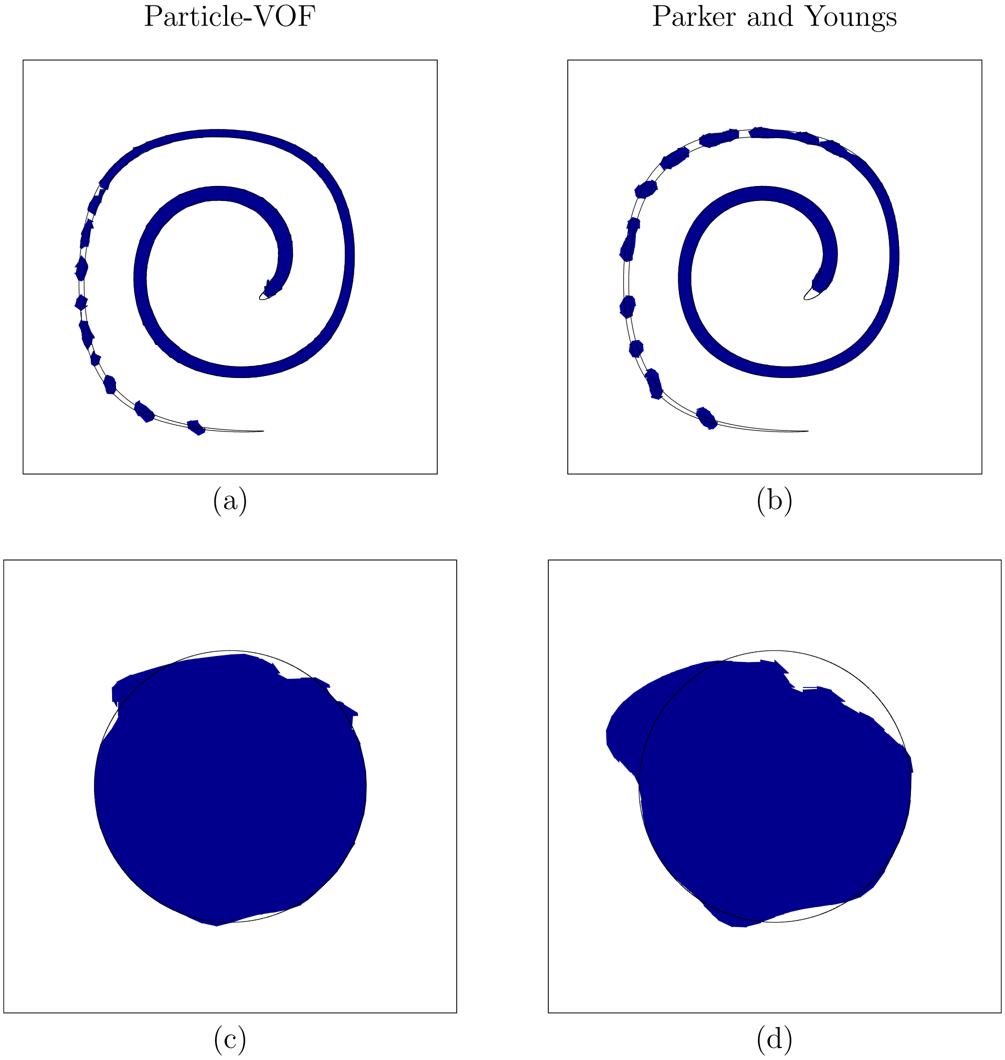


Figure 3.11: Comparison of indicator function reconstructions between the particle-VOF (left column) and Parker and Youngs' (right column) VOF methods. Solid colour denotes the reconstructed material region, while the line shows the exact interface position. Plots (a) and (b) show $t = 4$ on the single vortex test, (c) and (d) show $t = 8$ on the single vortex test (both using a 64^2 grid).

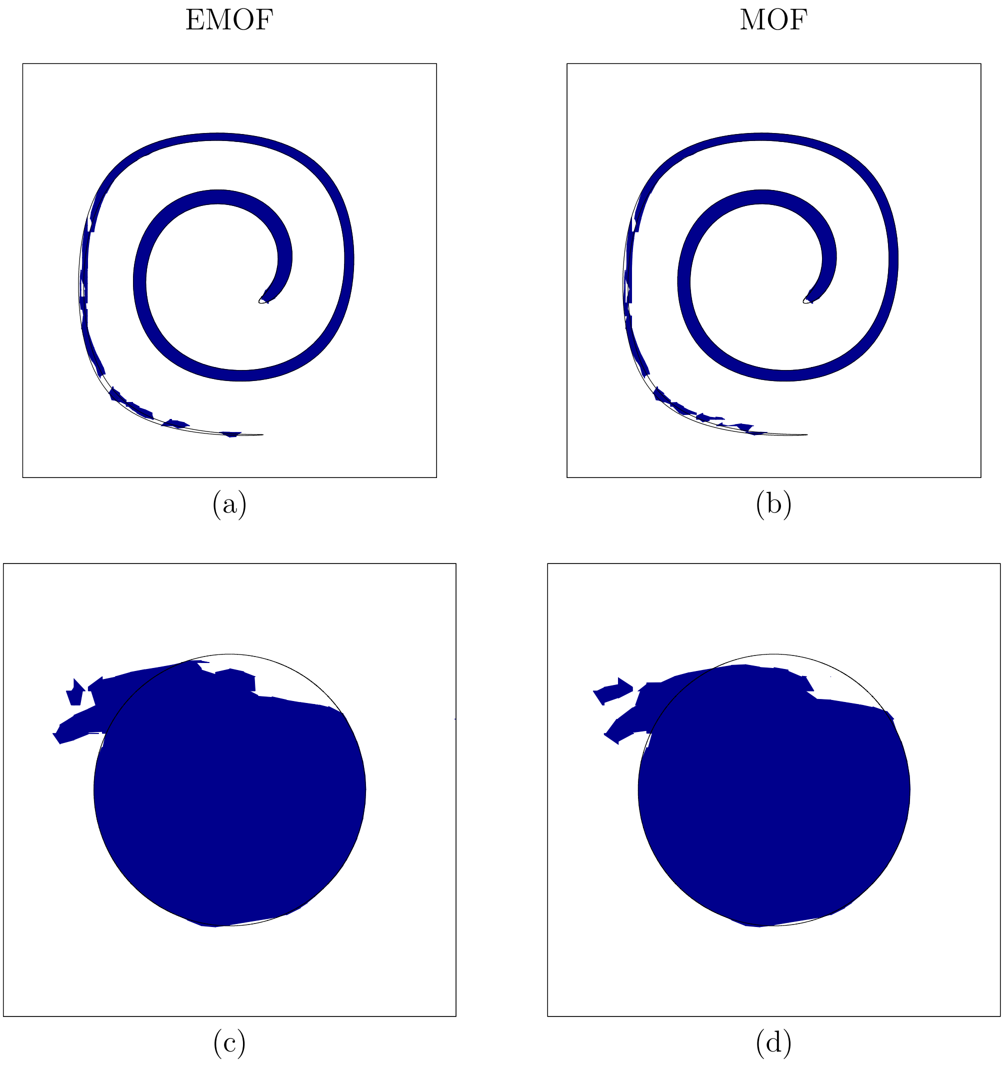


Figure 3.12: Comparison of indicator function reconstructions between the EMOF (left column) and MOF (right column) methods. Solid colour denotes the reconstructed material region, while the line shows the exact interface position. Plots (a) and (b) show $t = 4$ on the single vortex test, (c) and (d) show $t = 8$ on the single vortex test (both using a 64^2 grid).

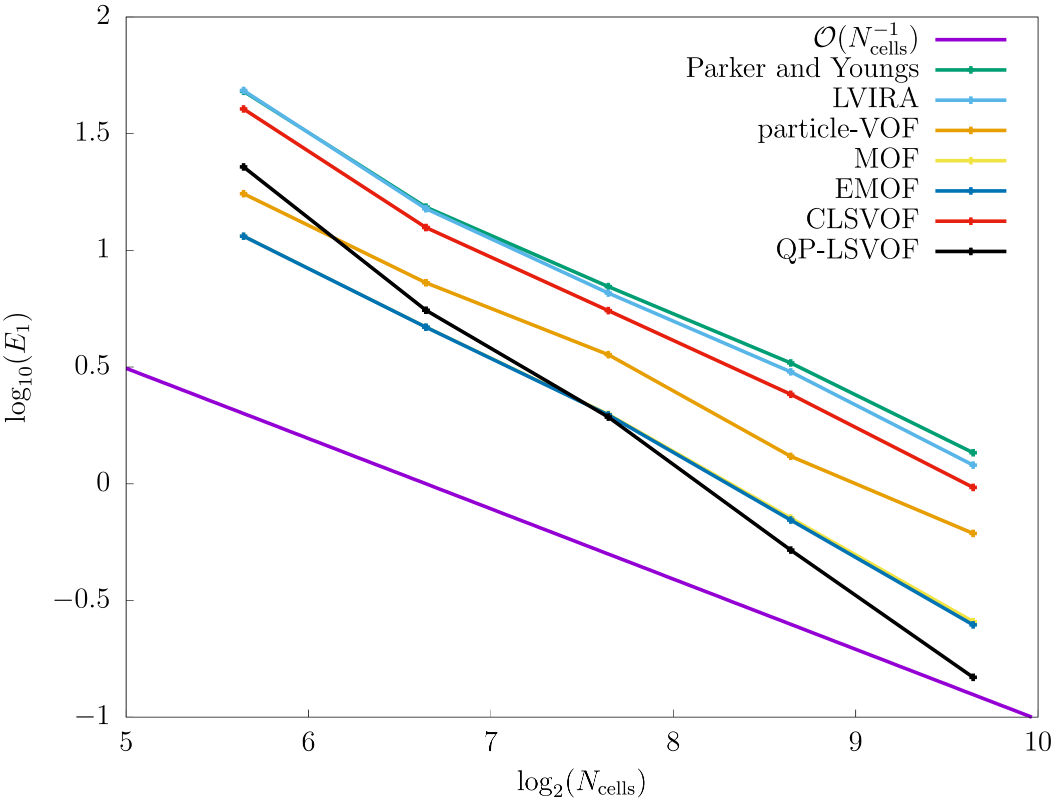
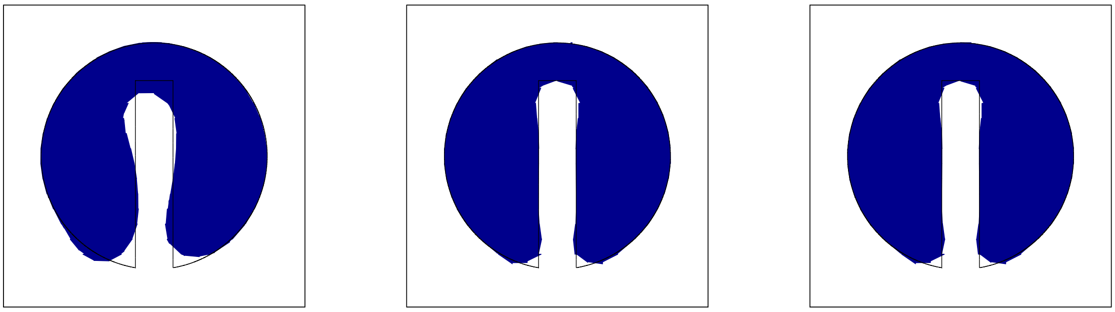


Figure 3.13: Comparison of geometric error between interface tracking methods on the zalesak disk test. N_{cells} refers to the number of computational cells in each spatial dimension.

Figure 3.13 shows a comparison of the error convergence between all interface tracking methods on this test case. All methods exhibit first order convergence on this test case due to the sharp corners present on the interface, although the QP-LSVOF method converges noticeably quicker than all other methods. While the LVIRA method is only marginally superior to Parker and Youngs, the particle-VOF method consistently exhibits less than half the error of the other two reconstruction methods. The EMOF method exhbits almost exactly the same accuracy as the MOF method on all resolutions, and the two lines are almost indistinguishable.

A comparison of the reconstructed interface on the coarsest resolution between the Parker and Youngs, MOF, and EMOF methods is shown in figure 3.14. This is further evidence that the approximations in EMOF do not cause a loss of accuracy.

Some error values from state-of-the-art methods in the literature are shown in table 3.2. Results from our implementations compare favourably against previous implementations of the MOF method, as well as against the particle level set method.



Parker and Youngs

MOF

EMOF

Figure 3.14: Comparison of indicator function reconstructions at the end of the zalesak disk test case ($t = 628$) using the coarse 50^2 grid. All plots show the region $[30, 70] \times [55, 95]$.

	N_{cells}		
Method	100	200	400
CLSMOF (from [85])	7.76	3.07	1.16
MOF (from [85])	6.11	2.22	0.75
Particle level set (from [56])	10.1	2.88	-
EMOF	4.69	1.97	0.69

Table 3.2: Some results from state-of-the-art methods in the literature for the symmetric difference error after one revolution in the Zalesak disk test, compared against results obtained with the EMOF method.

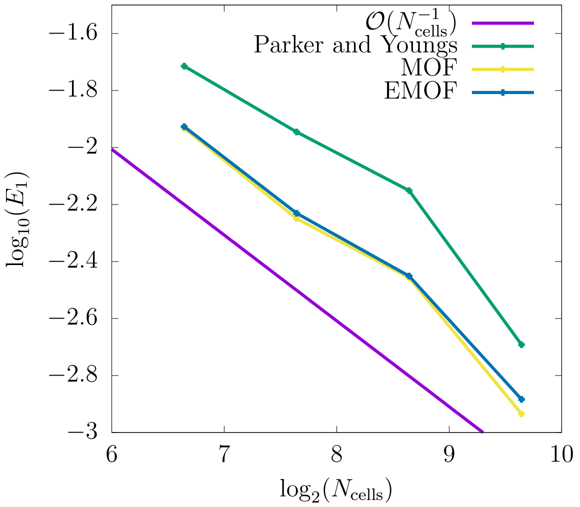


Figure 3.15: Comparison of geometric error between interface tracking methods on the multi vortex test. N_{cells} refers to the number of computational cells in each spatial dimension.

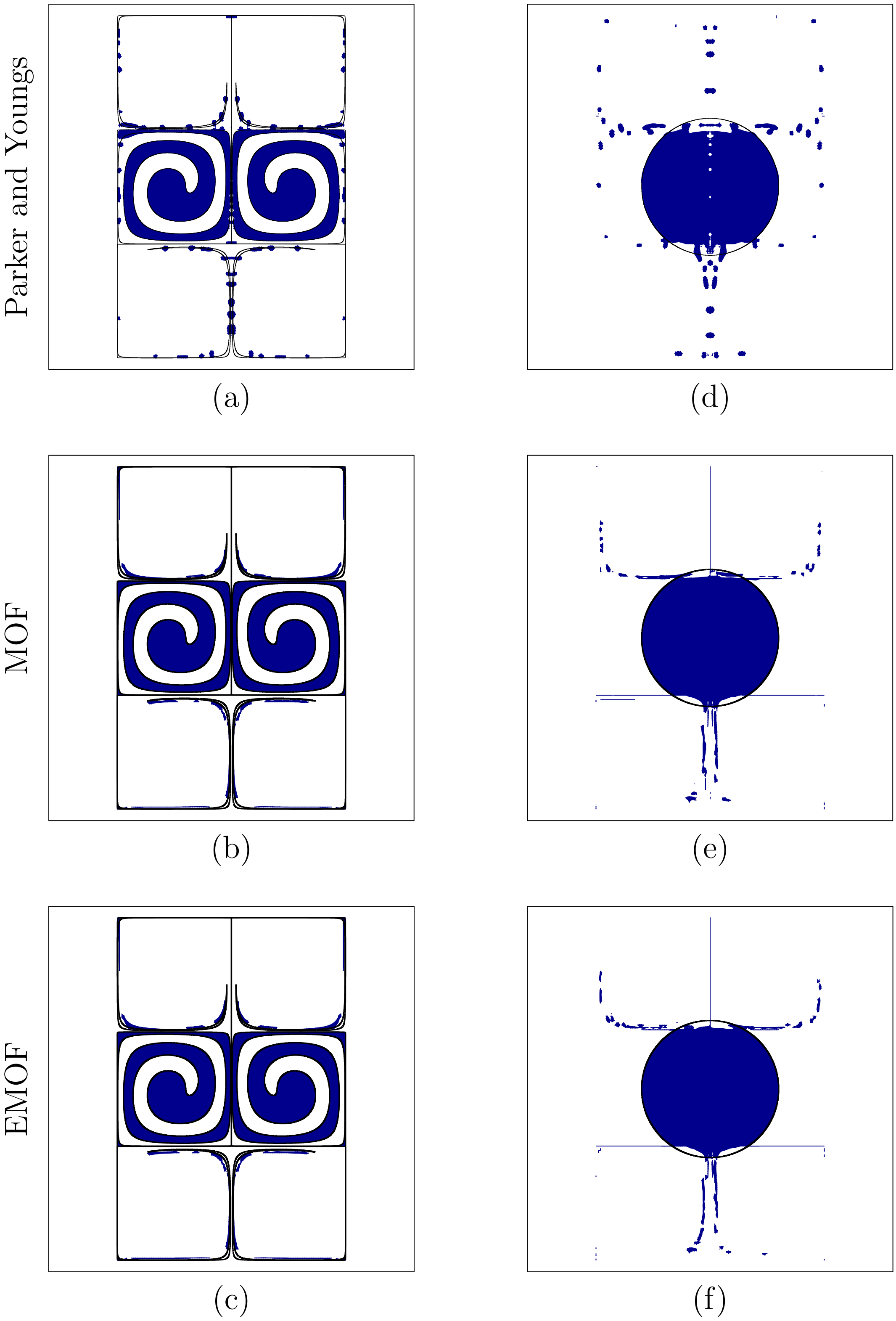
3.6.3 Multi-vortex test

This test imposes a particularly severe deformation to the interface, and features extremely thin filaments which are aligned with the grid. The extreme nature of this test makes it suitable to try and discern differences between the MOF and EMOF methods – since both methods will be forced to represent unresolvably-thin filaments. Results from Parker and Youngs method are also shown as a benchmark. The error convergence of these three methods is shown in figure 3.15, and comparisons of the reconstructed interface geometry are shown in figure 3.16. Results from the particle VOF and QP-LSVOF methods are not shown on this test case, since they do not add any conclusions beyond those already drawn from the single vortex test.

Despite the unresolvable filaments, the EMOF method results in almost-identical errors to MOF. Encouragingly, comparing the interface reconstructions between MOF and EMOF in figure 3.16 show that both methods fail in almost exactly the same way.

3.6.4 Frontogenesis test

The frontogensis test case [108], which models a meteorological process, has not been published in the context of interface tracking methods before. In the frontogensis test case an initially planar interface is drawn out into a spiral featuring progressively thinner and thinner filaments. This test has the useful property of possessing an analytic solution for all $t > 0$. As a result it is possible to watch the evolution of the volume fraction error as the interface is drawn out into progressively thinner filaments. The error initially increases linearly with time, but at some point the filaments become too thin for the interface tracking method to maintain, and as they break the error increases dramatically.



(a) (d)

(b) (e)

(c) (f)

Figure 3.16: Comparison of indicator function reconstructions between Parker and Youngs' method (top row), the MOF method (middle row), and the EMOF method (bottom row) on the multi-vortex test case using a 200^2 grid. All plots show the region $[0.1, 0.9] \times [0.1, 0.9]$. The black line shows the interface position in the exact solution. Plots **(a)**, **(b)** and **(c)** correspond to time $t = 2$, **(d)**, **(e)** and **(f)** correspond to time $t = 4$.

This analytic solution for the indicator function $z(x, y, t)$ can be written as [109]:

$$z(x, y, t) = z_0(y \cos(\omega t) - x \sin(\omega t)), \tag{3.77}$$

where

$$z_0(y) = z(x, y, 0) = \begin{cases} 0 & y > 0 \\ 1 & y \leq 0 \end{cases} . \tag{3.78}$$

and ω is defined in table 3.1. In order to estimate the exact value of $z_{i,j}$ in a cell we sample the value of the exact indicator function over 25^2 uniformly distributed points in each cell.

A plot of the error as a function of time for all interface tracking methods is shown in figure 3.17. The first thing to note is that only two methods offer any real improvement over Parker and Youngs on this test – MOF and EMOF, whose errors are essentially equal for all times. Since this test examines the minimium filament width which can be represented by an interface tracking method, this means that particle-VOF does not offer an improvement in this respect. The accuracy increases in previous tests are instead due to the particles offering superior resolution of sharp corners. The QP-LSVOF method performs particularly badly – this is due to regularisation of the level set and illustrates the need for a level set correction approach.

Comparisons of the interface geometry at various times are shown in figure 3.18. This is a particularly clear example of the superiority of MOF and EMOF in representing thin filaments.

3.6.5 Computational efficiency

The relative computational efficiency of each method was identical across the four test problems, and as a result we only present efficiency results from the Zalesak disk test case. In order to compare the relative computational efficiencies of these methods as accurately as possible, they are implemented in the same piece of software and share as many underlying routines as possible. We present the time taken to run the Zalesak disk test case in table 3.3. This shows that the cost of the particle-VOF method lies somewhere between Parker and Youngs and LVIRA, despite the additional burden of marker particle storage and advection. The particle-VOF method requires approximately half the resolution to achieve the same accuracy as Parker and Youngs, yet costs only approximately double. This compares favourably to the four-fold increase in cost which follows from doubling resolution. In addition to being one of the most accurate methods, the EMOF method is also one of the cheapest, and it achieves almost a factor of three speed-up over the MOF method.

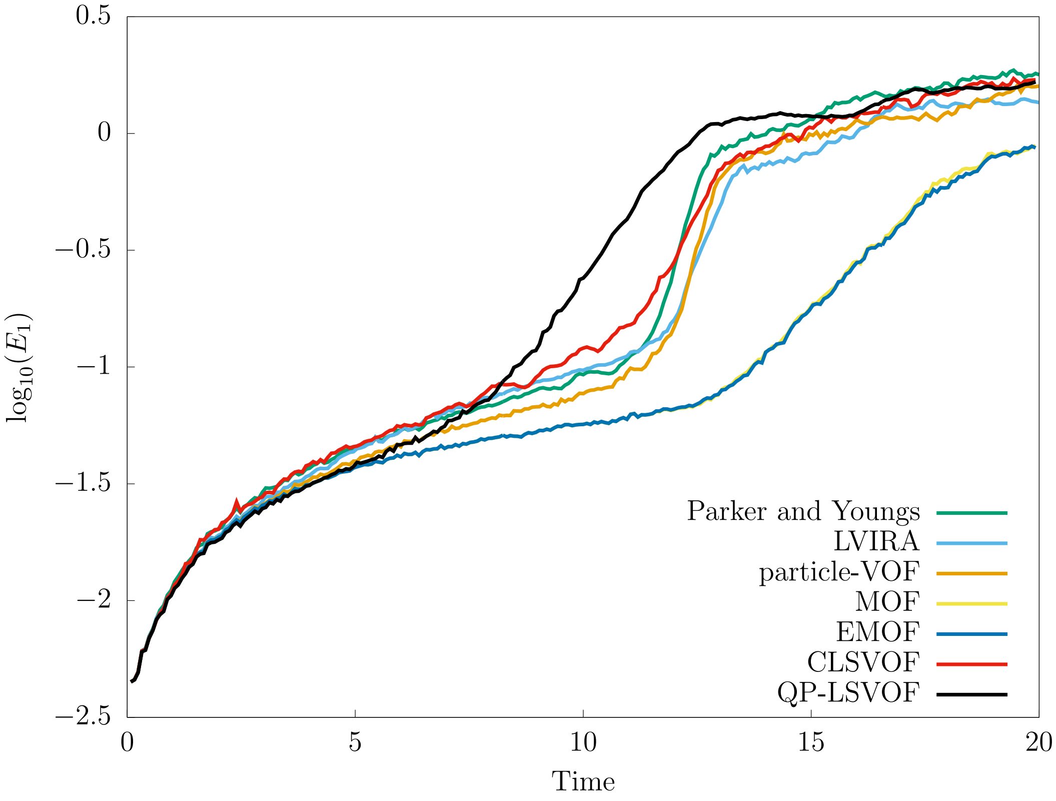


Figure 3.17: Comparison of geometric error between interface tracking methods on the frontogenesis test.

	N_{cells}				
Method	50	100	200	400	800
Parker and Youngs	0.71	3.1	13	59	285
EMOF	1.5	5.9	24	100	450
CLSVOF	0.89	4.1	20	100	600
Particle-VOF	1.2	4.1	20	110	650
QP-LSVOF	1.7	6.8	31	150	820
LVIRA	3.8	16	63	260	1100
MOF	4.1	18	73	300	1200

Table 3.3: Run-time in seconds for the Zalesak disk test. Methods are ordered by the run-time on the finest resolution.

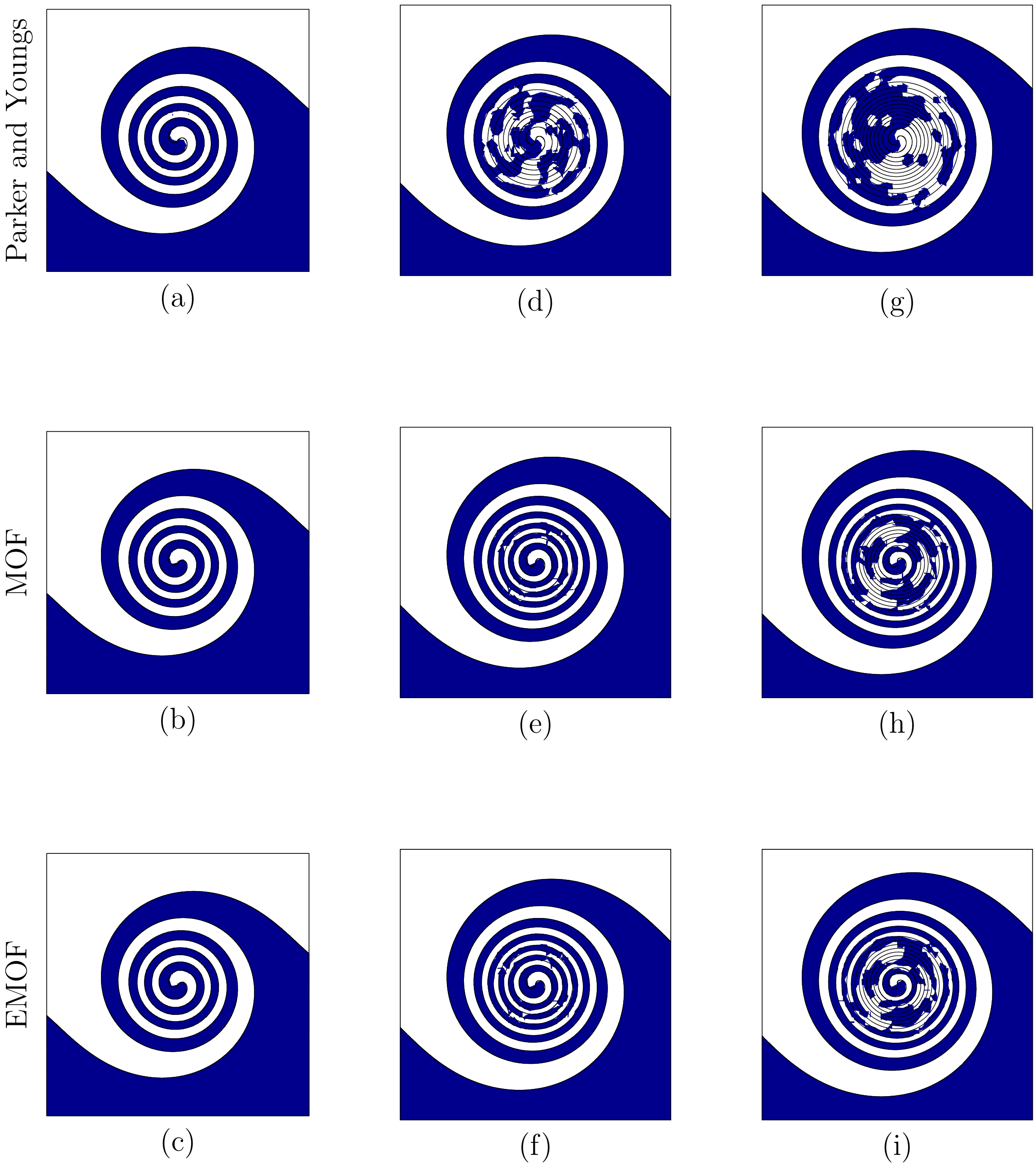


Figure 3.18: Comparison of indicator function reconstructions between Parker and Youngs' method (top row), the MOF method (middle row), and the EMOF method (bottom row) on the frontogenesis test case using a 100^2 grid. All plots show the region $[-2.5, 2.5] \times [-2.5, 2.5]$. The black line shows the interface position in the exact solution. Plots **(a)**, **(b)** and **(c)** correspond to time $t = 10$, **(d)**, **(e)** and **(f)** correspond to time $t = 15$, and **(g)**, **(h)** and **(i)** correspond to time $t = 20$. It is clear that the MOF method is able to represent thinner filaments than Parker and Youngs' method, and that EMOF gives an almost identical solution to MOF.

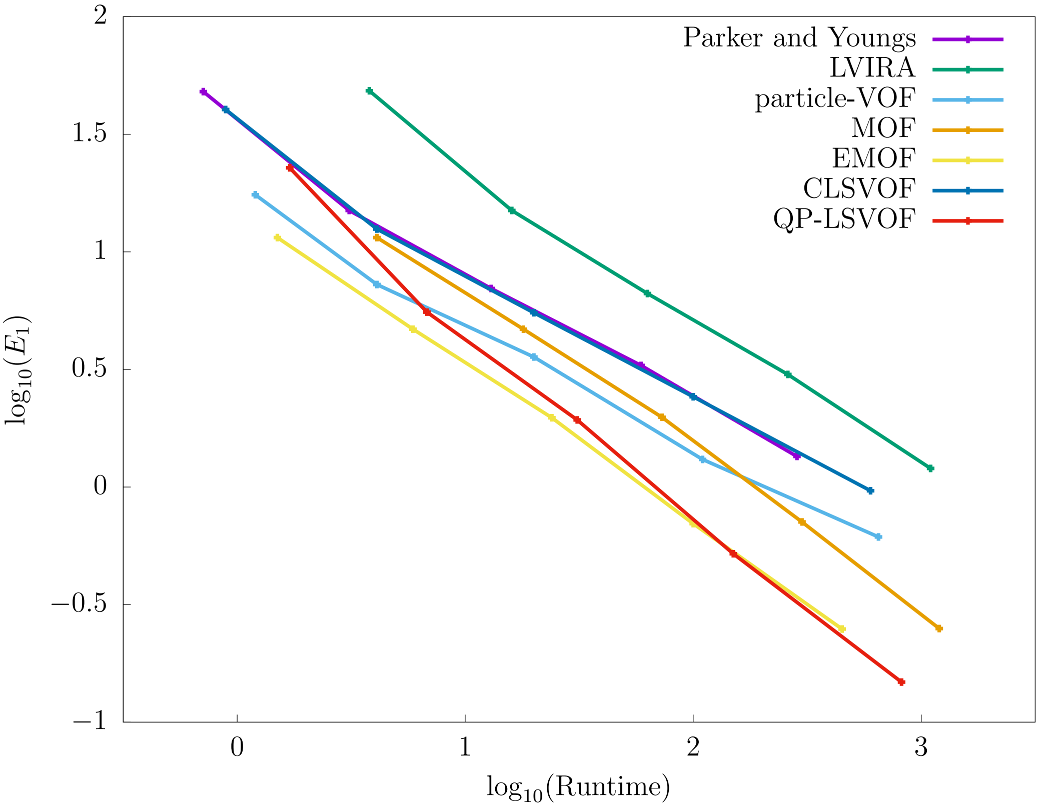


Figure 3.19: Comparison of geometric error as a function of run time between interface tracking methods on the zalesak disk test.

3.7 Conclusions

In this chapter we have presented three independent developments to interface tracking methods.

Particle-VOF Firstly, we developed an interface reconstruction procedure based on a linear least squares fit to marker particles on the interface. The addition of marker particles was shown to consistently increase the accuracy of VOF reconstruction on several classic test problems, particularly around sharp corners, although the frontogenesis test demonstrated that the minimum resolvable filament width is not improved. The particle-based reconstruction process is simple, and can be easily retro-fitted to existing VOF codes. The accuracy of this approach remains inferior to MOF-type methods however.

EMOF The EMOF method is an efficient moment-tracking method which utilises a pre-computed two-dimensional function to speed up evaluations of the objective function in the MOF interface reconstruction problem. The results across all four advection tests consistently show that the EMOF method has succeeded in maintaining the accuracy of MOF while providing a significant speed-up. The EMOF method runs approximately a factor of three times faster than MOF, and just under a factor of two times slower than Parker and Youngs' VOF. The method is very simple to implement, and retains the small numerical domain of dependence of MOF, making it trivial to parallelise. The amount of memory required for the look-up table is relatively small by modern standards, and the most significant disadvantage is the incompatibility of the method with curvilinear grids.

Of the methods tested here, the EMOF method is consistently one of the most accurate, whilst being the second fastest. As a result, we chose this method to take forward and couple to a flow solver. Results obtained by coupling the EMOF method to a ghost fluid solver are presented in section 4.6.

QP-LSVOF The reconstruction-free volume fraction update framework underlying the QP-LSVOF method is a particularly novel approach to updating volume fractions– nothing similar exists in the literature. The accuracy results from QP-LSVOF demonstrate the potential of this approach to updating volume fractions, but the lack of a successful level set correction step mean that this method has limited practical applicability. However, we emphasise that the reconstruction-free volume fraction update framework is applicable for any arbitrary initial guess – a level set field is not the only way to use this framework.

CHAPTER 4

The ghost fluid method

We now turn our attention to flow solvers capable of simulating multi-material flow. In this chapter we investigate the ghost fluid method, originally proposed by Fedkiw et al. in [38]. The ghost fluid method maintains separate grids for each material, and tracks the interface location with a level set field. The effect of the material interface is modelled by appropriately setting the states of the ghost cells (cells which lie in the region occupied by the other material). After settings these states, an unmodified single-material solver may be applied to each material. The great advantage of ghost fluid methods is their simplicity: by defining ghost material states throughout the domain any single material method can be applied to each material separately. This simplicity comes at the cost of conservation, since the flux is no longer single-valued at the interface. The ghost fluid approach is quite general, and can be used to track other discontinuities (e.g. shocks) governed by any conservation law systems.

A large range of ghost fluid methods now exist. The first attempt at a systematic comparison is given in [110], where several varieties of the method are analysed, but full comparison of accuracy, conservation error, and robustness over a wide range of problems has yet to be seen. Furthermore, the effect of the accuracy of the interface tracking method on the accuracy of solutions has never been quantified. It is also not clear how the conservation properties of the interface tracking method affect the conservation properties of the method as a whole. In the remainder of this chapter we begin to address these questions before proposing a novel coupling of the ghost fluid method to the efficient moment-of-fluid interface tracking method developed in the previous chapter.

4.1 The ghost fluid method algorithm

The following sequence describes in general terms how a generic ghost fluid method advances the multi-material problem from time level n to time level $n + 1$ (possibly as part of a multi-step time integrator).

1. Set the state of any freshly-cleared cells (defined in level set terms by $\phi_i^{n-1}\phi_i^n < 0$). This means that a new real state has been uncovered at this cell, which must be initialised. No consensus exists on the optimal strategy in this case, even in the 1D case. This issue is touched on in [111], where it is recommended to simply set the state equal to the ghost state which previously existed there.

2. Define the ghost states appropriately (this differs according to the various types of ghost fluid method), where the level set interface tracking method is used to differentiate between the real fluid and the ghost fluid at each grid point.

3. Extrapolate the ghost fluid states (and possibly the interface extension velocity field) away from the interface. The problem of extrapolating variables away from the interface in dimension greater than one is not trivial. There are two main approaches to this problem. If the real fluid region is defined by $\phi(\mathbf{x}) < 0$, the scalar ψ can be extrapolated into the ghost region by solving the following hyperbolic partial differential equation to steady state in pseudo-time τ:

   $$\frac{\partial \psi}{\partial \tau} + \frac{\nabla \phi}{|\nabla \phi|} \cdot \nabla \psi = 0. \tag{4.1}$$

   This equation can be solved using standard discretisations. The characteristics have unit speed, meaning that at time t, ψ has been extrapolated into a band of width t in the ghost region. The second commonly-seen approach to this problem applies the fast marching method, which leverages the direction of information transfer to extrapolate ψ faster than is possible with the PDE-based approach. The ghost cell extrapolation is only important in a narrow band of ghost cells along the interface (whose width depends on the stencil size of the single-material solver).

4. Compute a time step which is stable for both fluids.

5. Apply a single material flow solver to update each material independently.

6. Advance the level set to t^{n+1}. The level set function used to track the interface ($\phi(x,t)$) is evolved according to the partial differential equation

   $$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \tag{4.2}$$

   where $\mathbf{u}$ is the extension velocity field. The ghost fluid method is used to set the value of u in cells adjacent to the interface, and then the values are extrapolated into the rest of the domain. Many numerical methods are available for the solution of this advection equation- from the first order upwind method to more sophisticated method-of-lines approaches with high order WENO spatial discretisations and Runge-Kutta time stepping.

4.2 A review of the ghost fluid method literature

The definition of the ghost cells is the principal manner by which ghost fluid methods differ, and the freedom in setting these values allows for a range of interfacial boundary conditions to be captured. In this section we present a survey of the various ghost fluid methods, first presenting methods which differ only in the way in which the ghost cells are set, and then exploring some more fundamentally different methods which still fall under the ghost fluid method category.

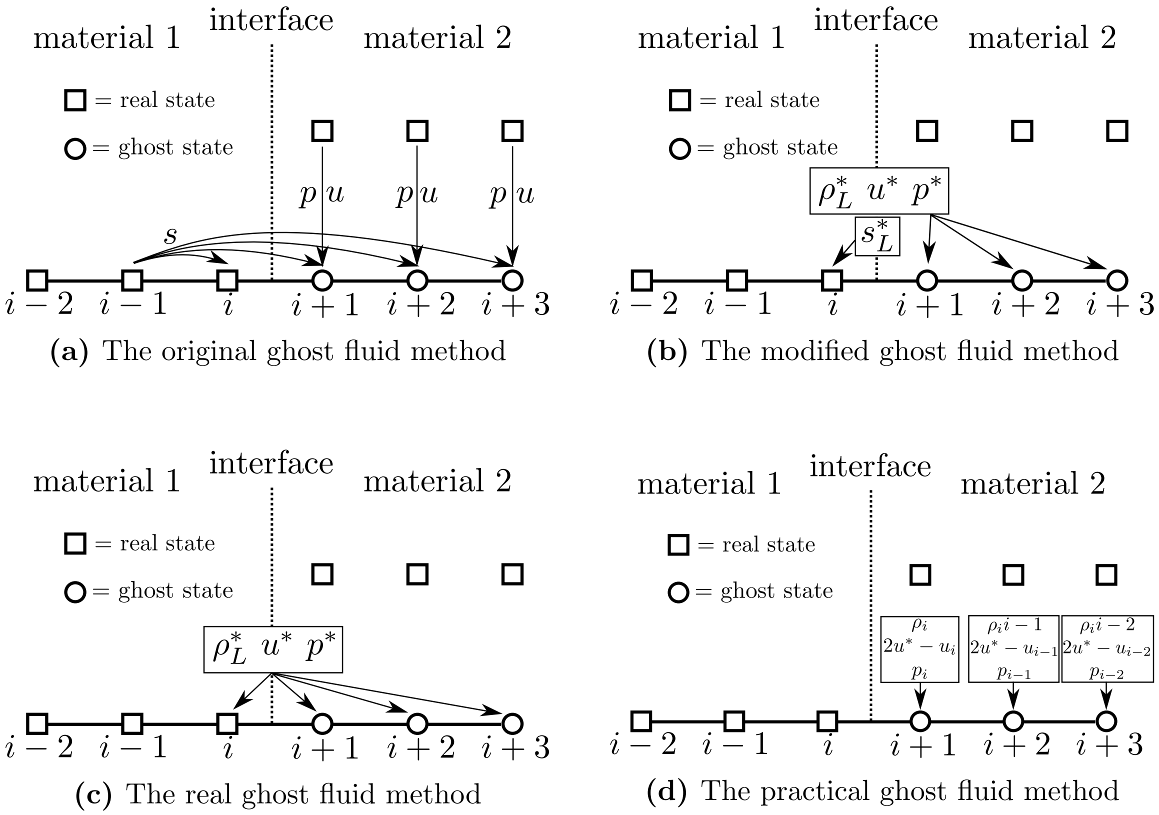
The original ghost fluid method Fedkiw et al. [38] set pressure and velocity in each ghost cell equal to that of the real material in the same cell in the original description of the ghost fluid method (OGFM). One additional quantity is required to reconstruct the conserved variables- this is chosen to be entropy, since entropy is discontinuous at a contact discontinuity, but otherwise constant in regions where the solution is smooth. The entropy in each ghost cell is found by constant extrapolation outwards from the interface.

The so-called *isobaric fix* to the original ghost fluid method, proposed by Fedkiw et al. in the original paper extrapolates the entropy from one cell further into the interior of the material. The entropy is extrapolated to the real cell adjacent to the interface as well as the ghost cells- this is designed to reduce the overheating effect observed close to the interface.

The OGFM is reported to face difficulties when large differences in states or material properties exist at the interface [112]. In this case, the method may exhibit oscillations near the interface or fail altogether.

The modified ghost fluid method The first major advances to the ghost fluid method was due to Liu et al. in 2003 [39] in the modified ghost fluid method (MGFM). Under this approach, a mixed Riemann problem is solved approximately across the material interface to predict the states adjacent to the interface. This predicted state is extrapolated outwards to the ghost states. The density of the real cell adjacent to the interface is fixed using the entropy predicted in the mixed Riemann problem solution.

The interface ghost fluid method The interface ghost fluid method (IGFM) of [112] applies the method of characteristics to the problem of setting ghost states. Although only derived for the stiffened and ideal gas equations of state, their method correctly captures



(a) The original ghost fluid method

(b) The modified ghost fluid method

(c) The real ghost fluid method

(d) The practical ghost fluid method

Figure 4.1: The operation of the four ghost fluid methods under investigation in one dimension. Star-state values are obtained by solution of a mixed Riemann problem at the interface.

wave interactions at the interface. Under the assumption that all interface interactions are isentropic, the interface state is computed. This development then introduced the idea of over-writing the outermost real fluid state with the interface state. The ghost pressures and velocities are extrapolated outwards equal to the interface pressure and velocity, while density is set by extrapolating entropy from the outermost real cell. In one and two-dimensional test cases the IGFM is shown to provide superior accuracy compared to the OGFM. The conservation properties are also improved.

The real ghost fluid method The current widely-used state of the art approach to the ghost fluid method is known as the real ghost fluid method (RGFM) [40]. The RGFM can be thought of as an alternative to the IGFM where the ghost states are instead populated using the exact solution of a mixed Riemann problem normal to the interface. The initial data for this mixed Riemann problem are taken from real states on opposite sides of the interface. The outermost layer of real cells is also overwritten by these star states.

The Riemann ghost fluid method The Riemann ghost fluid method, due to Sambasivan and Udaykumar in [111], modifies the RGFM in one way: the initial data for the mixed Riemann problem are computed by interpolating the real fluid primitive variables at a probe point a small distance inside each material, to prevent the solution being affected by erroneous interfacial states. It is not obvious that this modification is an improvement over the RGFM, and the two methods are not directly compared in [111].

The practical ghost fluid method A very recent development is the practical ghost fluid method (PGFM) [113]. The authors systematically examine all possible wave structures in the interaction between the real and ghost fluid on the interface, in order to derive a ghost state such that the real star state in the Riemann problem between the real fluid and ghost fluid is identical to the corresponding star state in the mixed Riemann problem normal to the interface. It is shown that the moving wall boundary condition (where the exact interface velocity is used) fulfills this condition. The advantage of this approach is that only interfacial velocity is needed from the mixed Riemann problem on the interface. The method is shown to perform very similarly to the MGFM on standard 1D and 2D shock interaction problems.

This literature review is finished by a brief description of some novel variations on the ghost fluid approach, which go beyond just setting the state in ghost cells.

Ghost fluid method for the poor The GFMP, proposed in [114] and developed in [115], reduces the overhead of the GFM by avoiding the storage of any ghost cells. Instead, two fluxes are computed across the interface, each using different equations of state. It is shown that for 1D ideal gases advanced with a Roe flux, the total mass and total momentum are exactly conserved, while energy conservation errors are negligible. In the extension of the scheme to multidimensions and general equations of state, a mixed Riemann problem is solved normal to the interface. Testing in [115] shows superior robustness properties relative to the OGFM on 1D shock tube test cases. The GFMP utilises a conservative level set formulation:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho\phi u) = 0. \tag{4.3}$$

Explicit simplified interface method The so-called ESIM [116] is an interesting approach to increasing the accuracy of the ghost fluid method extrapolations. By coupling zero-th and first order jump conditions with Taylor-like expansions of the primitive variables across the interface, the ESIM improves the accuracy and order of convergence compared to the GFM when the states next to the interface have non-flat profiles. Unfortunately the method has only been demonstrated for 1D stiffened gases, and no further progress has been reported since publication in 2005.

Conservative ghost fluid methods Since the flux of mass, momentum and energy across the interface is different for each material the ghost fluid method is not conservative. Conservation errors also occur when the zero level set passes across a cell centre during the time step, changing the definition of the real material in that cell. A couple of attempts have been made to eliminate the conservation errors in the ghost fluid method. The first, by Nguyen et al. [117] consists of tracking the conservation errors, and redistributing mass, momentum and energy at the end of every time step so that global conservation is satisfied (at the cost of local conservation). The second, by Liu et al. [118] is fully conservative for mass and momentum. Their approach is to store a third set of conserved variables in cells close to the interface. Then, when the interface passes over a cell centre, rather than changing a ghost cell into a real cell, the conserved variables in the new real cell are set in such a way as to reduce the conservation error.

Unfortunately it appears that neither attempt at constructing a conservative ghost fluid method was particularly successful as the methods have not seen wide uptake, and they appear to be detrimental to the accuracy of the solution close to the interface.

4.3 Quantitative comparison of one-dimensional ghost fluid methods

In order to cast more light on the relative advantages of each ghost fluid method, in this section we present a comparison of common ghost fluid methods on three one-dimensional test cases. In one dimension there are fewer degrees of freedom associated with the operation of the ghost fluid method. As a result, in this situation the Riemann GFM is identical to the real GFM, and only the following four approaches are compared:

1. The original ghost fluid method (OGFM), described in figure 4.1a.
2. The modified ghost fluid method (MGFM), described in figure 4.1b.
3. The real ghost fluid method (RGFM), described in figure 4.1c.
4. The practical ghost fluid method (PGFM), described in figure 4.1d.

The four types of ghost fluid method are applied to three problems involving materials governed by the ideal gas and stiffened gas equations of state. The exact solutions for all shock tube problems are obtained from the stiffened gas exact Riemann solver which is fully derived in appendix A. All accuracy errors are measured using the L1 error norm as described by equation 2.15. Since one of the principal drawbacks of the ghost fluid method is the lack of conservation, the conservation error in mass, momentum and energy is measured at every time step in each numerical experiment. Given an initial set of vectors of conserved variables in the problem, the vector describing the total mass, momentum and energy in the domain $[x_L, x_R]$ is:

$$\mathbf{U}_{\text{tot}}^{(0)} = \Delta x \sum_i \mathbf{U}_i^{(0)}. \tag{4.4}$$

Then by the application of the definition of a conservation law, the total mass, momentum and energy at time level $n+1$ is given by:

$$\mathbf{U}_{\text{tot}}^{(n+1)} = \mathbf{U}_{\text{tot}}^{(n)} + \Delta t \left(\mathbf{F}_L - \mathbf{F}_R \right), \tag{4.5}$$

where $\mathbf{F}_{L|R}$ is the flux at $x = x_{L|R}$ computed by the flow solver used to update the solution over that time step. In the computation of the total conserved variables according to the ghost fluid method, the sign of the level set function at each cell centre is used to determine which material state to take as the state in that cell. We consider a system in which both materials are governed by the one-dimensional Euler equations (equation 1.1). For both materials, the system is closed by the stiffened gas equation of state:

$$p = (\gamma - 1)\rho e - \gamma p_\infty. \tag{4.6}$$

After the ghost fluid method has set all ghost cells, the MUSCL-Hancock method is used to advance the conserved variables of both materials. This single-fluid algorithm is described in section 2.3. Unless otherwise stated, the CFL number of all simulations is 0.8, the number of cells is 100, and the computational domain is $[0, 1]$. All initial conditions are presented in terms of the vector of primitive variables for each fluid: $\mathbf{W} = (\rho, u, p)^T$.

4.3.1 Gas-gas shock tube problem

This is a classic shock tube problem, with the addition of a material interface at the initial discontinuity. This is a test of the ability of the method to advect the material interface at the correct speed, to reach appropriate states on each side of the discontinuity, and of conservation errors. In this test, both materials are ideal gases with $\gamma_1 = 1.4$ and $\gamma_2 = 1.2$. The interface is initially located at $x = 0.5$, with material 1 to the left hand side of this. The initial states are:

$$\begin{aligned} \mathbf{W}_1 &= \left(1.0, 0, 10^5\right)^T, \\ \mathbf{W}_2 &= \left(0.125, 0, 10^4\right)^T. \end{aligned}$$

The solution is output at time $t = 0.0007$. The convergence rate of the L1 density error, and the convergence errors in this test case are shown in Figure 4.2. Results from this test can also be found in test 2A in [38].

Remarks on Experiment 1

- All four GFMs perform well on this test, since the material properties are fairly similar across the interface.

- The OGFM does not correctly locate the interface due to the fact that level set advection is performed using real fluid velocities in this scheme, as opposed to the star-state velocity from the mixed Riemann problem. This suggests that the approach of using real fluid velocity for level set advection is inferior to using the interface extensions velocity field.

- The RGFM gets the shock and rarefaction positions slightly wrong. To be specific, they have both travelled one grid cell further than they should have. This is directly caused by the fact that the RGFM overwrites real states using information from the mixed Riemann problem, thereby allowing information to propagate across two grid cells in a single time step.

- The OGFM has the largest conservation error. The MGFM and PGFM have lower conservation errors than the RGFM, due to the initial offset caused by the RGFM overwriting real states.
- All four GFMs exhibit first order convergence of the L1 error norm as expected due to the discontinuities in the solution. The MGFM tends to be the most accurate, while the RGFM is the least accurate due to the offset shock positions.

4.3.2 Gas-water shock tube problem

This problem describes a strong shock in air incident on an air-water interface. This time there are large differences in material properties across the interface, making it a more difficult test of robustness. In this test, both materials are stiffened gases with $\gamma_1 = 1.4$, $p_{\infty 1} = 0$ and $\gamma_2 = 7.15$, $p_{\infty 2} = 3309$. The interface is initially located at $x = 0.5$, with material 1 to the left hand side of this. The initial states are:

$$\mathbf{W}_1 = (0.00596521, 911.8821, 100)^T ,$$
$$\mathbf{W}_2 = (1.0, 0, 1.0)^T .$$

The solution is output at time $t = 0.0007$. Plots of density and conservation errors for the methods which successfully ran this problem are shown in figure 4.3. Results from this test can also be found in Problem 2 in [40].

Remarks on Experiment 2

- The most significant finding from this test case is that the OGFM and PGFM both generate unphysical states after the first time step and fail. The superior robustness of the MGFM and RGFM is only apparent in problems involving a large change in material properties across the interface.

4.3.3 Oscillating water column problem

This problem describes a bubble of water in air, oscillating in a closed tube. The fluid quickly reaches an oscillating steady state, meaning that this test case is useful for measuring conservation errors over very long time periods. It may be argued that the conservation errors from this test are more useful than those computed in shock tube problems where

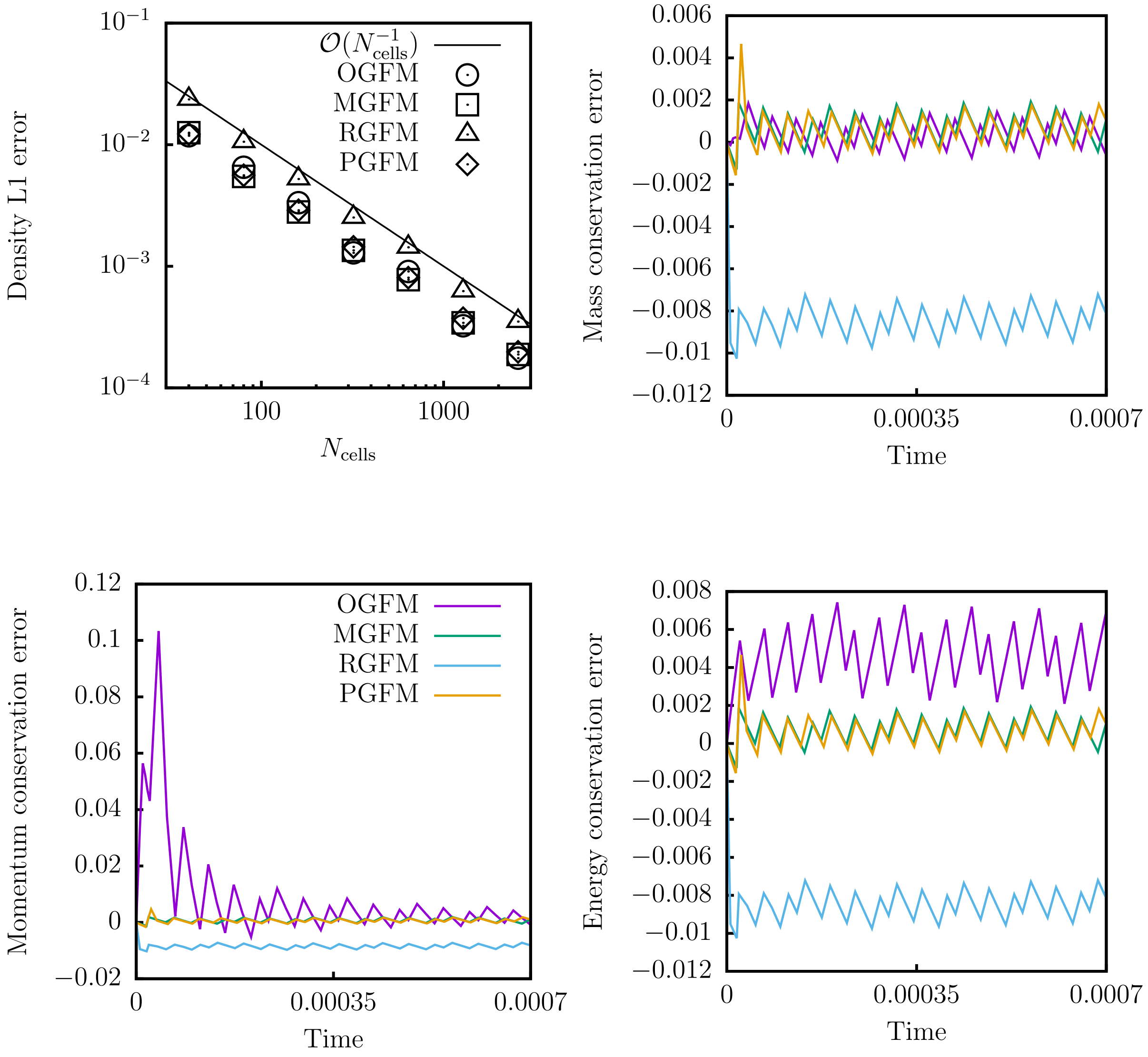


Figure 4.2: The top-left plot shows convergence of the 1-norm of the density error for various GFMs. The RGFM is noticeably less accurate due to the fact that the shock position is consistently offset from the true solution. The other three plots show the relative error in mass, momentum and energy conservation as a function of time on the $N = 80$ grid.

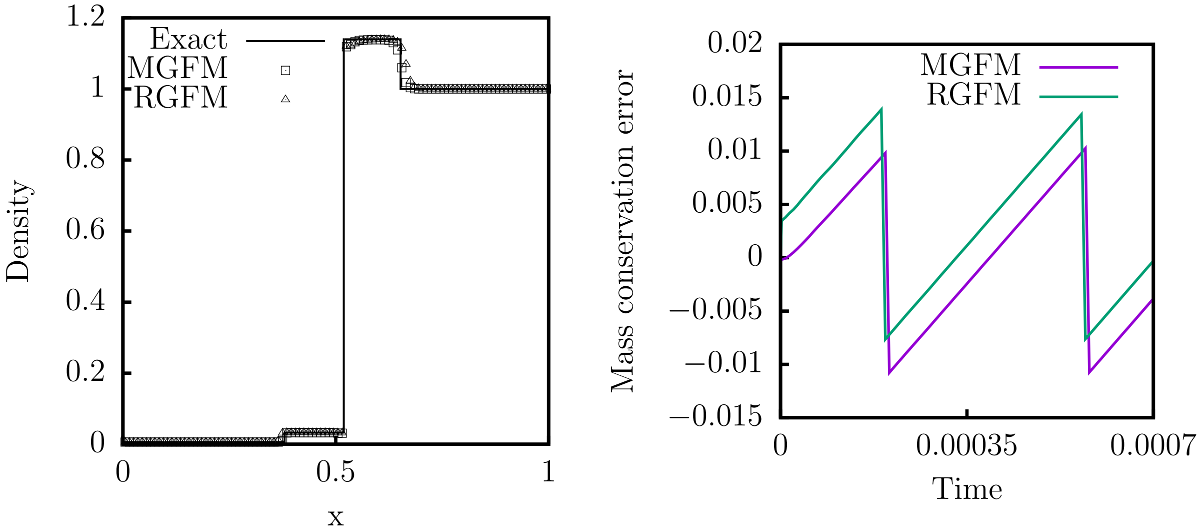


Figure 4.3: The left-hand plot shows the density solutions obtained at the end of the test. Note the slightly offset shock positions given by the RGFM. The right hand plot shows the relative mass conservation error of each method, which oscillated below 1%. Plots of momentum and energy conservation for this test are identical, and are not shown.

the pressure and velocity change across the material interface very quickly becomes zero, and then remain there.

In this test, both materials are stiffened gases with $\gamma_1 = 1.4$, $p_{\infty 1} = 0$ and $\gamma_2 = 7$, $p_{\infty 2} = 3000$. The domain is $[-1, 1]$, discretised by 80 cells. Reflective boundary conditions are applied to each end of the tube. The water (material 2) initially occupies the region $x \in [-0.1, 0.1]$, with air on either side. The initial states are:

$$\mathbf{W}_1 = (0.001, 1, 1)^T ,$$
$$\mathbf{W}_2 = (1, 1, 1)^T .$$

The simulation is run until time $t = 100$, which corresponds to approximately 7×10^5 time steps. A similar test case is shown in [40]. Plots of the total mass of air and water over $t \in [0, 100]$ are shown in Figure 4.4. In this case the masses are computed by estimating the volume fraction of the material in cell i, f_i, and then computing:

$$M(t) = \Delta x \sum_i f_i \rho_i(t). \tag{4.7}$$

Remarks on Experiment 3

- The OGFM and PGFM quickly generate unphysical vacuum states at the air-water interface and fails. The two other GFMs are able to run the test.

- The period of oscillation from the MGFM is almost 10% larger than the period

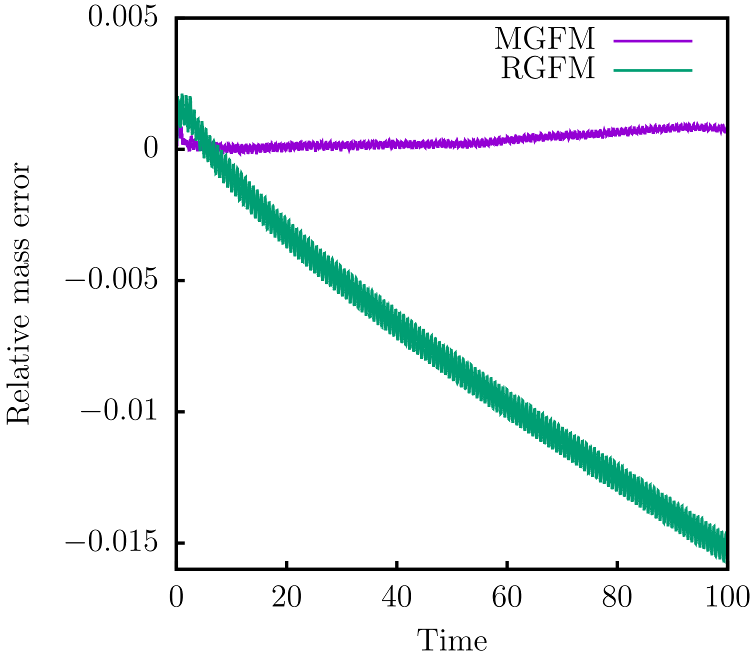


Figure 4.4: This plot shows the superior conservation properties of the mGFM over the rGFM over an extremely large ($> 10^5$) number of time steps. The oGFM and PGFM failed to run the test.

given by the RGFM. The period given by the RGFM is in better agreement with results in [40].

- The RGFM tends to lose mass over long time periods. The mass loss of air is significantly worse in the RGFM compared to the MGFM, which gives a slight gain in mass over the test. However, even after ∼140 oscillations and 7×10^5 time steps, the change in the total mass of air in the RGFM is less than 2%.

4.3.4 Conclusions from one-dimensional comparison

This concludes our comparison of four one-dimensional ghost fluid methods. We have shown that the original ghost fluid method consistently generates unphysical states and fails to run problems involving significant changes in material properties across the interface, such as the air-water test cases presented here. However, where both materials are ideal gases the original ghost fluid method performs suprisingly well, yielding similar L1 errors and conservation errors to the modified ghost fluid method. As a result, we will apply the original ghost fluid method to two-dimensional ideal gas problems later in the thesis. Another surprise was the poor performance of the real ghost fluid method; on both gas-gas and air-water problems the accuracy and conservation properties were inferior to the modified ghost fluid method. The practical ghost fluid method proved to be insufficiently robust on air-water problems.

The scope of these results are limited to the case in which the ghost fluid method is used alongside the MUSCL-Hancock flow solver. The performance of the various ghost fluid methods appears to depend heavily on the flow solver used. For example, although not mentioned in the literature, in numerical experiments not shown here we have found that the random choice method [119] along with the real ghost fluid method yields near-perfect solutions to shock tube problems. This is due to the random choice method providing the exact solution on either side of discontinuities, with only the position of the discontinuity being subject to an error which is inversely proportional to the number of sampling points tested. Unfortunately the random choice method has not been extended to multiple spatial dimensions.

4.4 Coupling between the ghost fluid method and the interface tracking method

In multi-material simulations with the ghost fluid method in two or more spatial dimensions, the interface tracking method becomes critically important in determining the overall flow solution. The solution is the result of a coupled system, with the evolution of the interface tracking method depending on the fluid state (specifically the fluid velocity), and the evolution of the fluid state depending on the interface position.

We use a simple splitting approach to advance the coupled system: in the update from time level n to time level $n + 1$, the ghost fluid method uses the level set field at time level n, and the level set field is updated using the fluid velocity at time level n. We have experimented with a level set update using the fluid velocity at the half-step time level $n + 1/2$ in order to make this splitting formally second order accurate, but found negligible change in the solutions. Some interface tracking methods which we apply, such as the CLSVOF method, require a velocity field which is defined throughout the domain rather than only at cell centres. We obtain these velocities through bilinear interpolation between the cell centred values.

4.5 The effect of interface tracking accuracy on ghost fluid method solutions

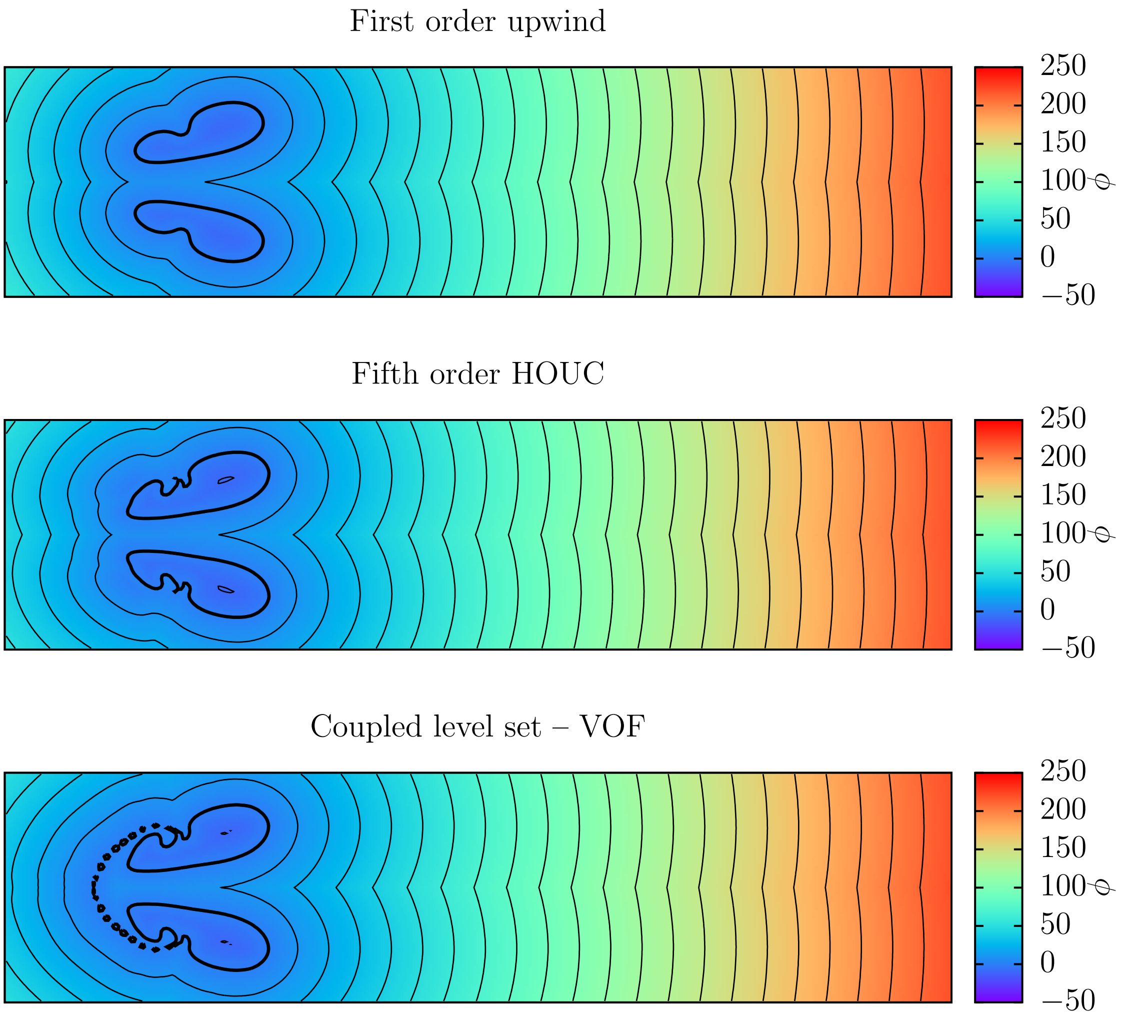
In this section we examine the effect of interface tracking method accuracy on the numerical solution obtained with a two-dimensional ghost fluid method. We choose a test problem in which a shock travelling at Mach 1.2 is incident on a circular bubble of helium in air, since the resulting shock-interface interaction results in complex interface geometry featuring a thin filamentary structure. This filamentary structure becomes progressively thinner, and is difficult to resolve with any Eulerian interface tracking method as the width falls below the cell width used to discretise this problem.

The computational domain is the region $[0, 325] \times [0, 89]$, with transmissive boundary conditions applied to the left and right hand sides, and reflective boundary conditions applied to the top and bottom. Both materials are modelled by the Euler equations with ideal gas equation of state, with air having $\gamma_1 = 1.4$ and helium having $\gamma_2 = 1.667$. The helium is initially contained in a circular region centred at $(175, 44.5)$ with radius 25, and the initial condition of the primitive variables in the ghost fluid method are:

$$\begin{aligned} x \geq 225 : \mathbf{W}_1 &= (1.3764, -0.394, 0, 1.5698)^T , \\ x < 225 : \mathbf{W}_1 &= (1, 0, 0, 1)^T , \\ \mathbf{W}_2 &= (0.138, 0, 0, 1)^T . \end{aligned}$$

All units are dimensionless. The original ghost fluid method with isobaric fix is used to run this test case, along with an unsplit MUSCL-Hancock method using an exact Riemann solver. The CFL number was set to 0.4, and the test case was run with a relatively coarse grid of 325×89. We have coupled the ghost fluid method to a wide variety of level set methods, with spatial orders of accuracy ranging from one to nine (methods of order 3 and above are advanced in time using the third-order Runge-Kutta method, otherwise the forward Euler method is applied) [54]. We also run this test case with the ghost fluid method coupled to the CLSVOF interface tracking method [55], which concurrently updates a grid of volume fractions and a level set field in order to prevent spurious mass loss due to smearing of the level set function. The use of this interface tracking method allows us to distinguish between mass change due to the two-valued flux in the ghost fluid method, and due to mass loss from the level set field. Conveniently, the ghost fluid method works as usual by accessing only the level set field. The relative change in helium bubble mass, as well as final level set field solutions are shown in figures 4.6 and 4.5.

These results firmly indicate that there is a positive relationship between the accuracy of the interface tracking method, and the mass conservation properties of the scheme. As expected, the CLSVOF method is superior to level set interface tracking methods, and figure 4.5 is very clear in demonstrating how CLSVOF maintains helium mass in a series of droplets after the filament becomes too thin to be resolvable on this grid. Given this behaviour, in the following section we propose a novel ghost fluid method which is able to use the efficient moment-of-fluid interface tracking method developed in chapter 3 of this thesis. This interface tracking method maintains the advantage of conservation which the CLSVOF method possesses, but was shown to be both computationally cheaper as well as more accurate.



First order upwind

Fifth order HOUC

Coupled level set – VOF

Figure 4.5: The level set field at $t = 280$, with the position of the zero level contour emphasised in bold, for various interface tracking methods coupled to the original ghost fluid method on the shocked helium bubble test.

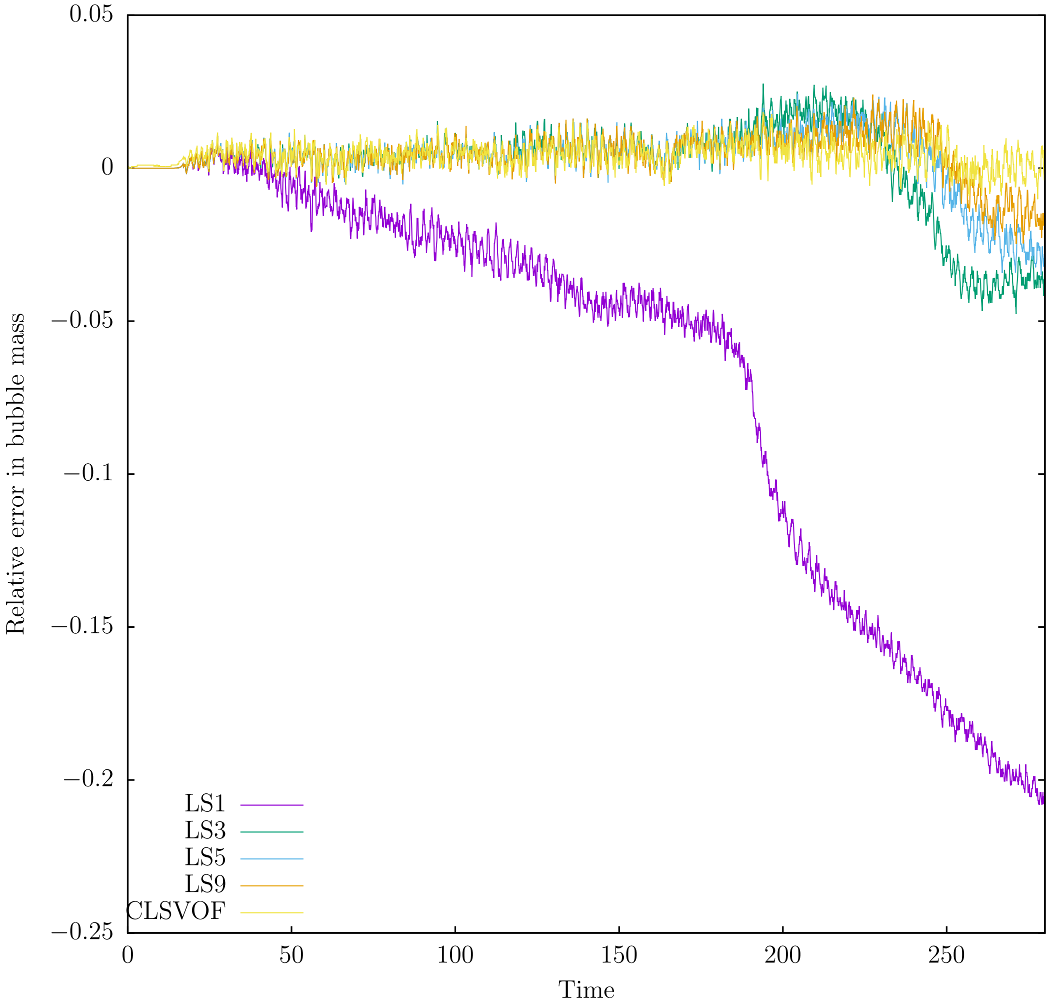


Figure 4.6: A comparison of the effect of the interface tracking method on the mass conservation of helium in the shocked helium bubble test. The “LSx” interface tracking methods refer to a level set method with “x”-order spatial accuracy, and the CLSVOF method is the coupled level set and VOF method [55].

4.6 A ghost fluid method with EMOF interface tracking

The algorithmic step which prevents the ghost fluid method from being coupled to a volume fraction-based interface tracking method without modification is the ghost state extrapolation step. The exact operation of this step varies depending on the type of ghost fluid method, but all involve solving the constant extrapolation partial differential equation (equation 4.1). This equation assumes knowledge of a signed distance function describing the interface, which we only possess in the level set and CLSVOF interface tracking methods. In order to apply the superior accuracy and conservation properties of the volume tracking methods developed in chapter 3, the basic ghost fluid algorithm must be adjusted. In this section we describe a modification to the ghost fluid method which populates ghost cell values without a global extrapolation step, and is then coupled to the volume fraction-based interface tracking methods developed in chapter 3.

A similar problem has been encountered in the context of adaptive mesh refinement with ghost fluid methods, where it was found that the pseudo-time marching in the extrapolation step was difficult to incorporate into the physical time advancement step [120]. The resulting extrapolation-free approach is known as characteristics-based matching (CBM). This is a dimensionally-split method which uses a level set to identify the interface, and then uses the solution to a mixed Riemann problem to set the fluxes in a band around the interface, the width of which depends on the order of the single material flow solver. Since no ghost cells are maintained, no extrapolation is required. Freshly-created cells have their states populated using a nearest-neighbours approach. Similarly to ghost fluid methods, there is a two-valued flux across the interface so the approach is non-conservative. Bo and Grove [121] recently presented a CBM method which was coupled to the LVIRA VOF interface tracking method, a natural extension to the original CBM method due to the lack of ghost cells. Although no ghost cells are maintained, this method features many similarities to the real ghost fluid method.

In this section we take a different approach to the problem, and maintain a narrow band of ghost cells around the interface which are populated through a simple inverse distance weighting procedure. This is straightforward to implement and computationally efficient, and works equally well in a dimensionally-split or unsplit framework.

For completeness we note the naive option of achieving this coupling by generating a signed distance function from the volume fractions after every time step, and then using the ghost fluid method as usual with this distance function. We have implemented a numerical scheme in which this function is generated efficiently by first computing the signed distance in a narrow band around the interface. The problem of populating values in the remainder of the domain can then be cast as a boundary-value problem for the

Eikonal equation, which is solved extremely quickly using the fast sweeping method [105]. Results obtained using this approach are very similar, but our adjusted ghost fluid method is preferred due to its superior efficiency and ease of implementation.

4.6.1 Numerical method

We now describe our numerical method. Note that this approach can be applied with any single material flow solver – the important factor is the width of ghost cells which must be maintained around the interface, denoted by s. For the MUSCL-Hancock method we have $s = 2$. The approach is based on the original ghost fluid method, in which the pressure and velocity are set equal to the real fluid velocity, and the density is set by extrapolating entropy across the interface. Other than the novel procedure used to set ghost states, this scheme operates identically to existing ghost fluid methods – the standard procedure to update both grids is followed:

1. Choose the largest stable time step.

2. Set the ghost states using the procedure described in the following section which avoids a global extrapolation operation.

3. Advance both grids using any single-material solver.

4. Update the volume fractions using a VOF or MOF interface tracking method.

4.6.1.1 Setting ghost states

The ghost states are set through two loops over all cells. In the first loop we store the real fluid density and pressure on either side of the interface. The (i, j)-th cell can easily be identified as mixed by checking if $0 < z_{i,j} < 1$, and in every mixed cell we send out probes normal to the interface. From the volume-of-fluid interface reconstruction, we obtain the interface mid-point and unit normal vectors, denoted by $\mathbf{x}_{i,j}$ and $\mathbf{n}_{i,j}$. The real fluid density and pressure are then computed through bilinear interpolation of surrounding states at positions:

$$\mathbf{x}_{i,j} \pm \Delta \mathbf{n}_{i,j}, \tag{4.8}$$

where we set the probe length, Δ to $\min(\Delta x, \Delta y)$. These two quantities are stored in the vectors $S_{i,j}^{(1)}$ and $S_{i,j}^{(2)}$ for later retrieval.

In the second loop the ghost fluid values are set. Ghost states are easily set in mixed cells. Assuming that $z_1 > \frac{1}{2}$, then fluid 2 is the ghost fluid in this cell. The ghost velocity and pressure are set equal to the real values:

$$\begin{aligned} u_{i,j}^{(2)} &= u_{i,j}^{(1)}, \\ v_{i,j}^{(2)} &= v_{i,j}^{(1)}, \\ p_{i,j}^{(2)} &= p_{i,j}^{(1)}. \end{aligned}$$

The ghost fluid density is set by extrapolating the entropy from the reference values in $S_{i,j}^{(2)}$:

$$S_{i,j}^{(2)} = \begin{pmatrix} \rho_{\text{ref}} \\ p_{\text{ref}} \end{pmatrix}. \tag{4.9}$$

With the stiffened gas equation of state, the isentropic extrapolation gives the new density as:

$$\rho_{i,j}^{(2)} = \rho_{i,j}^{(2)} \left(S_{i,j}^{(2)}, p_{i,j}^{(1)} \right) = \rho_{\text{ref}} \left(\frac{p_{i,j}^{(1)} + p_\infty}{p_{\text{ref}} + p_\infty} \right)^{\frac{1}{\gamma}}. \tag{4.10}$$

The conserved variables are then regained from the primitives set above. In empty or full cells which also require ghost states, we apply an inverse distance-weighting procedure to set the reference density and pressure for isentropic extrapolation. These reference values are stored in a vector denoted by S_{invdist}. The procedure for cell (i, j) for the $z_{i,j} = 1$ case is described in algorithm 3. The ghost fluid state is then set as before, copying across the pressure and velocity from the real fluid 1 state in the cell, and setting the density in the same manner as equation 4.10:

$$\rho_{i,j}^{(2)} = \rho_{i,j}^{(2)} \left(S_{\text{invdist}}, p_{i,j}^{(1)} \right). \tag{4.11}$$

The $z_{i,j} = 0$ case follows similarly.

4.6.2 Comparing level set and VOF interface tracking methods

We now present numerical results obtained using our novel ghost fluid method (which is based on the OGFM) coupled to the efficient moment-of-fluid (EMOF) interface tracking method presented in chapter 3. For comparison, we have implemented the original ghost fluid method with isobaric fix, coupled to three existing interface tracking methods which evolve a level set field:

- The level set method with first order upwind spatial discretisation and forward Euler time stepping, denoted by LS1 [54],

- The level set method with fifth order upwind-central spatial discretisation and third order Runge-Kutta time stepping, denoted by LS5 [54],

Algorithm 3: Computing the reference density and pressure used for isentropic extrapolation in cell (i, j). The vector $\mathbf{c}_{i,j}$ is the position of the (i, j)-th cell centre. The position and velocity are stored in the vector S_{invdist}.

```
$S_{\text{invdist}} \leftarrow 0$ ;
$d \leftarrow 0$ ;
for $k = -s, -s+1, ..., s$ do
    for $l = -s, -s+1, ..., s$ do
        if $0 < z_{i+k,j+l} < 1$ then
            $m \leftarrow \frac{1}{|\mathbf{c}_{i+k,j+l} - \mathbf{c}_{i,j}|}$ ;
            $d \leftarrow d + m$ ;
            $S_{\text{invdist}} \leftarrow S_{\text{invdist}} + mS^{(2)}_{i+k,j+l}$ ;
    end
end
$S_{\text{invdist}} \leftarrow \frac{S_{\text{invdist}}}{d}$ ;
```

- The coupled level set and VOF method, denoted by CLSVOF [55].

We compare conservation, accuracy, and computational efficiency over a range of two dimensional test cases. The aim of the numerical tests presented here is to investigate the relative effects of each interface tracking method, rather than the performance of the ghost fluid method itself. See [39, 111, 113] for detailed investigations of the performance of the ghost fluid method.

One dimensional simulations are not considered since the problem of interface tracking becomes trivial in this case and so all methods perform identically. All schemes are implemented in serial in the same piece of software, and share as many subroutines as possible. The single fluid update step is carried out using the unsplit MUSCL-Hancock method with primitive-variable reconstruction and minbee limiting, as described in section 2.3. All simulations are run using a CFL number of 0.4, which is a conservative choice based on the stability of the unsplit MUSCL-Hancock method. We use an exact Riemann solver, as described in appendix A. Unless otherwise mentioned, all units are dimenionless. All initial conditions are presented in terms of the vector of primitive variables for each fluid: $\mathbf{W} = (\rho, u, v, p)^T$.

4.6.2.1 Three-state two-material problem

This problem simulates the evolution of a vortex at the triple point. The resulting thin filamentary structure highlights the differences between interface tracking methods, and allows for meaningful qualitative comparison. Results from this test case obtained using mixture model methods are also shown in section 5.3.3. The computational domain is $[0, 7] \times [0, 3]$ with transmissive boundary conditions. Both materials are ideal gases, with material 1 having $\gamma_1 = 1.5$ and material 2 having $\gamma_2 = 1.4$. Initially, material 2 occupies the rectangular region $[1, 7] \times [0, 1.5]$, with material 1 elsewhere. The initial states are:

$$\begin{aligned} x < 1 : \mathbf{W}_1 &= (1, 0, 0, 1)^T \,, \\ x \geq 1 : \mathbf{W}_1 &= (0.125, 0, 0, 0.1)^T \,, \\ \mathbf{W}_2 &= (1, 0, 0, 0.1)^T \,. \end{aligned}$$

Solutions on a 210×90 grid are shown in figures 4.7 and 4.8. Figure 4.7 demonstrates the sharp density change and the lack of pressure oscillations at the interface with the novel ghost fluid method. Figure 4.8 compares the interface representation at various times between the novel ghost fluid method with EMOF interface tracking, and the original ghost fluid method with various interface tracking methods. The improvement in moving from first- to fifth-order level set methods is immediatly apparent. The CLSVOF and EMOF methods both offer superior interface resolution to the fifth-order level set method, with the EMOF method slightly better able to maintain the filamentary structures. A comparison of the relative time spent running this test is provided in table 4.1, showing that eliminating the solution of the extrapolation equation from the algorithm results in an approximately 25% decrease in the run time in the high resolution case. In a parallel implementation of the method this figure is expected to be even better, since setting ghost states is a major source of inter-process communication.

4.6.2.2 Shocked helium bubble

We now examine the case of a Mach 1.22 shock wave in air incident on a circular bubble of helium. The initial conditions for this test case are described in section 4.5. The relationship between the conservation properties of each method and the computational run time is quantified by running this test case at three resolutions: 325×89, 650×178, and 1300×356. In each case the run time and mass error in the helium bubble are measured. Since the helium mass error oscillates significantly between time steps, we measure the L_1-norm of this error over the final 50 time units of the simulation:

$$E_1 = \frac{1}{50} \int_{230}^{280} |M_{\mathrm{He}}(t) - M_{\mathrm{He}}(0)| \; \mathrm{d}t, \tag{4.12}$$

where $M_{\mathrm{He}}(t)$ is the total mass of helium in the domain at time t. The plot of conservation error against runtime for all four methods is presented in figure 4.9. This shows an order of magnitude improvement in the conservation error from LS5 over LS1, and another order of magnitude improvement from the volume fraction-based methods over LS5. The improvement of EMOF over CLSVOF is much less dramatic; this is not suprising since both interface tracking methods conserve mass. Nevertheless, for a given conservation error, the novel ghost fluid method with EMOF interface tracking method offers the

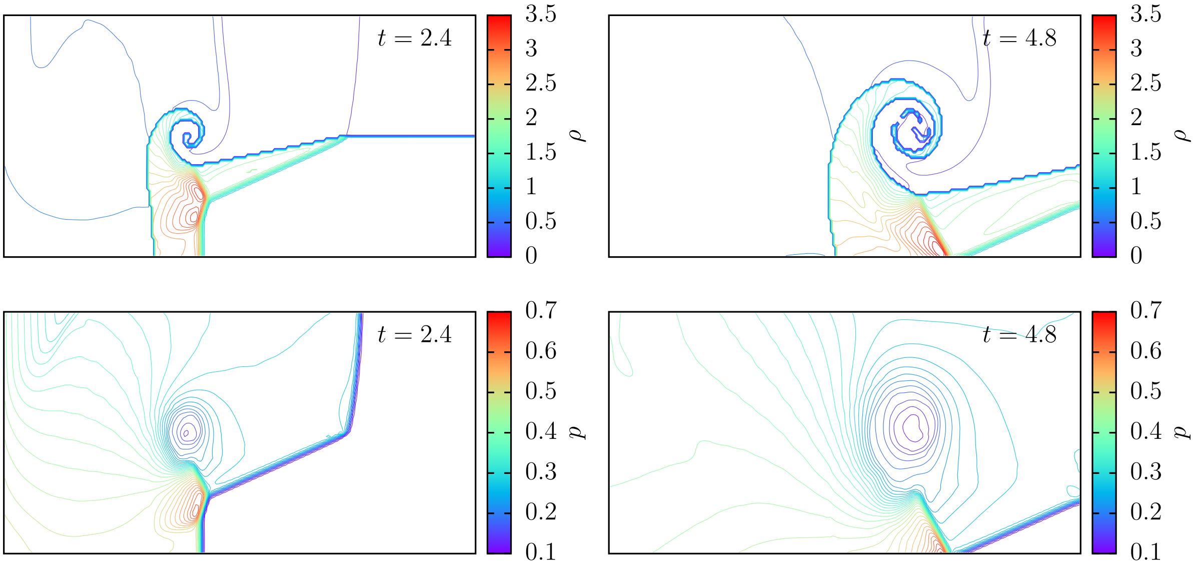


Figure 4.7: Numerical results for the three-state two-material problem on a 210 × 90 grid obtained with the novel ghost fluid method and EMOF interface tracking.

	Resolution	
Method	210 × 90	840 × 360
Novel GFM with EMOF	0.85	0.77
Original GFM with CLSVOF	1.0	1.0
Original GFM with first-order levelset	0.87	0.89
Original GFM with fifth-order levelset	0.89	0.94

Table 4.1: Relative run-time for the three-state two-material problem.

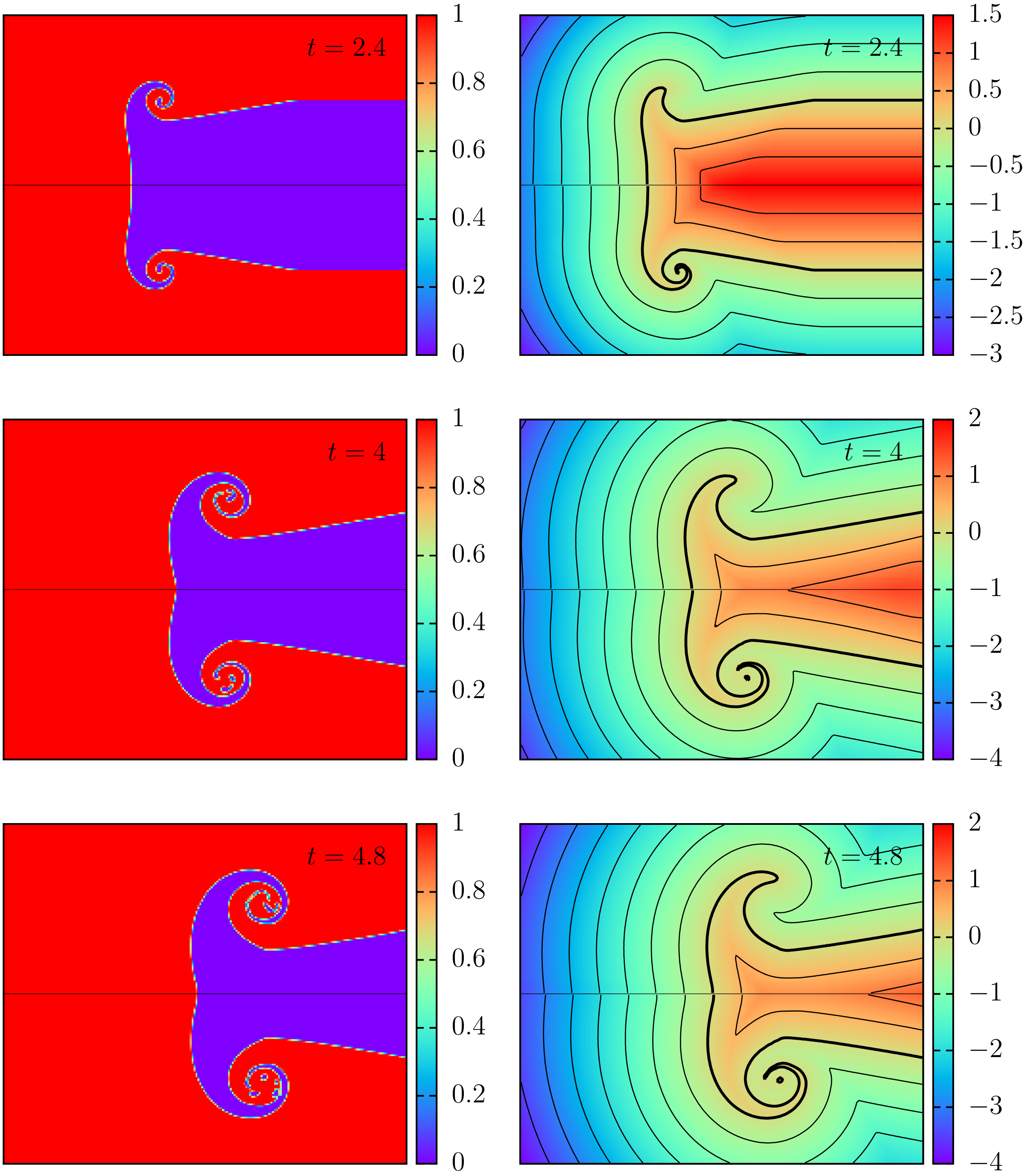


Figure 4.8: A comparison of four different interface representations on the three-state two-material test case with 210 × 90 grid cells. The left-hand column shows the volume fraction solutions obtained with the EMOF method (top half), and CLSVOF method (bottom half). The right hand column shows the level set field with overlaid contours obtained with a first order accurate discretisation (top half), and fifth-order in space with third order in time discretisation (bottom half).

smallest runtime.

4.6.2.3 Shocked R22 bubble

In the second shocked bubble experiment which we consider, a circular bubble of the refrigerant R22 undergoes collision with a planar shock wave. The computational domain is the region $[0, 445] \times [0, 89]$, with transmissive boundary conditions applied to the left and right hand sides, and reflective boundary conditions applied to the top and bottom. Both materials are modelled by the ideal gas equation of state, with air having $\gamma_1 = 1.4$ and R22 having $\gamma_2 = 1.249$. The R22 is initially contained in a circular region centred at $(0.225, 0.0445)$ with radius 0.025. The initial primitive variables are:

$$\begin{aligned} x \geq 0.275 : \mathbf{W}_1 &= \left(1.686, -113.5, 0, 1.59 \times 10^5\right)^T, \\ x < 0.275 : \mathbf{W}_1 &= \left(1.225, 0, 0, 1.01325 \times 10^5\right)^T, \\ \mathbf{W}_2 &= \left(3.863, 0, 0, 1.01325 \times 10^5\right)^T. \end{aligned}$$

This test was run with all three combinations of GFM and interface tracking method. The first set of results we present is a numerical Schlieren plot of density obtained with the novel ghost fluid method with EMOF interface tracking. These plots are displayed in figure 4.10. Rather than reproduce this plot two more times, in figures 4.11, 4.12 and 4.13, we plot the bubble edge as defined by the interface tracking method in each approach (i.e. the $z = 0.5$ contour in the EMOF and CLSVOF methods, and the $\phi = 0$ contour in the level set method). These plots better illuminate the differences between each method since away from the interface, the density solutions are similar.

First, the results in figure 4.10 obtained with the novel GFM and EMOF interface tracking method demonstrate the ability of this original method to handle complex shock-interface interactions. In the interface-only plots, the interface geometry obtained with the CLSVOF interface tracking method is qualitatively very similar to that obtained with the EMOF method despite the clear superiority of the EMOF method on the interface advection test cases in the previous chapter, and the fact that the EMOF method was better able to represent thin filaments on the three-state two-material problem. It appears that the overall solution is relatively insensitive to the precise accuracy of the interface tracking method, perhaps because the ghost fluid method (which is first order accurate at the interface) is the source of most of the solution error. However, both the EMOF and CLSVOF methods give better interface resolution than the fifth-order level set used in figure 4.13. This suggests that it is more important to use a conservative interface tracking method than to use the scheme with the highest formal accuracy.

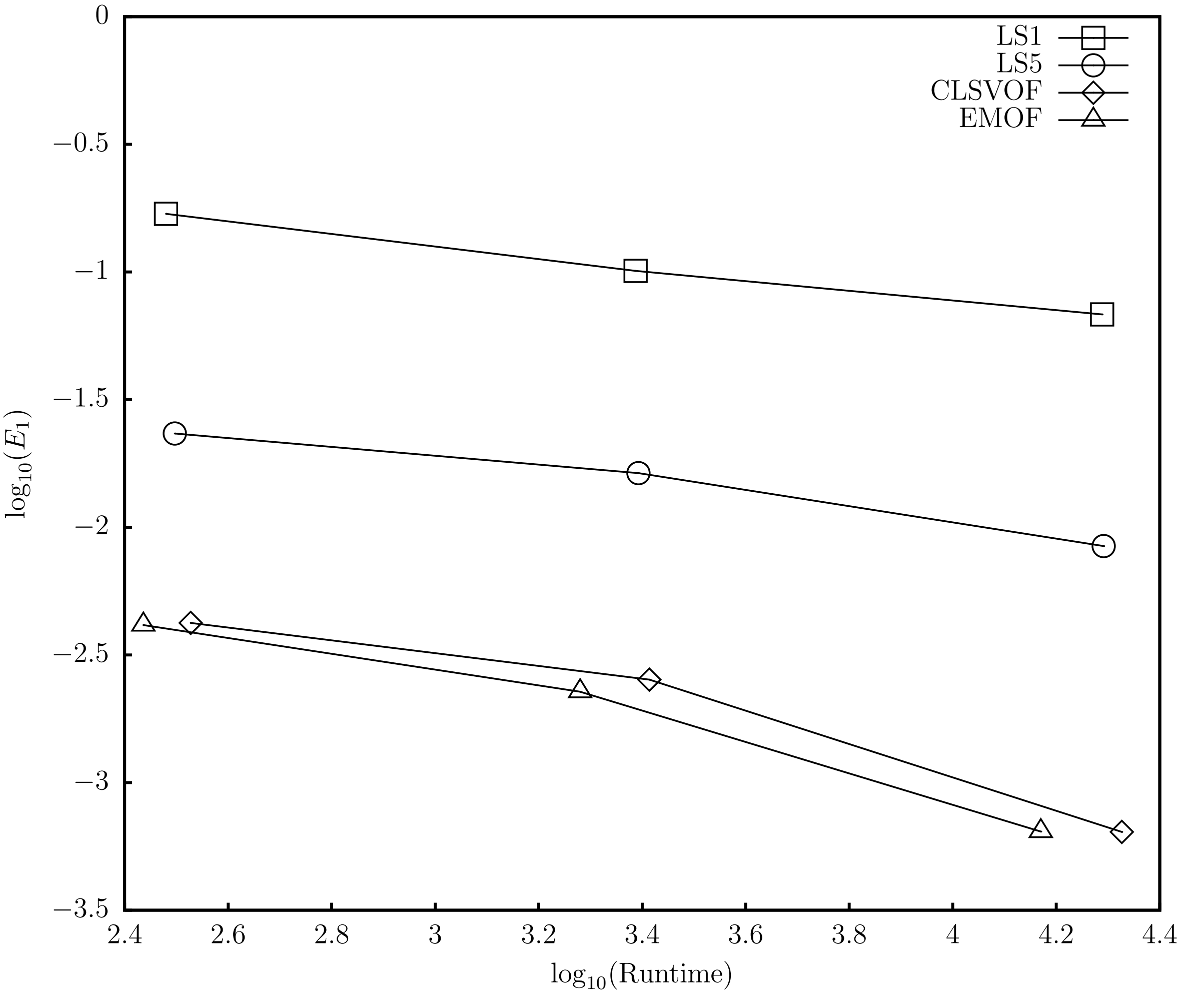


Figure 4.9: Helium mass error versus runtime on the shocked helium bubble test case. See equation 4.12 for the full definition of E_1, the final helium mass error. This demonstrates that the EMOF method is the most computationally-efficient interface tracking method to obtain a given conservation error.

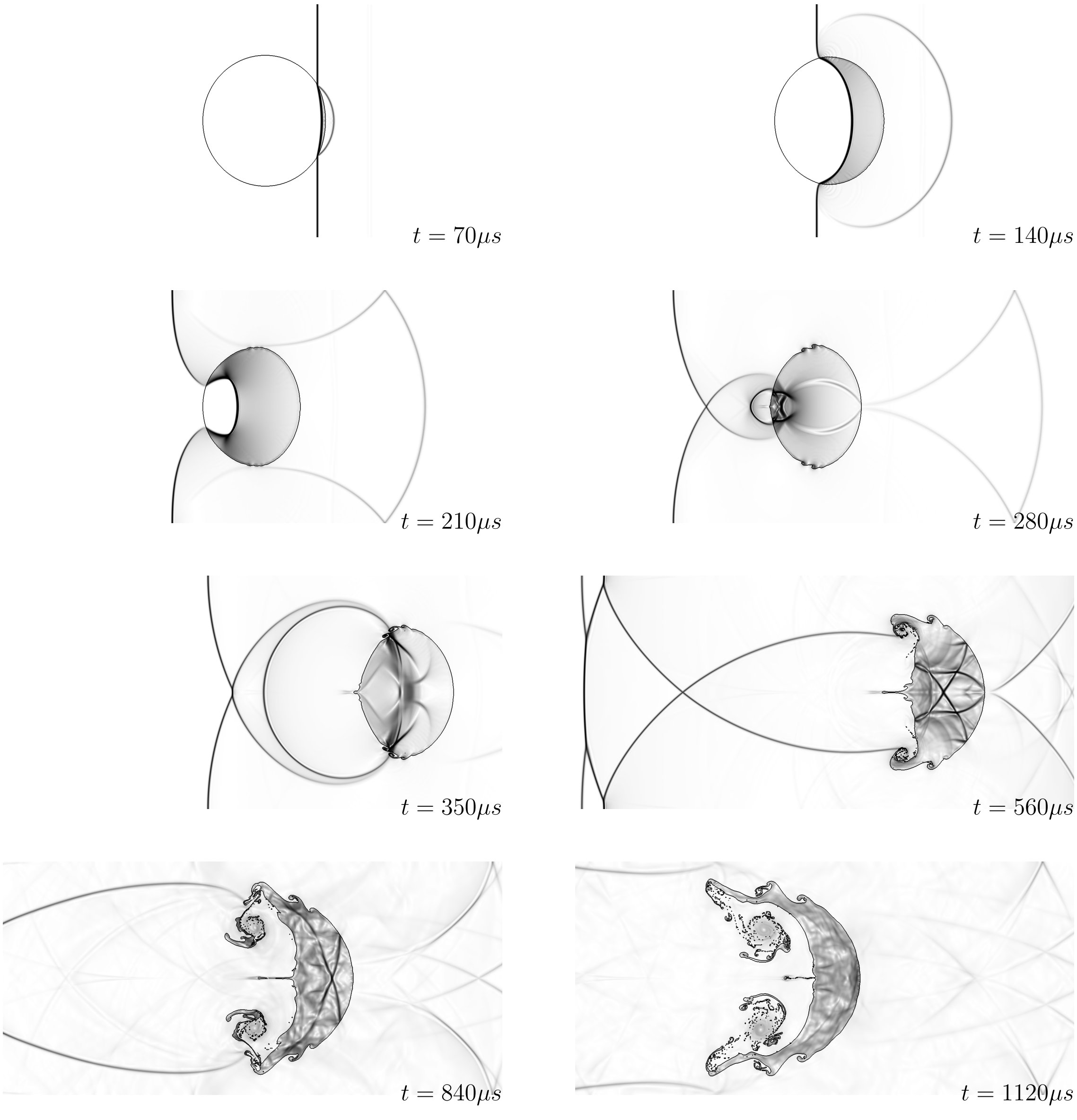


Figure 4.10: Numerical results for the shocked R22 bubble test on a 3560×712 grid. Each image is a numerical Schlieren plot of density, obtained using the novel ghost fluid method with EMOF interface tracking.

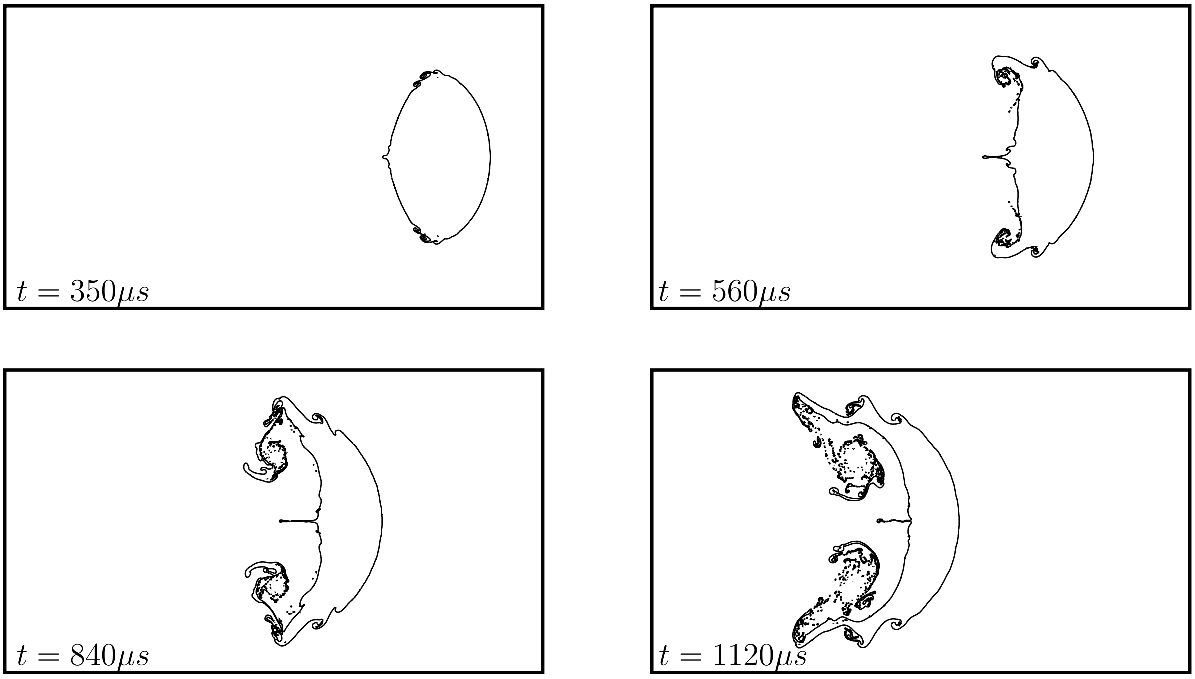


Figure 4.11: Numerical results for the shocked R22 bubble test on a 3560 × 712 grid, using the novel GFM with EMOF interface tracking. The $z = 0.5$ contour is plotted.

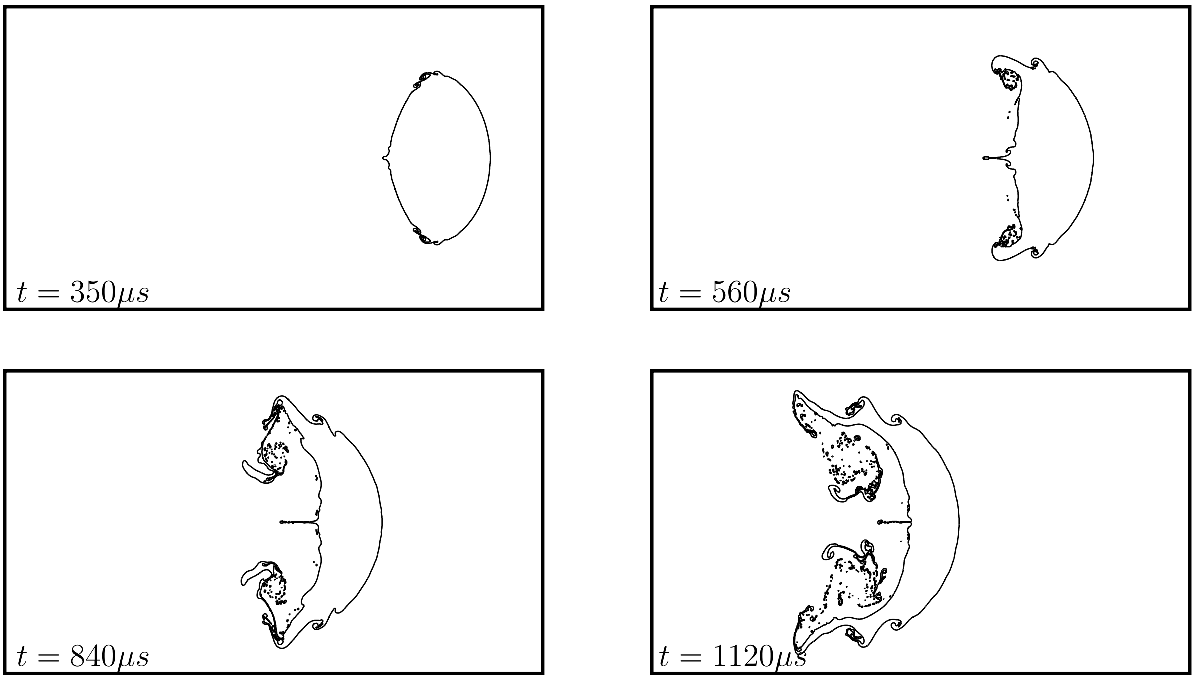


Figure 4.12: Numerical results for the shocked R22 bubble test on a 3560 × 712 grid, using the original GFM with CLSVOF interface tracking. The $z = 0.5$ contour is plotted.

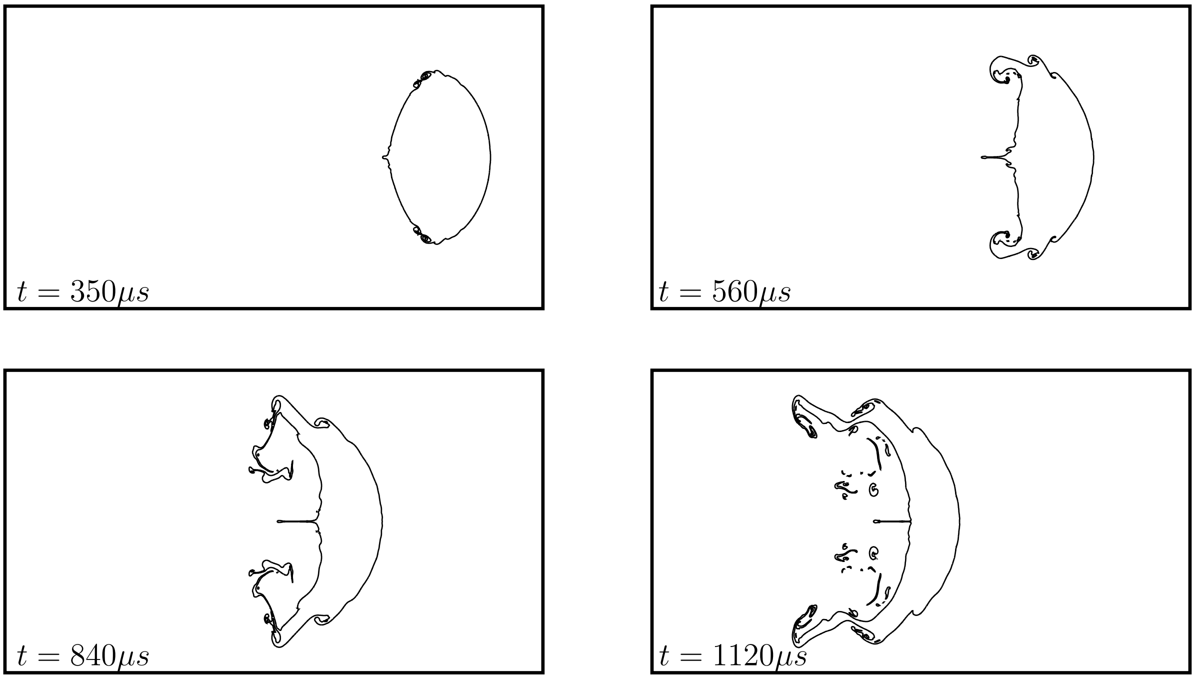


Figure 4.13: Numerical results for the shocked R22 bubble test on a 3560 × 712 grid, using the original GFM with fifth-order level set interface tracking. The $\phi = 0$ contour is plotted.

4.6.3 Numerical surface tension of the EMOF interface tracking method

One of the known shortcomings of the VOF interface tracking method is the numerical surface tension which the method carries. In this section we demonstrate that the EMOF method does not exhibit the same effect. The key to this demonstration is the application of a different kinematic boundary condition at the material interface– instead of the 'no-slip' boundary condition used previously, we modify the ghost fluid method to apply the 'slip' boundary condition.

The 'slip' boundary condition was touched on in section 1.1.2, and states that only the component of velocity normal to the interface must be continuous across the interface– the tangential component is allowed to jump. In a traditional ghost fluid method this boundary condition is implemented by extrapolating the tangential component of velocity away from the interface and into the ghost cell layer. In the context of the novel ghost fluid method where we do not have access to the global extrapolation equation, this boundary condition is implemented by using the same inverse distance weighting procedure which was used in order to extrapolate entropy into the ghost cells, detailed in algorithm 3.

In many cases a jump in tangential velocity at the interface is not stable [38], but the resulting Kelvin-Helmholtz instability is not well-posed for the Euler equations in the absence of surface tension, or some other regularising force (viscosity in the context of the Navier-Stokes equations) which prevents infinitesimally-small perturbations from

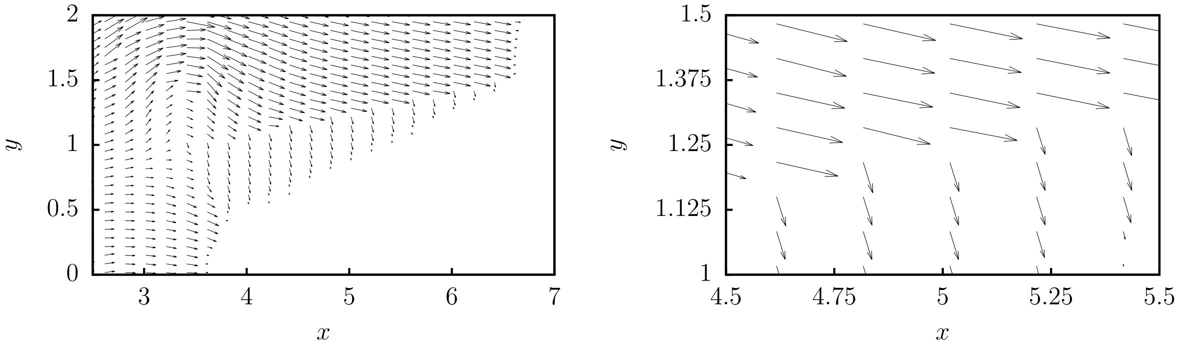


Figure 4.14: The velocity field obtained at time $t = 3.2$ on the three-state two-material problem with 'slip' interfacial boundary condition, showing the tangential velocity jump at the interface which later gives rise to an ill-posed Kelvin-Helmholtz instability. The right-hand side shows a zoomed view.

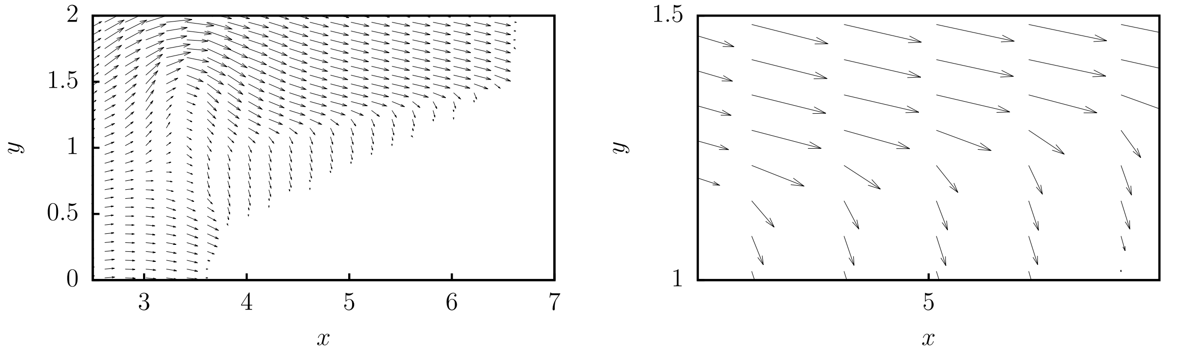


Figure 4.15: The velocity field obtained at time $t = 3.2$ on the three-state two-material problem with 'no-slip' interfacial boundary condition. The right-hand side shows a zoomed view.

transitioning to macroscopic scales. In numerical simulations, this can result in solutions being highly dependent on the unphysical viscosity and surface tension introduced by the numerical methods, and may prevent convergence of the solutions. In this section we present simulations with both the 'slip' and the 'no-slip' boundary conditions, and examine the effect of the numerical surface tension of two interface tracking methods on the solutions: the EMOF method, and Parker and Youngs VOF method.

The following numerical results are obtained from the three-state two-material test which was fully described in section 4.6.2.1. Other than the new boundary condition, all other simulation parameters are identical. Figure 4.15 shows the velocity field immediately after passage of the shock wave across the domain, for both interfacial boundary conditions, demonstrating the tangential velocity shear across the interface when the 'slip' boundary condition is applied. Figures 4.16 and 4.17 compare the interface, which is approximated by the $z = 0.5$ contour in this case, for all four possible configurations (each interface tracking method with each interfacial boundary condition) at two time levels.

These plots clearly show that the Kelvin-Helmholtz instability at the material interface is only apparent when the EMOF interface tracking method is applied. This demonstrates that this interface tracking method carries significantly less numerical surface tension than Parker and Youngs' method, which almost completely damps out the instability. Another contributing factor is that the small errors in the interface normal angle computed by the EMOF method may seed the instability, whereas Parker and Youngs' method computes a more smoothly-varying interface normal angle.

4.7 Conclusions

In this chapter we have studied several aspects of the ghost fluid method. In a one-dimensional comparison study we found that the original ghost fluid method exhibited surprisingly low solution errors relative to more modern variants of the method, but was much less robust. We then investigated the effect of the interface tracking method on two-dimensional solutions obtained with the original ghost fluid method. Using the shocked helium bubble test problem, we found that the choice of interface tracking method has a dramatic effect on the solution accuracy and conservation error– in particular that the use of a conservative interface tracking method (the coupled level set-VOF method) resulted in a significant improvement in the conservation properties of the scheme over even the ninth-order accurate level set method. This relationship was explored in more detail through the development of a novel ghost fluid method, which permitted the VOF-type interface tracking methods from the previous chapter to be coupled. We confirmed that there is a significant improvement in the accuracy and conservation properties of the solutions obtained with VOF-type interface tracking methods over level set methods. The

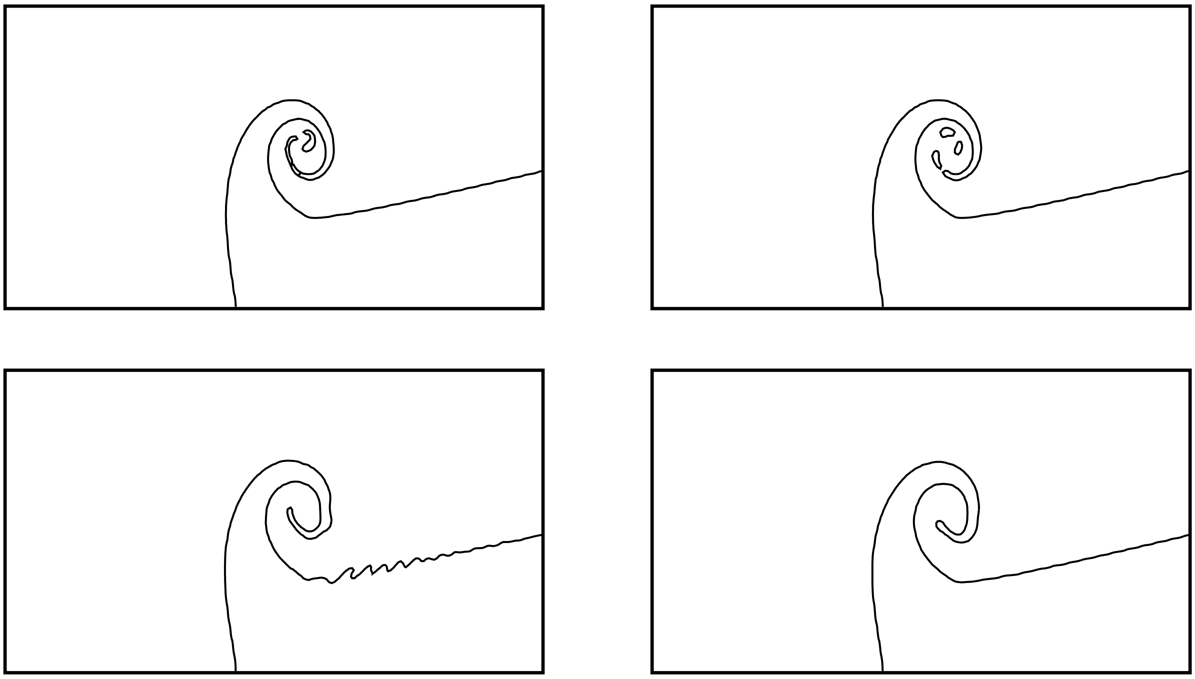


Figure 4.16: The $z = 0.5$ contour at time $t = 4.0$ on the three-state two-material test. Results in the left column are obtained with the EMOF interface tracking method, while in the right-hand column Parker and Youngs' method is used. In the top row, the 'no-slip' interface boundary condition is used, while the bottom row uses the 'slip' condition.

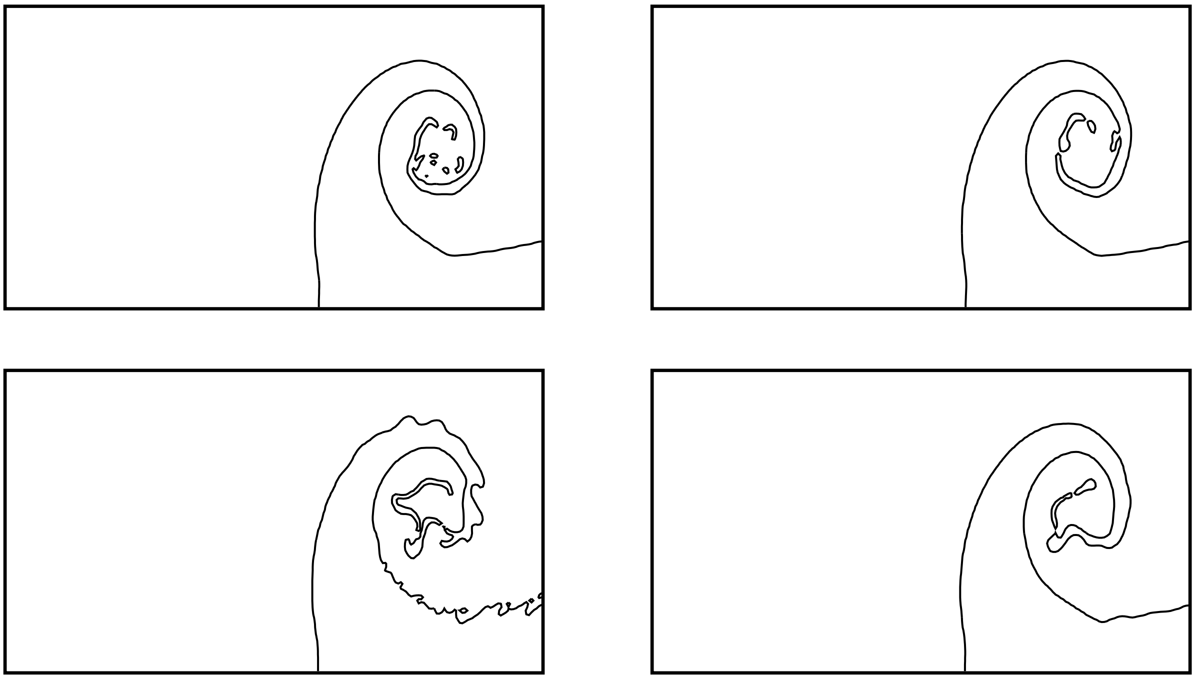


Figure 4.17: The $z = 0.5$ contour at time $t = 6.4$ on the three-state two-material test. Results in the left column are obtained with the EMOF interface tracking method, while in the right-hand column Parker and Youngs' method is used. In the top row, the 'no-slip' interface boundary condition is used, while the bottom row uses the 'slip' condition.

novel ghost fluid may also be of interest to researchers who are aiming to improve the parallel performance of their ghost fluid method code. Finally, we demonstrated that the EMOF interface tracking method proposed in chapter 3 exhibits lower numerical surface tension than Parker and Young's method.

CHAPTER 5

A volume-of-fluid based numerical method for compressible two-material flow

This chapter considers a second approach to the problem of two-material flow governed by the Euler equations. We now focus on the diffuse interface approach, and in particular the five-equation model originally described in [27–29, 57]. This system is equivalent to the full Baer-Nunziato seven-equation system [26] in the limit of instant velocity and pressure equilibration relative to characteristic flow timescales. Unlike the ghost fluid method, in the diffuse interface approach a single set of equations is solved on a single finite volume grid, and the multi-material nature of the flow is incorporated through adjusting the governing equations.

The five-equation system may be shown to be hyperbolic under some very basic assumptions [28], and can be solved using standard finite volume methods usually applied to the single-material Euler equations, although care is needed to ensure that volume fraction update is consistent with the update applied to the two mass conservation equations, to prevent incorrect material densities being recovered from these equations.

The material interface is represented as a contact discontinuity in the density, internal energy, and volume fraction. As the system evolves, this contact discontinuity diffuses over a number of computational cells– and since contacts lack the “self-sharpening” nature of shock waves the interface will continue to diffuse over more and more cells as the simulation runs. The extent of this diffusion is solely a function of the numerical solution method, and is not physical. Excessive diffusion of the interface is a particular problem when using coarse resolutions or large times. There are a number of existing approaches in the literature which aim to rectify the interface diffusion problem:

- The interface compression approach of [33] modifies the governing equations with a source term such that the interface remains smeared over a small, pre-set number of cells. This approach is compatible with a general Godunov-type method for the five-equation system, but does not sharpen the interface as effectively as other approaches.

- The anti-diffusion approach presented by So, Hu and Adams [32] takes the form of a post-processing operation applied to the volume fractions after every time step. The volume fractions are evolved according to a diffusion equation with a negative diffusion coefficient, and the other flow variables are then corrected appropriately in order to remain consistent with the sharpened volume fractions. This approach retains conservation, but is unable to maintain a completely sharp interface, and is not fully general as it is reliant on matching the strength of the anti-diffusion operator to the diffusion inherent in the particular numerical scheme chosen.

- In [122] it is shown that through the use of a limited downwind volume fraction flux, the Lagrange-remap numerical scheme maintains a sharp interface without the use of an explicit interface reconstruction procedure. A further advantage of this approach is the applicability to a broad range of equations of state.

- The THINC (Tanget of Hyperbola for INterface Capturing) scheme is a method for efficiently and simply advecting volume fractions, introduced in [83]. In subsequent work [34], the same sub-grid hyperbolic tangent reconstruction is applied to model the other physical variables in the five-equation system, along with any standard MUSCL or WENO reconstruction away from the interface. The system can then evolved using a semi-discrete wave propagation method [123] without introducing excessive smearing of the material interface. This approach has the advantage that the THINC interface tracking method is simple to implement, since it avoids the need for geometrical interface reconstructions. However, it remains less accurate than other contemporary interface tracking methods such as moment-of-fluid. The THINC approach has very recently been extended to Godunov-type numerical schemes in [124].

The numerical scheme presented in this chapter is our contribution towards this effort, and offers a fifth approach towards preventing the unphysical diffusion of the material interface.

This approach is based on the VOF method for compressible two-material flow presented by Miller and Puckett over 20 year ago [35]. To make the terminology clearer- the VOF interface tracking method simply represents and tracks a material interface, while Miller and Puckett's VOF method for two-material flow is a numerical method for solving the fluid dynamics of a two-material system, which is so-named because it incorporates a VOF interface tracking method. Miller and Puckett's approach is widely-cited but has not been directly developed, despite a number of advantages, including maintaining a sharp interface indefinitely, working with any Godunov-type single material solver, and correctly accounting for the relative compressibility of each material. We speculate that perhaps the original presentation, where long discretisation formulae are given without theoretical justification, was offputting. Miller and Puckett's VOF method appears in a one-dimensional comparison of multi-material methods in [125], and more importantly, the governing equations underpinning Miller and Puckett's discretisation were elucidated by Saurel et al. in [126], in the presentation of a diffuse-interface method based on the same system of equations.

Our numerical method is based on Miller and Puckett's approach, and maintains the advantages noted above. This method has been overhauled in numerous areas, the most important of which is the introduction of ideas developed for the solution of five-equation systems [28].

5.1 Five-equation model

A large class of two-material problems may be well-approximated by the five-equation model [27–29]. In this section we describe this model, and the diffuse-interface numerical method we use to solve it.

5.1.1 Governing equations

The five equation model describes conservation of individual material densities, mixture momentum, and mixture energy, as well as the advection of a material type indicator function, z. In one spatial dimension, the system can be written as:

$$\frac{\partial(z^{(1)}\rho^{(1)})}{\partial t}+\frac{\partial(z^{(1)}\rho^{(1)}u)}{\partial x}=0, \tag{5.1a}$$

$$\frac{\partial(z^{(2)}\rho^{(2)})}{\partial t}+\frac{\partial(z^{(2)}\rho^{(2)}u)}{\partial x}=0, \tag{5.1b}$$

$$\frac{\partial(\rho u)}{\partial t}+\frac{\partial(\rho u^2+p)}{\partial x}=0, \tag{5.1c}$$

$$\frac{\partial E}{\partial t}+\frac{\partial(u(E+p))}{\partial x}=0, \tag{5.1d}$$

$$\frac{\partial z^{(1)}}{\partial t}+u\frac{\partial z^{(1)}}{\partial x}=0, \tag{5.1e}$$

where $z^{(2)}=1-z^{(1)}$ and we define the following mixture quantities:

$$\rho=\rho^{(1)}z^{(1)}+\rho^{(2)}z^{(2)}, \quad \rho e=\rho^{(1)}z^{(1)}e^{(1)}+\rho^{(2)}z^{(2)}e^{(2)}, \tag{5.2}$$

so that the total energy, $E = \rho e + \frac{1}{2}\rho u^2$. Each material is governed by the stiffened gas equation of state:

$$p = (\gamma - 1)\rho e - \gamma p_\infty. \tag{5.3}$$

The system is closed following the assumption $p^{(1)} = p^{(2)}$. For the stiffened gas case, this leads to the mixture pressure:

$$p = \frac{\rho e}{\epsilon} - \gamma p_\infty, \tag{5.4a}$$

$$\epsilon = \frac{z^{(1)}}{\gamma^{(1)} - 1} + \frac{z^{(2)}}{\gamma^{(2)} - 1}, \tag{5.4b}$$

$$\gamma p_\infty = \frac{1}{\epsilon}\left(\frac{z^{(1)} p_\infty^{(1)} \gamma^{(1)}}{\gamma^{(1)} - 1} + \frac{z^{(2)} p_\infty^{(2)} \gamma^{(2)}}{\gamma^{(2)} - 1}\right). \tag{5.4c}$$

Of importance in the solution of this system is the mixture sound speed. This is defined as

$$c = \sqrt{\frac{1}{\rho}\left(\gamma p_\infty + p(1 + \frac{1}{\epsilon})\right)}. \tag{5.5}$$

5.1.2 Diffuse interface numerical method for the five-equation model

We have implemented a second-order accurate solver for the diffuse interface model. The conserved variables are evolved according to the standard Godunov-style update:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-\frac{1}{2}} - \mathbf{F}_{i+\frac{1}{2}}\right), \tag{5.6}$$

where $\mathbf{F}_{i-\frac{1}{2}}$ is the flux of the state on the $\frac{x}{t} = 0$ characteristic in the solution to the Riemann problem with initial data $\mathbf{U}_{i-1}^n$ and $\mathbf{U}_i^n$. The initial data for the Riemann problem is obtained using the MUSCL-Hancock method with primitive variable reconstruction, which is described in section 2.3.

The Riemann problem is solved approximately using the HLLC approximate Riemann solver. Under this approach, described in [127] it is sufficient to simply estimate the left and right wave speeds (S_L, S_R) in order to derive an estimate of the flux of conserved variables. The simple estimate

$$S_L = \min(\bar{u} - \bar{c}, u_L - c_L), \quad S_R = \max(\bar{u} + \bar{c}, u_R + c_R) \tag{5.7}$$

is found to work acceptably well, where $u_{L|R}$ and $c_{L|R}$ are the particle velocities and sound speeds in the left/right states given as initial data in the Riemann problem, and $\bar{u}$ and $\bar{c}$ are simple arithmetic means of the left and right state quantities. For the sake of brevity, the full HLLC flux definition is not reproduced here, but many references are available, for example [67].

The treatment of the advection equation for z is awkward in this system. The numerical scheme must be chosen so that the update of z is consistent with the update of ρz in the HLLC solver, otherwise unphysical oscillations with arise on the interface. We use the following second-order update scheme:

$$z_i^{n+1} = z_i^n - \frac{\Delta t}{\Delta x} \left(u^*_{i+1/2} z^*_{i+1/2} - u^*_{i-1/2} z^*_{i-1/2} - z_i^n (u^*_{i+1/2} - u^*_{i-1/2}) \right), \tag{5.8}$$

where $u^*_{i+1/2}$ is computed by the HLLC Riemann solver:

$$u^*_{i+1/2} = \frac{p_R - p_L + \rho_L u_L (S_L - u_L) - \rho_R u_R (S_R - u_R)}{\rho_L (S_L - u_L) - \rho_R (S_R - u_R)}, \tag{5.9}$$

where L and R subscripts denote the left and right initial data submitted to the Riemann solver at edge $i + 1/2$. Finally $z^*_{i+1/2}$ is defined as:

$$z^*_{i+1/2} = \frac{z_i^n}{2} \left(1 + \mathrm{sign}(u^*_{i+1/2}) \right) + \frac{z_{i+1}^n}{2} \left(1 - \mathrm{sign}(u^*_{i+1/2}) \right). \tag{5.10}$$

The extension of the method to multi-dimensions is achieved trivially with a dimensional splitting approach.

5.2 Six-equation model

The Miller and Puckett approach originates from an unpublished manuscript by Collela, Glaz, and Ferguson dating from 1996 [128]. In the same year, the full method was described by Miller and Puckett [35]. The method is based on the Godunov-type schemes which have been used successfully to solve systems of hyperbolic conservation laws in single-fluid problems. Two assumptions on the flow are made: firstly that the velocity field is single-valued (effectively enforcing a no-slip boundary condition between materials), and secondly that pressure is continuous across material interfaces. These assumptions are identical to those underlying the five-equation model, and in fact the use of a pressure relaxation step ensures that the six-equation numerical methods presented here converge to the five-equation solution. The Miller and Puckett approach is built on the assumption of a sharp interface, and requires an interface tracking method capable of computing the volume of each material advected across cell edges.

In this section, we take advantage of numerical methods developed for the five-equation system to simplify the operation of the Miller and Puckett method, and then present an implicit energy update formula with much-improved robustness.

5.2.1 Governing equations

In the Miller and Puckett approach, the system is described by six equations. The total energy equation in the five-equation model is decomposed, leading to equations describing the conservation of individual material density, mixture momentum, individual material energy, as well as an advection equation for volume fraction. In one spatial dimension, the system can be written as:

$$
\begin{aligned}
\frac{\partial(z^{(1)}\rho^{(1)})}{\partial t} + \frac{\partial(z^{(1)}\rho^{(1)}u)}{\partial x} &= 0 && \text{(5.11a)}\\
\frac{\partial(z^{(2)}\rho^{(2)})}{\partial t} + \frac{\partial(z^{(2)}\rho^{(2)}u)}{\partial x} &= 0 && \text{(5.11b)}\\
\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} &= 0 && \text{(5.11c)}\\
\frac{\partial z^{(1)}E^{(1)}}{\partial t} + \frac{\partial(uz^{(1)}E^{(1)})}{\partial x} &= -\frac{z^{(1)}\bar{K}}{K^{(1)}}u\frac{\partial p}{\partial x} - p\frac{z^{(1)}\bar{K}}{K^{(1)}}\frac{\partial u}{\partial x} && \text{(5.11d)}\\
\frac{\partial z^{(2)}E^{(2)}}{\partial t} + \frac{\partial(uz^{(2)}E^{(2)})}{\partial x} &= -\frac{z^{(2)}\bar{K}}{K^{(2)}}u\frac{\partial p}{\partial x} - p\frac{z^{(2)}\bar{K}}{K^{(2)}}\frac{\partial u}{\partial x} && \text{(5.11e)}\\
\frac{\partial z^{(1)}}{\partial t} + \frac{\partial(uz^{(1)})}{\partial x} &= \frac{z^{(1)}\bar{K}}{K^{(1)}}\frac{\partial u}{\partial x} && \text{(5.11f)}
\end{aligned}
$$

where as before we have $z^{(2)} = 1 - z^{(1)}$, $\rho = \rho^{(1)}z^{(1)} + \rho^{(2)}z^{(2)}$, and $E = z^{(1)}E^{(1)} + z^{(2)}E^{(2)}$. Each material is governed by the stiffened gas equation of state:

$$p^{(i)} = (\gamma^{(i)} - 1)\rho^{(i)}e^{(i)} - \gamma^{(i)}p_\infty^{(i)}. \tag{5.12}$$

The quantity $\bar{K}$ is the isentropic bulk modulus of the mixture, defined as

$$\bar{K} = \left(\frac{z^{(1)}}{K^{(1)}} + \frac{z^{(2)}}{K^{(2)}}\right)^{-1}, \tag{5.13}$$

and $K^{(i)}$ is the isentropic bulk modulus of material i. In order to obtain separate equations for energy, the evolution of the total energy of each material is written as a hyberbolic conservation law with two source terms, obtained by expanding the spatial derivative in the evolution equation for total energy of the mixture. The two source terms in equations 5.11d and 5.11e correspond to changes in the kinetic and internal energy respectively. Note that summing the two energy equations and substituting equation 5.13 recovers the evolution equation for total energy:

$$\frac{\partial E}{\partial t} + \frac{\partial(uE)}{\partial x} = -u\frac{\partial p}{\partial x} - p\frac{\partial u}{\partial x} = -\frac{\partial(up)}{\partial x}. \tag{5.14}$$

The volume fraction evolution equation is also more sophisticated than the advection equation used in the classic five-equation model; the relative compressibility of each phase is accounted for when computing the volume fraction change due to the divergence of the velocity field.

Although Miller and Puckett adopt a six-equation model, the assumptions made on the flow are identical to those in the five-equation model. The choice of a six-equation model is motivated instead by numerical concerns. Indeed, as explained in [126], the six-equation model should not be considered as a physical model, but rather as a step-model to solve the five-equation model.

5.2.2 Numerical method

We first briefly summarise the original numerical method from [35], before presenting our approach.

5.2.2.1 A recap of Miller and Puckett's numerical method

The following sequence of steps summarise the update of the conserved variables across one time level:

1. Construct a so-called effective single phase inside every cell, by estimating density, specific internal energy, bulk modulus, and the isentropic pressure derivative of bulk modulus inside every cell, pure and mixed.

2. Solve an approximate Riemann problem on every cell edge using a linear approximation to the Hugoniot of the mixture, obtaining estimates for star-state velocity and pressure.

3. Use star-state velocities to estimate volumes of each material which are advected across each cell edge, using a VOF interface tracking method.

4. In pure fluid regions away from the interface, the conserved variables are updated using the boundary fluxes in the usual Godunov-style update.

5. In regions local to the interface the conserved variables are updated using a series of formulae described in section 5.2.3.

6. Relax mixed cells to pressure equilibrium.

5.2.2.2 An overview of our approach

Our developments are based on the deployment of ideas used to solve the five-equation model. By summation of the two individual energy equations, a five-equation system is recovered, and we use this system rather than the "effective single phase" of Miller and Puckett to compute star states. This allows the exact same HLLC approximate Riemann solver described in section 5.1.2 to be utilised to compute fluxes, star-state pressures, and velocities at all cell edges. The input data for the HLLC solver is obtained using the MUSCL-Hancock method, which ensures that our implementation is second-order accurate in single material regions. The advantage of this approach is that the large number of approximate Riemann solvers developed for the five-equation system can be used without modification. This replaces steps 1 and 2 described above in section 5.2.2.1. Steps 3 and 4 proceed identically– the star-state values are used by the interface tracking method and pure fluid regions are updated using this flux. We then significantly redesign the update formulae for mixed cells, resulting in a much more robust method. The pressure relaxation step for mixed cells is then applied as before.

The method is been described in one spatial dimension, but the extension to the multi-dimensional case is trivial using a splitting approach. Note that this method is not fully conservative – there is a multi-valued flux around the material interface, where the method switches from using the mixed cell update formulae to using a conventional single-material flux updated. Nevertheless, conservation properties of the scheme are superior to many other multi-material approaches (such as the ghost fluid method– in our numerical experiments we have observed the conservation error of this scheme to be over an order of magnitude smaller than the GFM). The superior conservation of this method follows from the property that when a cell completely empties of one material, all of the mass and energy associated with that material (ρz and Ez) is correctly fluxed into neighbouring cells. Future work relating to the conservation properties of the method is described in section 5.4.

5.2.2.3 Interface evolution

The Miller and Puckett approach relies on an interface tracking method to compute the volume of each material advected across each cell edge. This task requires a volume-of-fluid method, and we use Parker and Young's method in our implementation. Interface reconstruction is performed at the beginning of every time step, and the advected volumes are computed using the first-order accurate approach illustrated in figure 5.1. At this point in the algorithm, the interface tracking method does not account for differential compression of materials. This effect is accounted for in the final volume fraction update using equation 5.15a, in which the relative bulk moduli of the two materials is included. We note that the coupling of an interface tracking method which uses additional information (such as MOF or EMOF) to the volume fractions is not trivial, since the volume fraction update used in the Miller and Puckett approach is not a purely geometric operation.

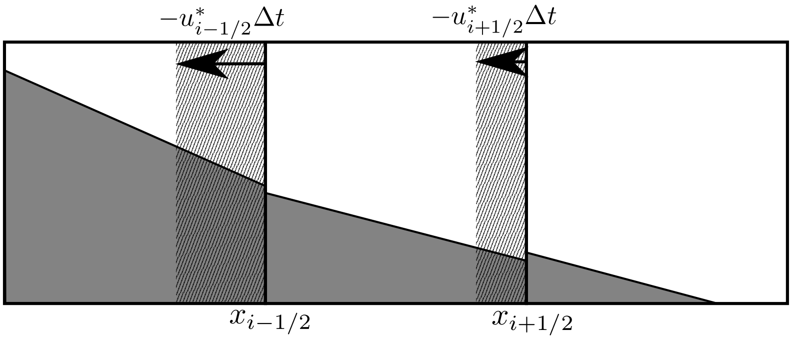


Figure 5.1: The volume fraction update. A piecewise-linear interface reconstruction is performed in each mixed cell, before the star-state velocities are used to find the region which is fluxed across each cell edge during the current time step. The advected volumes, $\Delta V^{(\alpha)}$, of each material is found from the intersection of the reconstructed material regions with the fluxed region.

5.2.3 The mixed cell update

These formulae from Miller and Puckett are used to update the volume fraction, density, velocity, and total energy of each material in mixed cells.

$$z_i^{(1),n+1} = \tilde{z}_i^{(1)} + \left(1 - \tilde{z}_i^{(1)} - \tilde{z}_i^{(2)}\right) \frac{\tilde{z}_i^{(1)} \tilde{K}_A}{K_A^{(1)}}, \tag{5.15a}$$

$$z_i^{(1),n+1} \rho_i^{(1),n+1} = z_i^{(1),n} \rho_i^{(1),n} + \frac{\Delta V_{i-1/2}^{(1)}}{\Delta x} \rho_{i-1/2}^{*(1)} - \frac{\Delta V_{i+1/2}^{(1)}}{\Delta x} \rho_{i+1/2}^{*(1)}, \tag{5.15b}$$

$$z_i^{(2),n+1} \rho_i^{(2),n+1} = z_i^{(2),n} \rho_i^{(2),n} + \frac{\Delta V_{i-1/2}^{(2)}}{\Delta x} \rho_{i-1/2}^{*(2)} - \frac{\Delta V_{i+1/2}^{(2)}}{\Delta x} \rho_{i+1/2}^{*(2)}, \tag{5.15c}$$

$$\rho_i^{n+1} u_i^{n+1} = \rho_i^n u_i^n + \frac{\Delta t}{\Delta x} \left(\tilde{\rho}_{i-1/2}^* u_{i-1/2}^{*2} + p_{i-1/2}^* - \tilde{\rho}_{i+1/2}^* u_{i+1/2}^{*2} - p_{i+1/2}^*\right), \tag{5.15d}$$

$$\begin{aligned} z_i^{(1),n+1} E_i^{(1),n+1} &= z_i^{(1),n} E_i^{(1),n} + \frac{\Delta V_{i-1/2}^{(1)}}{\Delta x} E_{i-1/2}^{*(1)} - \frac{\Delta V_{i+1/2}^{(1)}}{\Delta x} E_{i+1/2}^{*(1)} \\ &+ \frac{\Delta t}{\Delta x} u_i^n \frac{\rho_i^{(1),n+1} z_i^{(1),n+1}}{\rho_i^{n+1}} \left(p_{i-1/2}^* - p_{i+1/2}^*\right) \\ &- p_i^n \left(1 - \tilde{z}_i^{(1)} - \tilde{z}_i^{(2)}\right) \frac{\tilde{z}_i^{(1)} \tilde{K}_A}{K_A^{(1)}}, \end{aligned} \tag{5.15e}$$

$$\begin{aligned} z_i^{(2),n+1} E_i^{(2),n+1} &= z_i^{(2),n} E_i^{(2),n} + \frac{\Delta V_{i-1/2}^{(2)}}{\Delta x} E_{i-1/2}^{*(2)} - \frac{\Delta V_{i+1/2}^{(2)}}{\Delta x} E_{i+1/2}^{*(2)} \\ &+ \frac{\Delta t}{\Delta x} u_i^n \frac{\rho_i^{(2),n+1} z_i^{(2),n+1}}{\rho_i^{n+1}} \left(p_{i-1/2}^* - p_{i+1/2}^*\right) \\ &- p_i^n \left(1 - \tilde{z}_i^{(1)} - \tilde{z}_i^{(2)}\right) \frac{\tilde{z}_i^{(2)} \tilde{K}_A}{K_A^{(2)}}. \end{aligned} \tag{5.15f}$$

A number of quantities which appear in the update formulae require further definition. In the volume fraction update, $\tilde{z}_i^{(1)}$ is the updated volume fraction prior to the adjustment which accounts for velocity field divergence:

$$\tilde{z}_i^{(1)} = z_i^n + \frac{\Delta V_{i-1/2}^{(1)}}{\Delta x} - \frac{\Delta V_{i+1/2}^{(1)}}{\Delta x}. \tag{5.16}$$

Of the star state quantities, $u_{i-1/2}^*$ and $p_{i-1/2}^*$ are obtained from the HLLC approximate Riemann solver, but the star-state densities and total energies are estimated as:

$$\rho_{i-1/2}^{*(\alpha)} = \begin{cases} \rho_{i-1}^{(\alpha),n} & \text{if } u_{i-1/2}^* > 0.0 \\ \rho_i^{(\alpha),n} & \text{if } u_{i-1/2}^* \leq 0.0 \end{cases}, \tag{5.17}$$

$$E_{i-1/2}^{*(\alpha)} = \begin{cases} E_{i-1}^{(\alpha),n} & \text{if } u_{i-1/2}^* > 0.0 \\ E_i^{(\alpha),n} & \text{if } u_{i-1/2}^* \leq 0.0 \end{cases}. \tag{5.18}$$

This is a rather crude estimate, but more sophisticated approaches are constrained by the requirement to correctly set the mass and energy of freshly-emptied cells to zero. In our experience, and also reported in [35], the overall flow solution is insensitive to improved estimates of these star state quantities.

The bulk moduli used to compute the volume fraction and internal energy changes are computed as a volume-weighted average of the bulk moduli of incoming fluid:

$$\frac{\tilde{z}_i^{(\alpha)} \Delta x}{K_A^{(\alpha)}} = \begin{cases} \frac{z_i^{(\alpha),n} - \Delta V_{i+1/2}^{(\alpha)}}{K_i^{(\alpha)}} + \frac{\Delta V_{i-1/2}^{(\alpha)}}{K_{i-1}^{(\alpha)}} & \text{if } \Delta V_{i-1/2}^{(\alpha)} \geq 0 \text{ and } \Delta V_{i+1/2}^{(\alpha)} \geq 0, \\ \frac{z_i^{(\alpha),n}}{K_i^{(\alpha)}} + \frac{\Delta V_{i-1/2}^{(\alpha)}}{K_{i-1}^{(\alpha)}} - \frac{\Delta V_{i+1/2}^{(\alpha)}}{K_{i+1}^{(\alpha)}} & \text{if } \Delta V_{i-1/2}^{(\alpha)} \geq 0 \text{ and } \Delta V_{i+1/2}^{(\alpha)} < 0, \\ \frac{z_i^{(\alpha),n} + \Delta V_{i+1/2}^{(\alpha)} - \Delta V_{i-1/2}^{(\alpha)}}{K_i^{(\alpha)}} & \text{if } \Delta V_{i-1/2}^{(\alpha)} < 0 \text{ and } \Delta V_{i+1/2}^{(\alpha)} \geq 0, \\ \frac{z_i^{(\alpha),n} - \Delta V_{i-1/2}^{(\alpha)}}{K_i^{(\alpha)}} - \frac{\Delta V_{i+1/2}^{(\alpha)}}{K_{i+1}^{(\alpha)}} & \text{if } \Delta V_{i-1/2}^{(\alpha)} < 0 \text{ and } \Delta V_{i+1/2}^{(\alpha)} < 0. \end{cases} \tag{5.19}$$

The bulk modulus of the mixture is then

$$\tilde{K}_A = \left(\frac{\tilde{z}_i^{(1)}}{K_A^{(1)}} + \frac{\tilde{z}_i^{(2)}}{K_A^{(2)}} \right)^{-1}. \tag{5.20}$$

Although upon first aquaintance this discretisation for the mixed cells appears complicated, the implementation of this method is very straightforward since the update just requires translating the analytic formulae above into code. This discretisation also has the important property that when a mixed cell completely empties of material α, the method correctly updates $\rho^{(\alpha)}$ and $E^{(\alpha)}$ to be zero.

5.2.3.1 Pressure relaxation

After the update is complete, the two materials in a mixed cell may not be in pressure equilibrium. A final pressure relaxation step is required so that $p^{(1)} = p^{(2)} = \bar{p}$. This step is crucial in forcing the solution of the six-equation model to converge to that of the five-equation model [126]. We use an identical pressure relaxation algorithm to Miller and Puckett, where the volume fractions and total energies are adjusted iteratively. This is summarised here for completeness. The required volume fraction adjustment, $\Delta z^{(1)}$, is the solution to:

$$\begin{cases} \bar{p} = p^{(1)} - \dfrac{K^{(1)}}{z^{(1)}} \Delta z^{(1)}, \\ \bar{p} = p^{(2)} - \dfrac{K^{(2)}}{z^{(2)}} \Delta z^{(2)}, \\ 0 = \Delta z^{(1)} + \Delta z^{(2)}. \end{cases} \tag{5.21}$$

This system has the solution

$$\Delta z^{(1)} = \frac{z^{(1)}}{K^{(1)}} (p^{(1)} - \bar{p}), \tag{5.22}$$

where

$$\bar{p} = \frac{\frac{z^{(1)} p^{(1)}}{K^{(1)}} + \frac{z^{(2)} p^{(2)}}{K^{(2)}}}{\frac{z^{(1)}}{K^{(1)}} + \frac{z^{(2)}}{K^{(2)}}}. \tag{5.23}$$

In order for the linearisation inherent in equation 5.21 to remain valid, the volume fraction change is limited such that $|\Delta z^{(i)}|/z^{(i)} \leq 0.05$. The change to our system of conserved variables is then:

$$\begin{aligned} z &\leftarrow z + \Delta z^{(1)}, \\ zE^{(1)} &\leftarrow zE^{(1)} - \bar{p} \Delta z^{(1)}, \\ (1-z)E^{(2)} &\leftarrow (1-z)E^{(2)} - \bar{p} \Delta z^{(2)}. \end{aligned} \tag{5.24}$$

5.2.3.2 Constant-velocity advection solution

It is important that the numerical scheme is able to properly simulate the evolution of a material interface between two fluids. We consider the following Riemann problem in which only z is allowed to vary, and analyze the behaviour of Miller and Puckett's update formulae, in the same vein as for the five-equation model in [28].

Lemma 5.1. The discrete update formulae given by Miller and Puckett (equations 5.15a-5.15f) correctly simulate the constant velocity and constant pressure evolution of a material interface, with left and right states given by:

$$\begin{aligned}
u_L &= u_R = u, \\
p_L &= p_R = p, \\
\rho^i_L &= \rho^i_R = \rho^{(i)} \\
e^i_L &= e^i_R = e^{(i)} \\
z_L &\neq z_R.
\end{aligned}$$

Specifically, the method correctly locates the material interface, and maintains the constant pressure.

Proof. The material interface lies to the left of the cell boundary, and without loss of generality we assume that $u > 0$, $z_L > 0$, $z_R = 0$. Since the interface is being translated from the left cell into the right cell, we apply the Miller and Puckett update formulae to the right cell. Updated quantities are denoted by the $n + 1$ superscript. The updated volume fraction is:

$$z^{n+1} = \frac{\Delta V_L^{(1)}}{\Delta x}, \tag{5.25}$$

where $\Delta V_L^{(i)}$ and $\Delta V_R^{(i)}$ refer to advected volumes across the left and right hand edges of the right hand cell. The updated density of material 1 reduces to

$$\rho^{(1),n+1} z^{n+1} = \frac{\Delta V_L^{(1)}}{\Delta x} \rho^{(1)}, \tag{5.26}$$

so that $\rho^{(1),n+1} = \rho^{(1)}$ as required. The updated material 2 density is

$$\rho^{(2),n+1}(1 - z^{n+1}) = \rho^{(2)} + \frac{\Delta V_L^{(2)}}{\Delta x}\rho^{(2)} - \frac{\Delta V_R^{(2)}}{\Delta x}\rho^{(2)} \tag{5.27}$$

and since $1 - z^{n+1} = 1 + \frac{\Delta V_L^{(2)}}{\Delta x} - \frac{\Delta V_R^{(2)}}{\Delta x}$, we obtain $\rho^{(2),n+1} = \rho^{(2)}$. In the momentum update, both star-state pressures must be equal, so:

$$\rho^{n+1} u^{n+1} = \rho^{(2)} u + \frac{\Delta t}{\Delta x} u^2 \left(\frac{\Delta V_L^{(1)} \rho^{(1)} + \Delta V_L^{(2)} \rho^{(2)}}{\Delta V_L^{(1)} + \Delta V_L^{(2)}} - \rho^{(2)} \right). \tag{5.28}$$

Since $\Delta V_L^{(1)} + \Delta V_L^{(2)} = u \Delta t$,

$$\rho^{n+1} u^{n+1} = \rho^{(2)} u + \left(\frac{\Delta V_L^{(1)} \rho^{(1)} + \Delta V_L^{(2)} \rho^{(2)}}{\Delta x} - \frac{\Delta V_R^{(2)}}{\Delta x} \rho^{(2)} \right) u, \tag{5.29}$$

and because $\rho^{n+1} = \frac{\Delta V_L^{(1)}}{\Delta x}\rho^{(1)} + \left(1 + \frac{\Delta V_L^{(2)}}{\Delta x} - \frac{\Delta V_R^{(2)}}{\Delta x}\right)\rho^{(2)}$, we obtain $u^{n+1} = u$. In the energy updates, the source terms are both equal to zero, since $P_L^* - P_R^* = 0$, and $1 - \widetilde{z^{(1)}} - \widetilde{z^{(2)}} = 0$. As a result, the only changes to the energy of each material are due to advection:

$$\begin{aligned} E^{(1),n+1} z^{n+1} &= \frac{\Delta V_L^{(1)}}{\Delta x} E^{(1)} \\ &= E^{(1)} z^{n+1}, \end{aligned} \tag{5.30}$$

and

$$\begin{aligned} E^{(2),n+1}(1 - z^{n+1}) &= E^{(2)} + \frac{\Delta V_L^{(2)}}{\Delta x} E^{(2)} - \frac{\Delta V_R^{(2)}}{\Delta x} E^{(2)} \\ &= E^{(2)}(1 - z^{n+1}). \end{aligned} \tag{5.31}$$

Since we know that the velocity and densities are unchanged, the specific internal energies, and hence pressures, of each material also remain constant. This concludes the proof that the method correctly simulates the constant-velocity advection of a material interface. □

5.2.4 A robust implicit energy update

In our numerical experiments, Miller and Puckett's original update formula has been found to be lacking in robustness. In this section a novel energy update formula is described which rectifies this problem.

The Miller and Puckett energy update (equations 5.15e and 5.15f) is a linearisation of the energy evolution equation for material α:

$$\frac{\partial(z^{(\alpha)} E^{(\alpha)})}{\partial t} + \frac{\partial(u z^{(\alpha)} E^{(\alpha)})}{\partial x} = -\frac{\rho^{(\alpha)} z^{(\alpha)}}{\rho} u \frac{\partial p}{\partial x} - \frac{z^{(\alpha)} \bar{K}}{K^{(\alpha)}} p \frac{\partial u}{\partial x}. \tag{5.32}$$

We have found the this linearisation is prone to computing negative internal energies, causing the simulation to halt. This problem tends to occur when large velocity and pressure gradients are present in the solution. Our approach to mollifying this problem is based on discretising the equation for the evolution of internal energy in mixed cells, and deriving a positivity-preserving update for internal energy in mixed cells. Although this internal energy is non-conservative (as is Miller and Puckett's total energy evolution equations), states in the vicinity of shocks are correctly computed since the star-state quantities used in these discretisations are computed from the conservative five-equation system. This will be demonstrated in our numerical tests of the method. Note that total energy is still evolved analagous to equation 5.15 away from the interface. The equations for internal energy which we consider at mixed cells are:

$$\frac{\partial(z^{(1)}e^{(1)}\rho^{(1)})}{\partial t}+\frac{\partial(z^{(1)}e^{(1)}\rho^{(1)}u)}{\partial x}=-\frac{z^{(1)}\bar{K}}{K^{(1)}}p\frac{\partial u}{\partial x}, \tag{5.33a}$$

$$\frac{\partial(z^{(2)}e^{(2)}\rho^{(2)})}{\partial t}+\frac{\partial(z^{(2)}e^{(2)}\rho^{(2)}u)}{\partial x}=-\frac{z^{(2)}\bar{K}}{K^{(2)}}p\frac{\partial u}{\partial x}. \tag{5.33b}$$

In the material α case this is discretised as:

$$\begin{aligned} z_i^{(\alpha),n+1}e_i^{(\alpha),n+1}\rho_i^{(\alpha),n+1} &= z_i^{(\alpha),n}e_i^{(\alpha),n}\rho_i^{(\alpha),n} \\ &+\frac{\Delta V_{i-1/2}^{(\alpha)}}{\Delta x}e_{i-1/2}^{*(\alpha)}\rho_{i-1/2}^{*(\alpha)}-\frac{\Delta V_{i+1/2}^{(\alpha)}}{\Delta x}e_{i+1/2}^{*(\alpha)}\rho_{i-1/2}^{*(\alpha)} \\ &-p\left(1-\tilde{z}_i^{(1)}-\tilde{z}_i^{(2)}\right)\frac{\tilde{z}_i^{(\alpha)}\tilde{K}_A}{K_A^{(\alpha)}}. \end{aligned} \tag{5.34}$$

where the factor p is yet to be defined. The updated total energy of material α follows as

$$z_i^{(\alpha),n+1}E_i^{(\alpha),n+1}=z_i^{(\alpha),n+1}\rho_i^{(\alpha),n+1}\left(e_i^{(\alpha),n+1}+\frac{1}{2}\left((u_i^{n+1})^2\right)\right). \tag{5.35}$$

We now examine the optimal choice of p. From the definition of $\tilde{z}$,

$$\left(1-\tilde{z}_i^{(1)}-\tilde{z}_i^{(2)}\right)=\frac{\Delta t}{\Delta x}\left(u_{i+1/2}^{*}-u_{i-1/2}^{*}\right), \tag{5.36}$$

which describes the compression or expansion of fluid over the time step. We use the following branch in order to choose an appropriate value for p in the linearisation of this term:

$$p=\begin{cases} p_i^{(\alpha),n} & \text{if } u_{i+1/2}^{*}-u_{i-1/2}^{*}<0, \\ p_i^{(\alpha),n+1} & \text{otherwise.} \end{cases} \tag{5.37}$$

As a result, if material in the cell is undergoing a compression, the existing cell pressure is used to obtain an explicit update for internal energy. Alternatively if the cell is undergoing expansion, the new pressure is substituted, obtaining the implicit update formula:

$$\begin{aligned} z_i^{(\alpha),n+1}e_i^{(\alpha),n+1}\rho_i^{(\alpha),n+1} &= z_i^{(\alpha),n}e_i^{(\alpha),n}\rho_i^{(\alpha),n} \\ &+\frac{\Delta V_{i-1/2}^{(\alpha)}}{\Delta x}e_{i-1/2}^{*(\alpha)}\rho_{i-1/2}^{*(\alpha)}-\frac{\Delta V_{i+1/2}^{(\alpha)}}{\Delta x}e_{i+1/2}^{*(\alpha)}\rho_{i-1/2}^{*(\alpha)} \\ &-\left((\gamma-1)e_i^{(\alpha),n+1}\rho_i^{(\alpha),n+1}-\gamma p_\infty\right)\left(1-\tilde{z}_i^{(1)}-\tilde{z}_i^{(2)}\right)\frac{\tilde{z}_i^{(\alpha)}\tilde{K}_A}{K_A^{(\alpha)}}. \end{aligned} \tag{5.38}$$

Since $z_i^{(\alpha),n+1}$ is known, this can be solved for $e_i^{(\alpha),n+1}\rho_i^{(\alpha),n+1}$:

$$z_i^{(\alpha),n+1} e_i^{(\alpha),n+1} \rho_i^{(\alpha),n+1} = \frac{z_i^{(\alpha),n} e_i^{(\alpha),n} \rho_i^{(\alpha),n} + \Delta_{\text{advec}} + \gamma p_\infty \Delta_{\text{vol}}}{1 + \dfrac{(\gamma - 1)\Delta_{\text{vol}}}{z_i^{(\alpha),n+1}}}, \tag{5.39}$$

where

$$\Delta_{\text{advec}} = \frac{\Delta V_{i-1/2}^{(\alpha)}}{\Delta x} e_{i-1/2}^{*(\alpha)} \rho_{i-1/2}^{*(\alpha)} - \frac{\Delta V_{i+1/2}^{(\alpha)}}{\Delta x} e_{i+1/2}^{*(\alpha)} \rho_{i-1/2}^{*(\alpha)}, \tag{5.40}$$

$$\Delta_{\text{vol}} = \left(1 - \tilde{z}_i^{(1)} - \tilde{z}_i^{(2)}\right) \frac{\tilde{z}_i^{(\alpha)} \tilde{K}_A}{K_A^{(\alpha)}}. \tag{5.41}$$

We note that this update preserves the ability of the original method to correctly advect material interfaces in constant-pressure conditions, as proven in section 5.2.3.2.

5.2.4.1 Positivity-preservation of the novel energy update

We now examine some properties of this internal energy update formula. To begin with, we examine the density update formula from Miller and Puckett.

Lemma 5.2. The density update described by equations 5.15b and 5.15c is positivity-preserving.

Proof. The density update for material α is

$$\rho_i^{(\alpha),n+1} = \frac{1}{z_i^{(\alpha),n+1}} \left(z_i^{(\alpha),n} \rho_i^{(\alpha),n} + \frac{\Delta V_{i-1/2}^{(\alpha)}}{\Delta x} \rho_{i-1/2}^{*(\alpha)} - \frac{\Delta V_{i+1/2}^{(\alpha)}}{\Delta x} \rho_{i+1/2}^{*(\alpha)} \right). \tag{5.42}$$

We assume that $\rho_{i-1}^{(\alpha),n}$, $\rho_i^{(\alpha),n}$ and $\rho_{i+1}^{(\alpha),n}$ are positive. First consider the case in which material is leaving the cell in both directions. From the definition of $\rho_{i-1/2}^{*(\alpha)}$ in equation 5.17, in this case $\rho_{i-1/2}^{*(\alpha)} = \rho_{i+1/2}^{*(\alpha)} = \rho_i^{(\alpha),n}$. Then the update formula simplifies to:

$$\rho_i^{(\alpha),n+1} = \rho_i^{(\alpha),n} \frac{\tilde{z}_i^{(\alpha)}}{z_i^{(\alpha),n+1}}. \tag{5.43}$$

Since all terms are ≥ 0, we have $\rho_i^{(\alpha),n+1} \geq 0$.

Next consider the case in which material leaves the cell from one side, and enters from the other. Without loss of generality, assume that $\Delta V_{i-1/2}^{(\alpha)}$, $\Delta V_{i+1/2}^{(\alpha)} \geq 0$. Since $\rho_{i-1/2}^{*(\alpha)} = \rho_{i-1}^{(\alpha),n}$ and $\rho_{i+1/2}^{*(\alpha)} = \rho_i^{(\alpha),n}$, we obtain

$$\rho_i^{(\alpha),n+1} = \frac{1}{z_i^{(\alpha),n+1}} \left(\rho_i^{(\alpha),n} \left(z_i^{(\alpha),n} - \frac{\Delta V_{i+1/2}^{(\alpha)}}{\Delta x} \right) + \frac{\Delta V_{i-1/2}^{(\alpha)}}{\Delta x} \rho_{i-1}^{(\alpha),n} \right). \tag{5.44}$$

This expression must be positive because $\frac{\Delta V^{(\alpha)}_{i+1/2}}{\Delta x} \leq z_i^{(\alpha),n}$.

The final case to consider is when material enters the cell from both sides, so $\Delta V^{(\alpha)}_{i-1/2} \geq 0$ and $\Delta V^{(\alpha)}_{i+1/2} \leq 0$. In this case all three terms on the right hand side of equation 5.42 are positive, so the updated density is also positive. □

Building on this, we examine the novel internal energy update. Initially the scope of the analysis is restricted to the ideal gas case (materials for which $p_\infty = 0$ in equation 5.12).

Lemma 5.3. The internal energy update described in section 5.2.4 is positivity-preserving for ideal gases with $\gamma > 0.0$.

Proof. The internal energy update can be written as:

$$e_i^{(\alpha),n+1}\rho_i^{(\alpha),n+1} = \begin{cases} \frac{z_i^{(\alpha),n}e_i^{(\alpha),n}\rho_i^{(\alpha),n}+\Delta_{\text{advec}}+\gamma p_\infty \Delta_{\text{vol}}}{z_i^{(\alpha),n+1}+(\gamma-1)\Delta_{\text{vol}}} & \text{if}\Delta_{\text{vol}} \geq 0, \\ \frac{z_i^{(\alpha),n}e_i^{(\alpha),n}\rho_i^{(\alpha),n}+\Delta_{\text{advec}}-p_i^{(\alpha),n}\Delta_{\text{vol}}}{z_i^{(\alpha),n+1}} & \text{otherwise.} \end{cases} \tag{5.45}$$

Note that the expression $z_i^{(\alpha),n}e_i^{(\alpha),n}\rho_i^{(\alpha),n} + \Delta_{\text{advec}}$ is identical to the density update, substituting ρe for ρ. Therefore, an identical proof to that used in lemma 5.2 shows that this expression is positive. Examining the $\Delta_{\text{vol}} \geq 0$ case first, we use the fact that $z_i^{(\alpha),n+1} = \tilde{z}_i^{(\alpha)} + \Delta_{\text{vol}}$ to simplify the denominator to $\tilde{z}_i^{(\alpha)} + \gamma\Delta_{\text{vol}}$. From the definition of $\tilde{z}_i^{(\alpha)}$ (equation 5.16), $\tilde{z}_i^{(\alpha)} + \gamma\Delta_{\text{vol}} \geq 0$. Since $p_\infty = 0$, both denominator and numerator have been shown to be non-negative, so the internal energy update is positivity preserving.

In the $\Delta_{\text{vol}} < 0$ case, the analysis is even simpler. It is immediately obvious that all terms in the equation are positive, so the resulting internal energy update must also be positive. □

The proof above does not generalise to stiffened gases for the following reason. Although an identical proof can be used under the assumption that $p_i^{(\alpha),n} \geq 0$, the update only preserves positive internal energy, and not positive pressure (equivalent to the condition $\rho e \geq \frac{\gamma p_\infty}{\gamma-1}$). Then, negative pressures may lead to negative internal energies in subsequent steps.

It is possible to regain the positivity-preservation property for stiffened gases by imposing a lower limit on the value of $p_i^{(\alpha),n} \geq 0$ used in the internal energy update in the $\Delta_{\text{vol}} < 0$ case:

$$p_i^{(\alpha),n} = \max\left(p_i^{(\alpha),n}, \frac{z_i^{(\alpha),n}e_i^{(\alpha),n}\rho_i^{(\alpha),n} + \Delta_{\text{advec}} - \epsilon}{\Delta_{\text{vol}}}\right), \tag{5.46}$$

where ϵ is a small positive constant. However in practice the update has proven sufficiently robust in simulations involving stiffened gases that we do not resort to this limiting procedure.

5.2.5 Algorithm summary

The following steps summarise the update of the conserved variables over one time level. In two dimensions this strategy may be used to advance the system by a single substep in a dimensionally-split approach.

1. Compute the wave speeds of both materials in each cell, and compute the largest stable time step.

2. Sum the two energy equations to obtain a five-equation system, perform MUSCL reconstruction of the primitive variables, and then apply any Riemann solver developed for this system to compute star state pressures, star state velocities, and fluxes of conserved variables across every cell edge.

3. Perform interface reconstruction with the VOF interface tracking method, and then use star-state velocities to compute the volume of each material advected across each cell edge this time step as illustrated in figure 5.1.

4. In single-material regions update the conserved variables as normal using the flux computed earlier. Local to the interface, apply the special update formulae for volume fraction (equation 5.15a), partial densities (equations 5.15b and 5.15c), momentum (equation 5.15d), and internal energies (equation 5.45). Total energies are then regained through addition of the kinetic energy component using the updated density and velocity.

5. Relax mixed cells to pressure equilibrium as described in section 5.2.3.1.

5.3 Numerical Results

We now present numerical results obtained using our novel numerical method for the six-equation system with the robust energy update, denoted as the MP method in the remainder of this section. To demonstrate the improvement in material interface resolution, we compare against results obtained using the diffuse interface method for the five-equation system (denoted as the DI method). All simulations are run using a CFL number of 0.8, and fluxes in both methods are computed using the second order MUSCL-Hancock method with HLLC approximate Riemann solver. Both methods were implemented from

	Update formula	
	Original	Robust internal energy
Gas-gas shock tube	✓	✓
Gas-water shock tube	×	✓
Three-state two-phase problem	×	✓
Shocked R22 bubble	✓	✓
Shocked helium bubble	✓	✓
Richtmyer-Meshkov instability	✓	✓
Underwater shocked bubble	×	✓
Tin implosion	×	✓

Table 5.1: A comparison of the robustness of Miller and Puckett's energy update formula (equations 5.15e and 5.15f) with the novel implicit approach (equation 5.45). The symbols ✓and × denote the success or failure of each approach to run the test problem.

scratch in a parallel framework using the openMP library [59]. A comparison of the robustness between the original Miller and Puckett method update formula and our robust internal energy update formula is presented in table 5.1, showing that our approach is able to run a much wider range of the problems which are presented in this section. All subsequent results in this section are obtained with the robust internal energy update formula. All initial conditions are presented in terms of the vector of primitive variables, which for our six-equation system is: $\mathbf{W} = (\rho^{(1)}z^{(1)}, \rho^{(2)}z^{(2)}, u, v, p^{(1)}, p^{(2)}, z^{(1)})^T$. Unless otherwise mentioned, all units are dimensionless.

5.3.1 Gas-gas shock tube

Since it was proven mathematically that the MP method correctly simulates constant-velocity advection of a material interface, we omit any advection tests and begin with a one-dimensional two-material shock tube problem. The computational domain is the region $[0, 1]$, and transmissive boundary conditions are applied. Both materials are ideal gases, with material 1 (in the $z = 1$ region) having $\gamma_1 = 1.4$ and material 2 having $\gamma_2 = 1.2$. The initial conditions are:

$$\begin{aligned} x < 0.5 : \mathbf{W} &= \left(1, 0, 0, 10^5, 0, 1\right)^T, \\ x \geq 0.5 : \mathbf{W} &= \left(0, 0.125, 0, 0, 10^4, 0\right)^T. \end{aligned}$$

The solution is output at time $t = 0.0007$. Results obtained with a 100 cell grid are shown in figure 5.2. For further comparison, see results obtained using the ghost fluid method in [38].

As expected, we see that the MP method is able to limit the mixing region to a single cell. Away from the material interface, both approaches resolve the shock and rarefaction waves equally well. Neither approach results in spurious pressure oscillations near the interface. Note the correct shock speed predicted by the MP method. The conservation errors in total mass, momentum, and energy of the mixture are shown in figure 5.3. This demonstrates that after the shock passes over the interface, conservation errors are negligible.

5.3.2 Gas-water shock tube

The second test case is a more demanding shock tube problem. This problem describes an air-water mixed Riemann problem. This time there are large differences in material properties across the interface, making it a more difficult test of robustness. Both materials are described by the stiffened gas equation of state, with equation of state parameters of $\gamma_1 = 4.4$, $p_{\infty 1} = 6 \times 10^8$, $\gamma_2 = 1.4$, $p_{\infty 2} = 0$, and the computational domain is $[-2, 2]$. Transmissive boundary conditions are applied. The initial states are:

$$\begin{aligned} x < 0.7 : \mathbf{W} &= \left(1000, 0, 0, 10^9, 0, 1\right)^T, \\ x \geq 0.7 : \mathbf{W} &= \left(0, 50, 0, 0, 10^5, 0\right)^T. \end{aligned}$$

The solution is output at $t = 9 \times 10^{-4}$, and results obtained with 100 grid cells are shown in figure 5.4.

Despite the fact that the MP method is not fully conservative, it actually locates the shock position better than the conservative diffuse interface method, which places the shock too far to the right. We also see very little density undershoot near the material interface in water.

5.3.3 Three-state two-material problem

This is our first two-dimensional problem, which simulates the evolution of a vortex at the triple point. The computational domain is $[0, 7] \times [0, 3]$ with transmissive boundary conditions. Both materials are ideal gases, with material 1 (in the $z = 1$ region) having $\gamma_1 = 1.5$ and material 2 having $\gamma_2 = 1.4$. The initial conditions are:

$$\begin{aligned} x < 1 : \mathbf{W} &= (1, 0, 0, 0, 1, 0, 1)^T, \\ x \geq 1,\ y \geq 1.5 : \mathbf{W} &= (0.125, 0, 0, 0, 0.1, 0, 1)^T, \\ x \geq 1,\ y < 1.5 : \mathbf{W} &= (0, 1, 0, 0, 0, 0.1, 0)^T. \end{aligned}$$

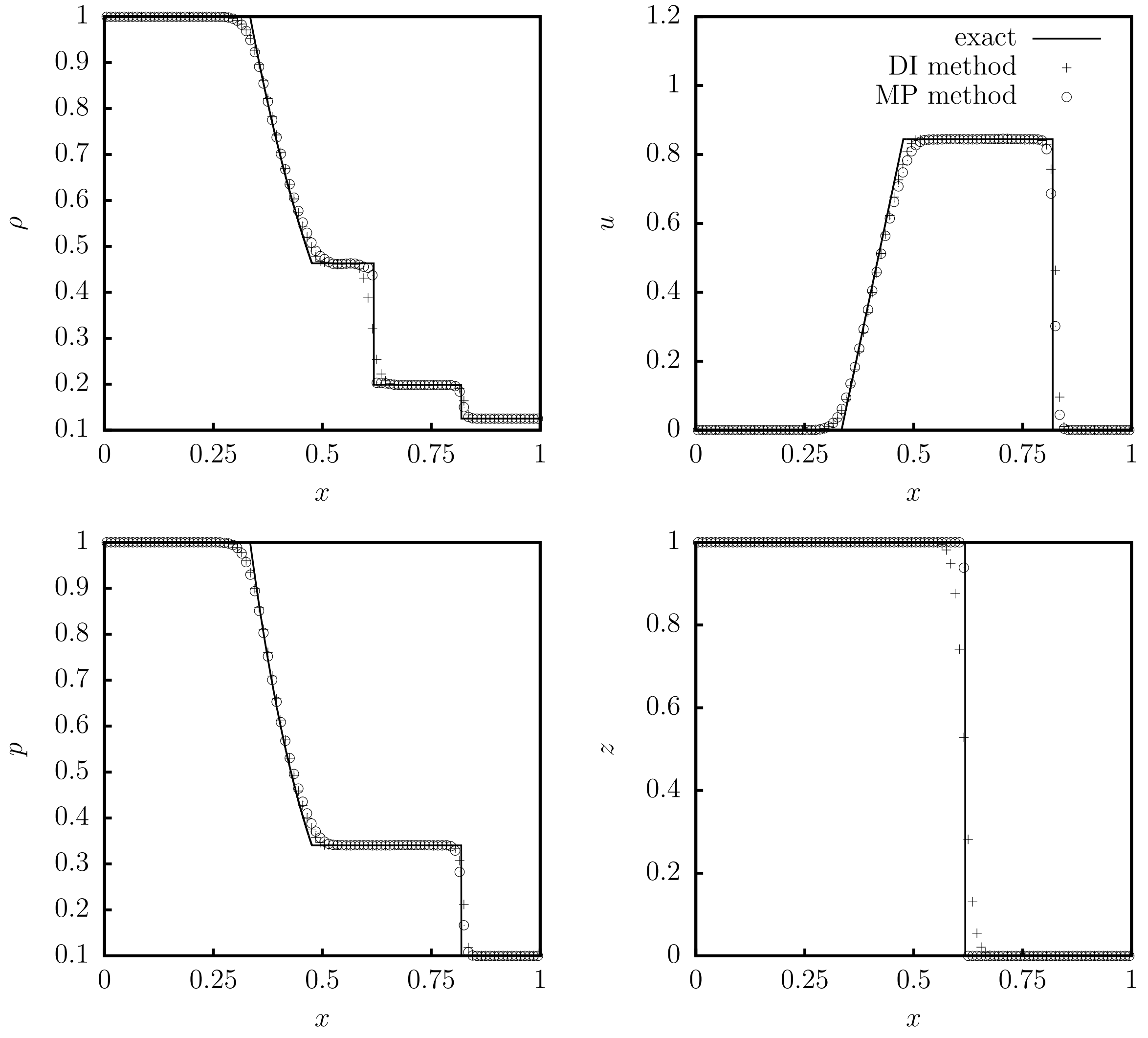


Figure 5.2: Numerical results for the gas-gas shock tube test case at $t = 0.14$. The solid line is the exact solution and the points compare the computed solutions from the MP method and the diffuse interface five-equation model, on a domain discretised with 100 cells.

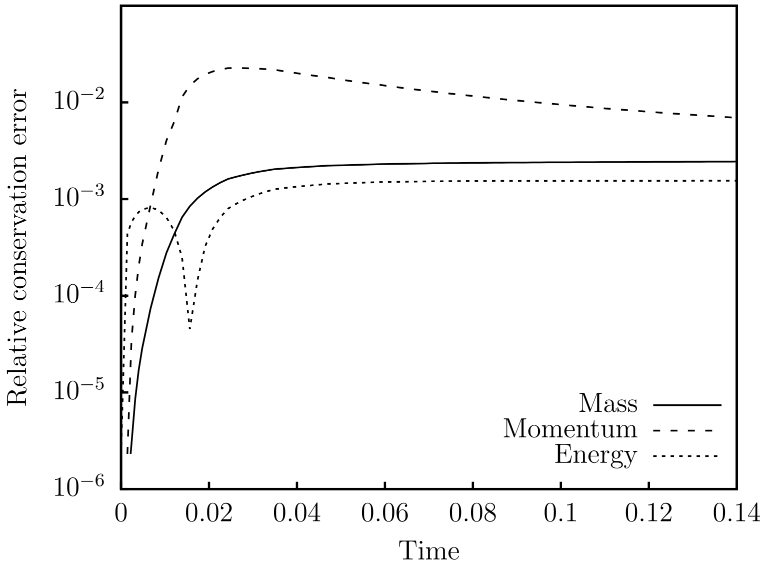


Figure 5.3: Evolution of the relative conservation errors in the MP method on the gas-gas shock tube problem.

The solution on a 210 × 90 grid is shown at two time levels in figures 5.5 and 5.6.

These results clearly show the improved resolution of the material interface using the MP method, without any change in the solution away from the material interface, and without inducing any spurious pressure oscillations. At the tip of the vortex, with the MP method, a few bubbles can be seen to break off – this is a result of the numerical surface tension induced by Parker and Youngs volume-of-fluid interface reconstruction. More accurate interface tracking methods would reduce this issue. The MP method compares favourably to results obtained with the anti-diffusion approach, and appears approximately equivalent to results obtained with the THINC approach presented in [34].

5.3.4 Shocked helium bubble

This is a well-studied problem in which a circular helium gas bubble in air is hit by a planar shock wave. The computational domain is the region $[0, 325] \times [0, 89]$, with transmissive boundary conditions applied to the left and right hand sides, and reflective boundary conditions applied to the top and bottom. Both materials are modelled by the ideal gas equation of state, with air having $\gamma_1 = 1.4$ and helium having $\gamma_2 = 1.667$. The helium is initially contained in a circular region centred at $(175, 44.5)$ with radius 25, with primitive variables:

$$\begin{aligned} x \geq 225 &: \mathbf{W} = (1.3764, 0, -0.394, 0, 1.5698, 0, 1)^T , \\ x < 225,\ (x-175)^2 + (y-44.5)^2 \geq 25^2 &: \mathbf{W} = (1, 0, 0, 0, 1, 0, 1)^T , \\ x < 225,\ (x-175)^2 + (y-44.5)^2 < 25^2 &: \mathbf{W} = (0, 0.138, 0, 0, 0, 1, 0)^T . \end{aligned}$$

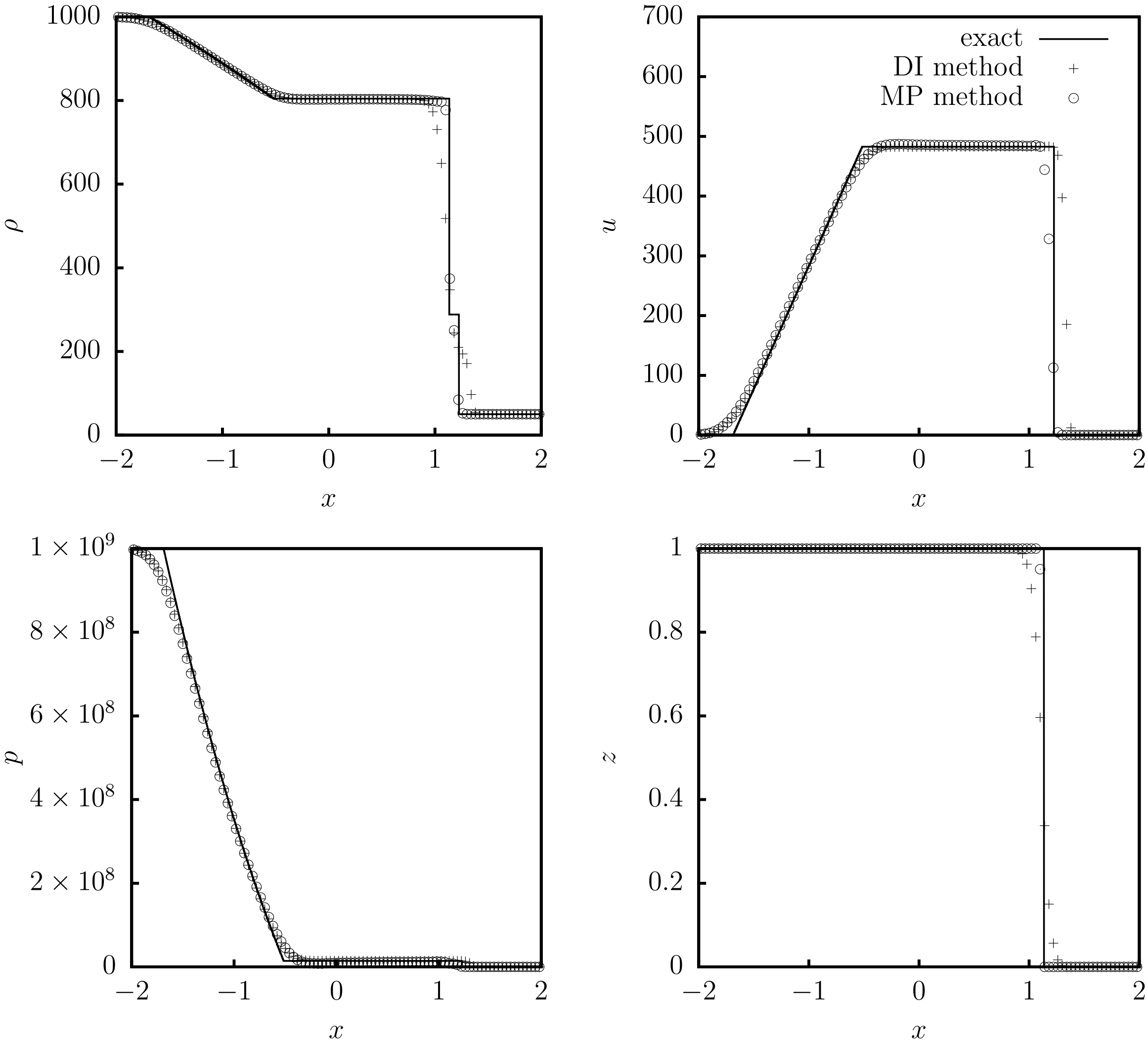


Figure 5.4: Numerical results for the gas-water shock tube test case at $t = 9 \times 10^{-4}$. The solid line is the exact solution and the points compare the computed solutions from the MP method and the diffuse interface five-equation model, on a domain discretised with 100 cells.

]

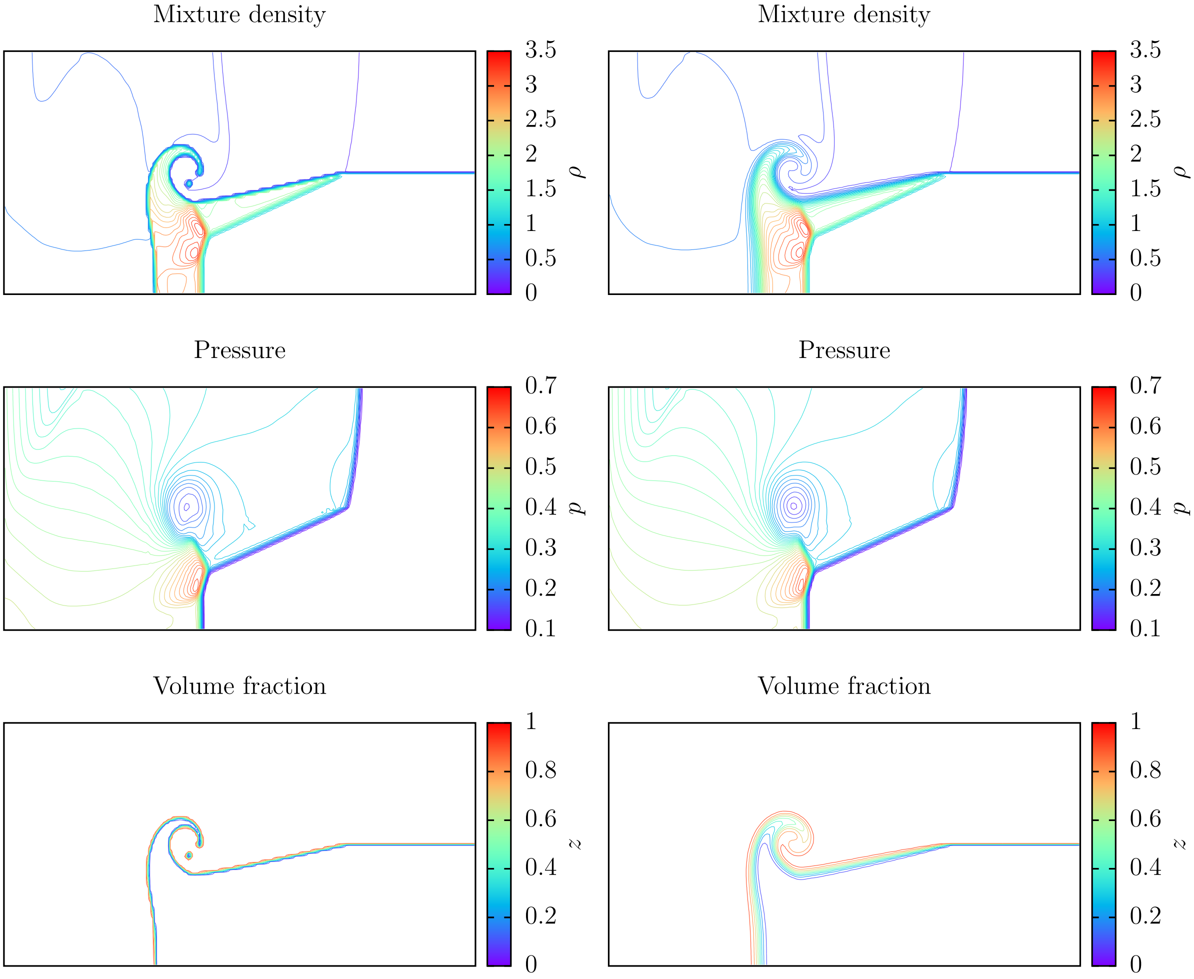


Figure 5.5: Numerical results for the three-state two-material problem at $t = 2.4$ on a 210×90 grid. The solutions in the left hand column are obtained with the MP method, and right hand column with the diffuse interface method. Contours are placed at the same levels for both methods.

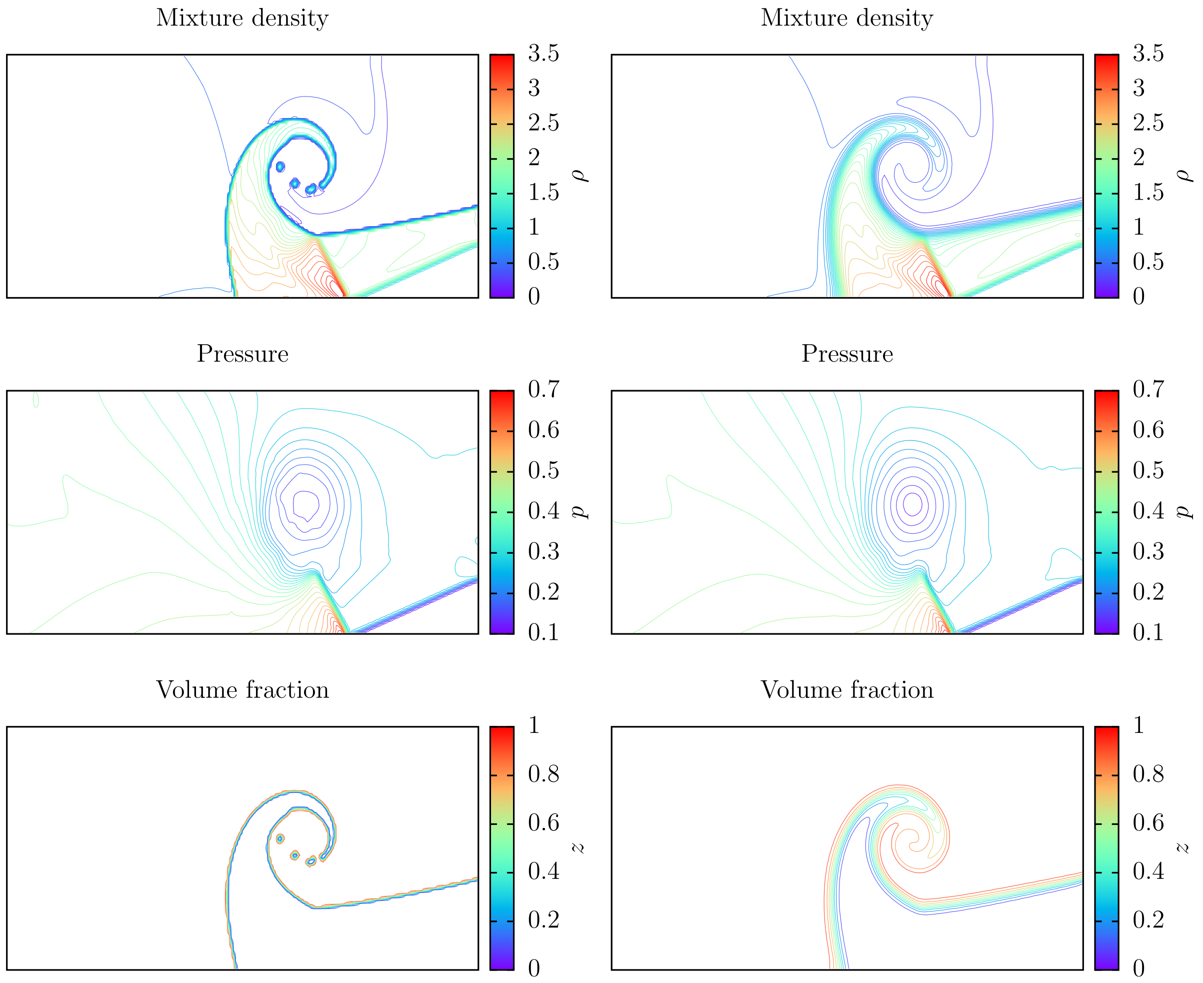


Figure 5.6: Numerical results for the three-state two-material problem at $t = 4.8$ on a 210×90 grid. The solutions in the left hand column are obtained with the MP method, and right hand column with the diffuse interface method. Contours are placed at the same levels for both methods.

	DI method	MP method	Relative difference
Mass	37962.6	37970.8	-2.2×10^{-4}
x-momentum	-14905.6	-14914.9	-6.2×10^{-4}
y-momentum	-2.1×10^{-14}	-3.2×10^{-6}	-
Energy	113911	113928	-1.5×10^{-4}

Table 5.2: The total mass, momenta, and energy in the domain at time $t = 280$ in the shocked helium bubble test, computed on a 325×89 grid. The relative difference is a useful measure of the conservation error of the MP method. For comparison with the mass conservation measurements of the GFM on this test case (section 4.6.2.2), the final relative error in the helium bubble mass is 2.94×10^{-4}. This is over an order of magnitude smaller than the best-performing GFM.

A sequence of numerical Schlieren images of density are presented in figure 5.7, computed on a 2600×712 grid for straightforward comparison to previous simulations. Our results can be compared qualitatively to experiment in [129], and to other numerical methods in [130] and [111]. These results are further evidence that the MP method gives identical solutions to the diffuse interface method for the five-equation model away from the interface, while maintaining a perfectly sharp interface representation.

We examine the conservation error in the MP method on this test case by running a much coarser 325×89 grid and summing mass, momenta, and total energy across the entire domain and comparing to the diffuse interface method, which is fully conservative. The results of this computation are shown in table 5.2. This shows that the conservation errors generated by the MP method are acceptably small.

5.3.5 Shocked R22 bubble

This is another common problem in which a circular R22 gas bubble in air is hit by a planar shock wave travelling at around Mach 1.2. The computational domain is the region $[0, 0.445] \times [0, 0.089]\text{m}^2$, with transmissive boundary conditions applied to the left and right hand sides, and reflective boundary conditions applied to the top and bottom. Both materials are modelled by the ideal gas equation of state, with air having $\gamma_1 = 1.4$ and R22 having $\gamma_2 = 1.249$. In this test case all quantities are given in SI units to facilitate easy comparison with experiment. The R22 is initially contained in a circular

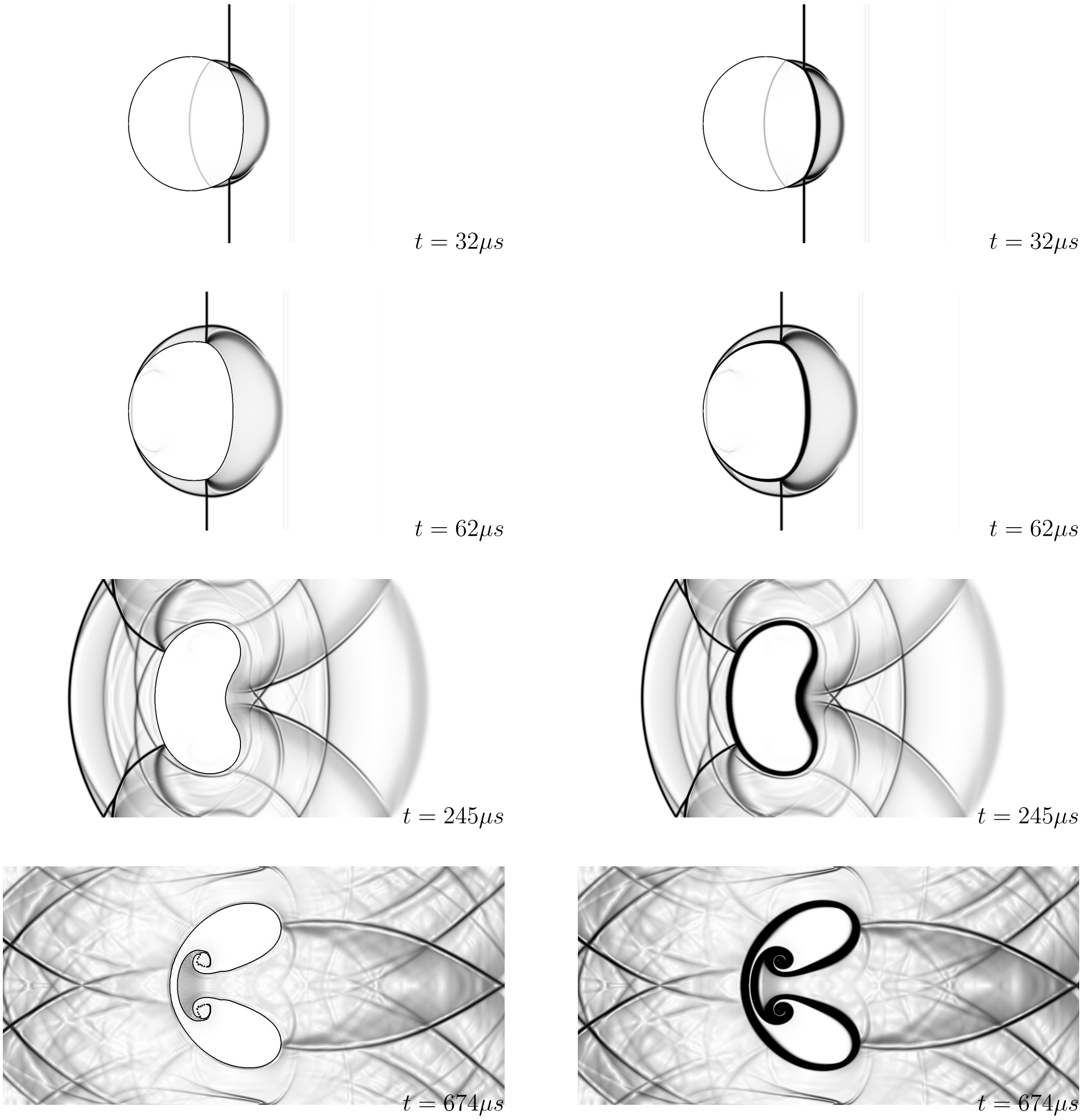


Figure 5.7: Numerical results for the shocked helium bubble test on a 2600 × 712 grid. All times are given relative to the initial incidence of the shock on the bubble. Each image is a numerical Schlieren plot of density. Results from the MP method are shown in the left column of images, and results using the DI method are shown in the right column of images. See figure 5.11 for experimental results from [129] at corresponding times.

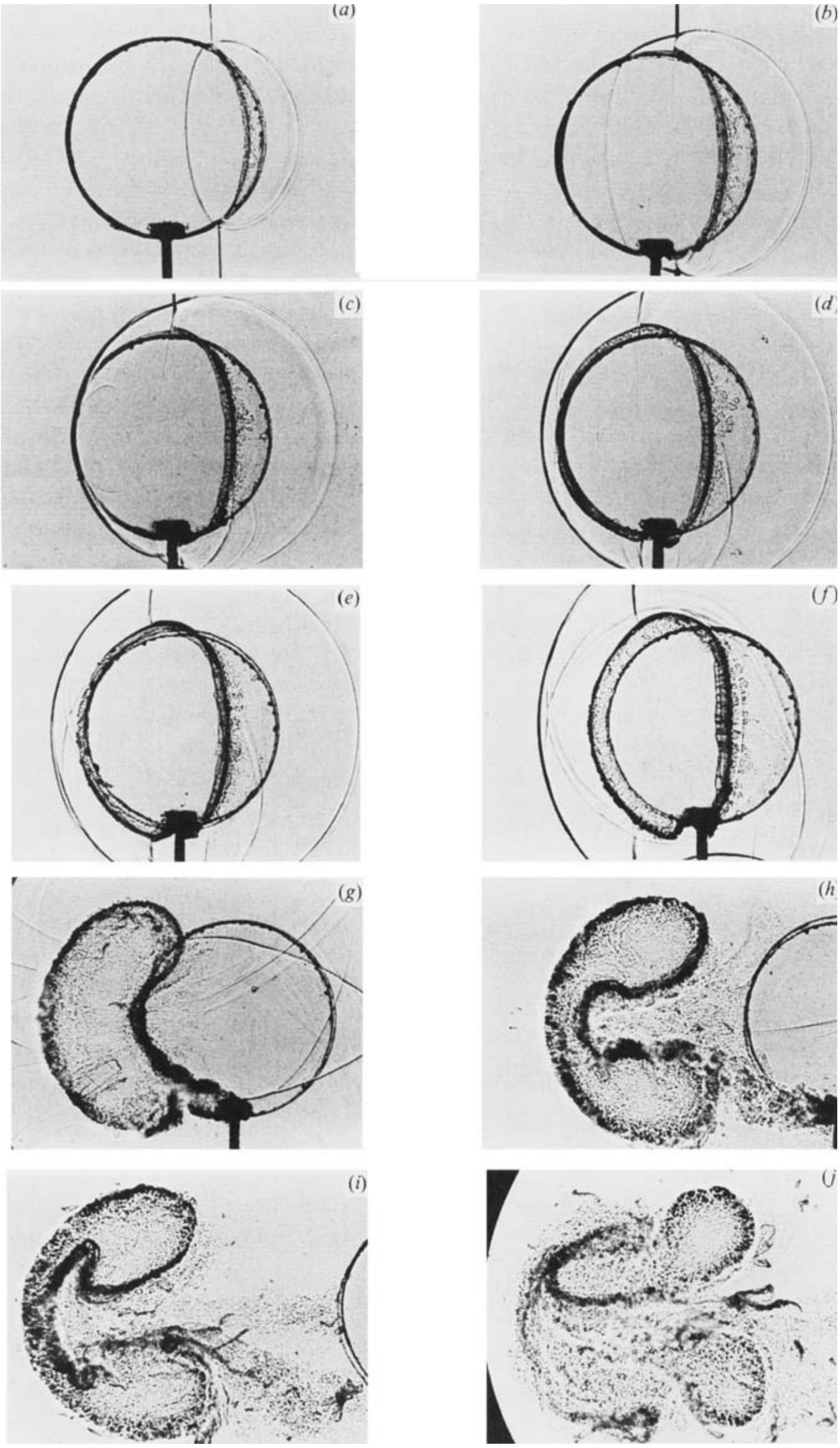


Figure 5.8: Shadow photographs of the interaction between a planar shock wave and a cylindrical helium volume from experiment in [129]. (a) t=32 μs, (b) 52 μs, (c) 62 μs, (d) 72 μs, (e) 82 μs, (f) 102 μs, (g) 245 μs, (h) 427 μs, (i) 674 μs, (j) 983 μs.

region centred at $(0.225, 0.0445)$ with radius 0.025m, with primitive variables:

$$\begin{aligned}
x \geq 0.275 &: \mathbf{W} = \left(1.686, 0, -113.5, 0, 1.59 \times 10^5, 0, 1\right)^T, \\
\left.\begin{array}{r} x < 0.275, \\ (x-0.225)^2 + (y-0.0445)^2 \geq 0.025^2 \end{array}\right\} &: \mathbf{W} = \left(1.225, 0, 0, 0, 1.01325 \times 10^5, 0, 1\right)^T, \\
\left.\begin{array}{r} x < 0.275, \\ (x-0.225)^2 + (y-0.0445)^2 < 0.025^2 \end{array}\right\} &: \mathbf{W} = \left(0, 3.863, 0, 0, 0, 1.01325 \times 10^5, 0\right)^T.
\end{aligned}$$

A sequence of numerical Schlieren images of density are presented in figures 5.9 and 5.10, computed on a 3560 × 712 grid. These images show qualitative agreement with bubble shapes observed in experiment [129], as well as computed with the THINC method [34]. These images further confirm that the MP method results in identical solutions to the DI method away from material interfaces.

In order to quantitatively confirm that our model gives correct shock-interface interactions, we measure space-time locations of several prominent features of this flow. A linear least squares fit is applied to each set of space-time points over the specified time interval in order to compute velocities. An illustration of the features we track is provided in figure 5.12. Space-time diagrams for both numerical methods are presented in figure 5.13. We measure velocities over the following time intervals (bearing in mind that the initial shock-bubble collision occurs at $t = 60\mu$s). The incident shock is measured over $t \in [60, 310]\mu$s. The upstream bubble wall velocity is measured over $t \in [60, 460]\mu$s. The downstream bubble wall velocity is measured over $t \in [260, 460]\mu$s. The refracted shock is measured over $t \in [60, 260]\mu$s. The transmitted shock is measured over $t \in [260, 310]\mu$s. Finally, the bubble jet velocity is measured over $t \in [260, 310]\mu$s. Our velocity measurements for both methods are compared to experiment as well as four other published numerical methods in table 5.3. Our results lie within the errors associated with the experimental measurements, and are very close to the measured values for two other interface sharpening approaches (THINC and anti-diffusion) measured in [34]. The velocity of the downstream bubble wall is significantly off, but this may be explained by the ambiguity in where to measure this velocity. For the purposes of reproducibility, we measured the point on the downstream bubble wall situated at $y = 0.0445$m, but the velocity of this point is skewed upwards by the bubble jet formation. Our measurement of the jet velocity at this point agrees closely with the experimental value.

Overall, these results confirm quantitatively that the MP method predicts very similar shock-interface dynamics to the conservative five-equation method. Our measurements are consistent with other numerical methods in the literature, and generally show close agreement to experiment.

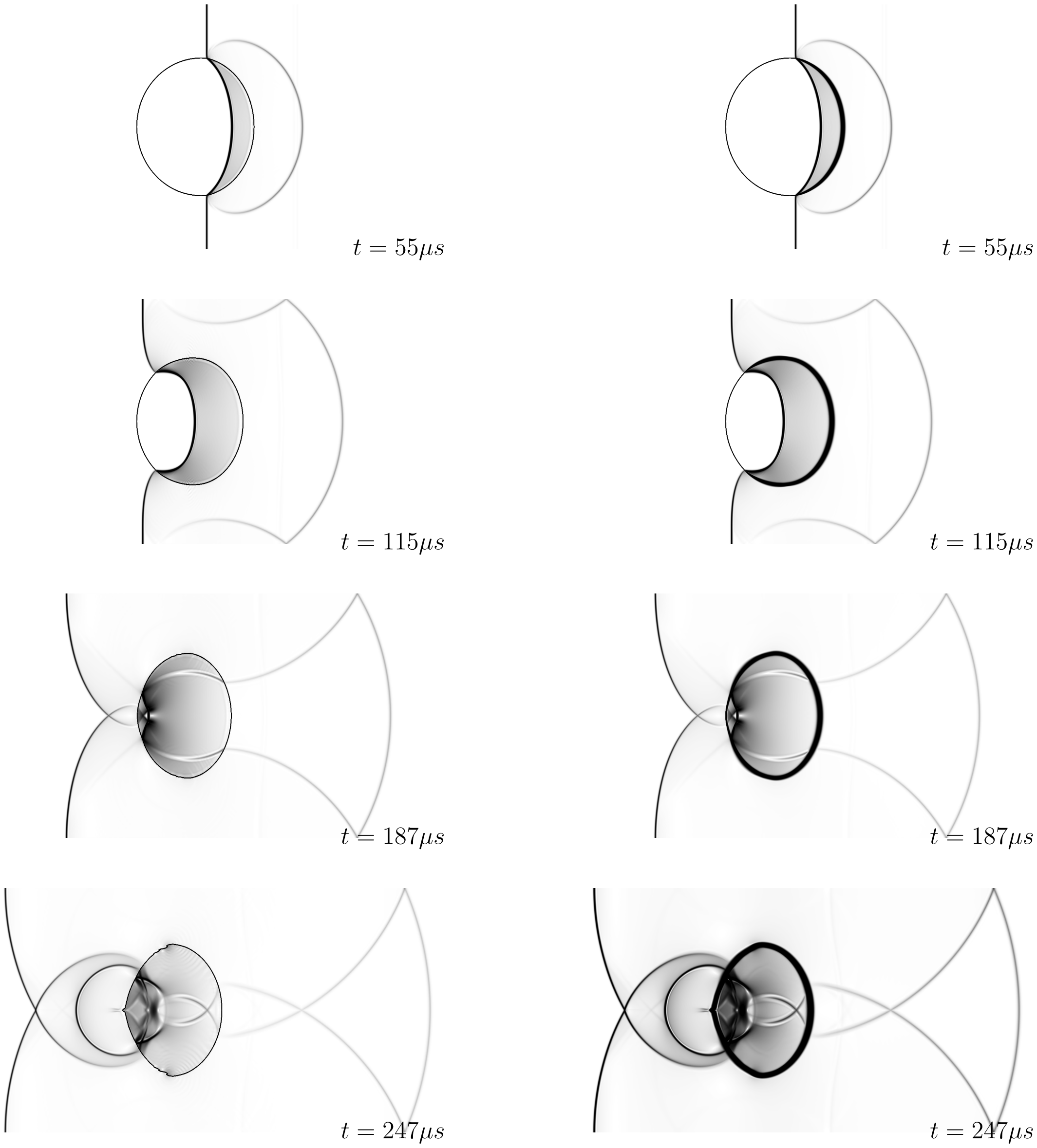


Figure 5.9: Numerical results for the shocked R22 bubble test on a 3560×712 grid. Each image is a numerical Schlieren plot of density over the region $[0.12, 0.32] \times [0, 0.089]$. Results from the MP method are shown in the left column of images, and results using the DI method are shown in the right column of images. All times are given relative to the initial impact of the shock on the R22.

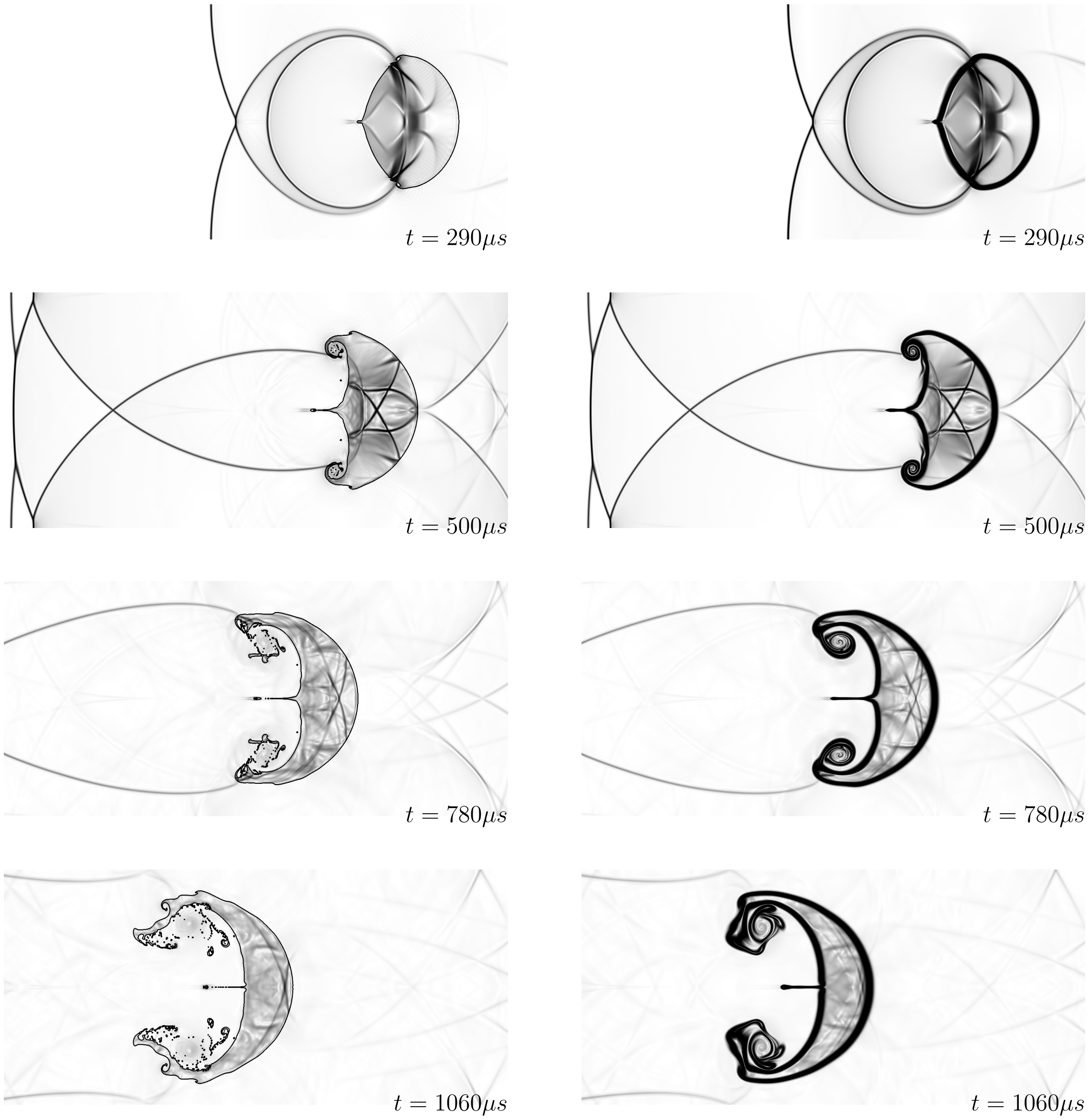


Figure 5.10: Numerical results for the shocked R22 bubble test on a 3560 × 712 grid. Each image is a numerical Schlieren plot of density over the region $[0.05, 0.25] \times [0, 0.089]$. Results from the MP method are shown in the left column of images, and results using the DI method are shown in the right column of images. All times are given relative to the initial impact of the shock on the R22.

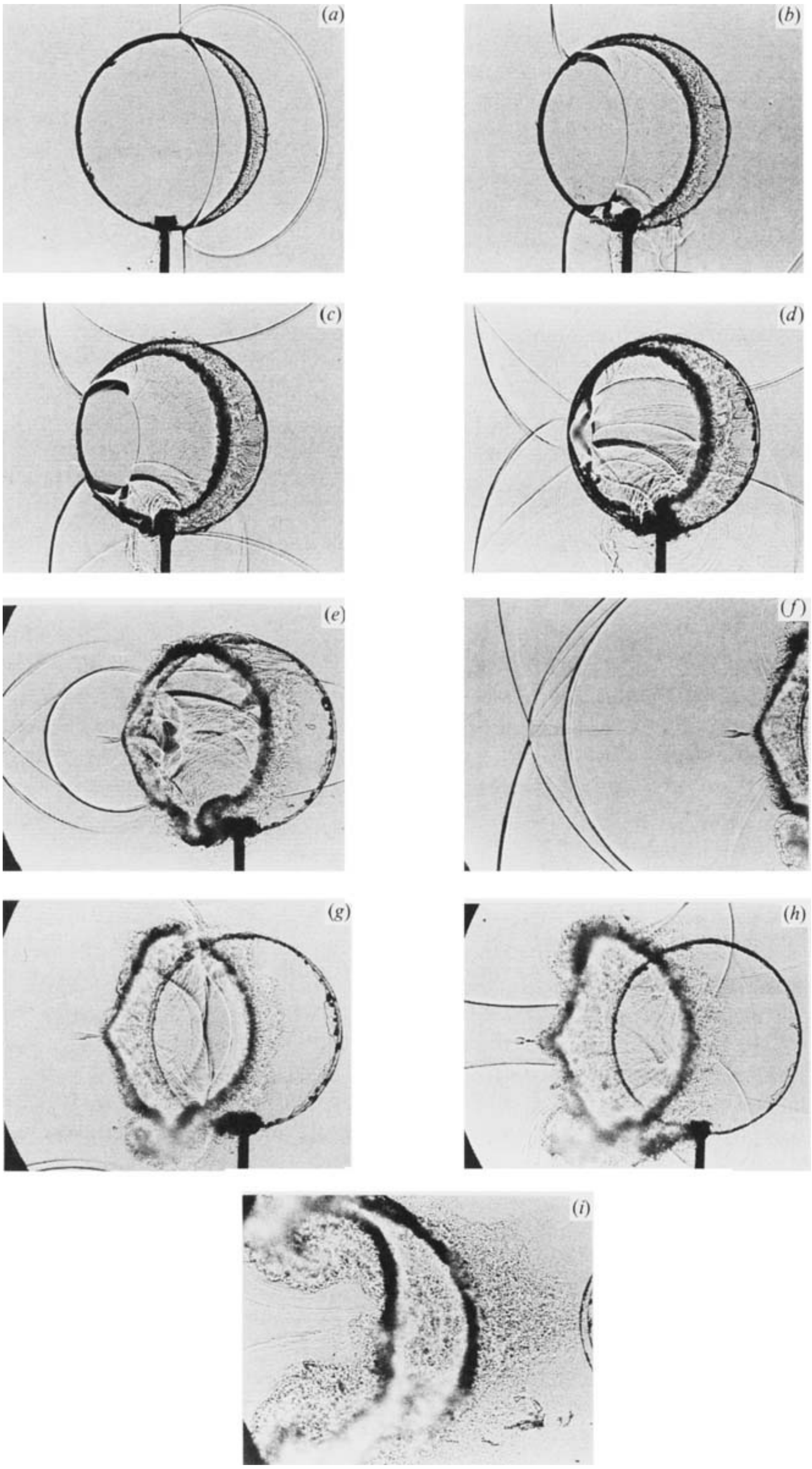


Figure 5.11: Shadow photographs of the interaction between a planar shock wave and a cylindrical R22 volume from experiment in [129]. (a) t=55 μs, (b) 115 μs, (c) 135 μs, (d) 187 μs, (e) 247 μs, (f) 318 μs, (g) 342 μs, (h) 417 μs, (i) 1020 μs.

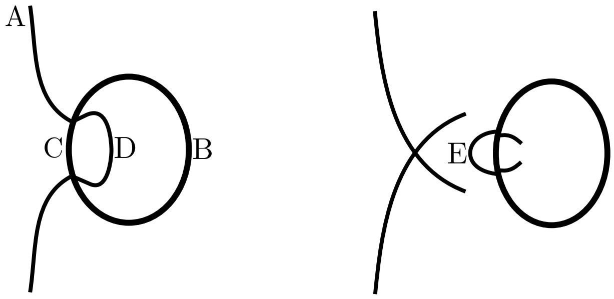


Figure 5.12: An illustration of the features we tracked in the shocked R22 test. A = incident shock, B = upstream bubble wall, C = downstream bubble wall, D = refracted shock, E = transmitted shock. We also track F = bubble jet, which is located at the same point as C, but measured over a smaller time interval.

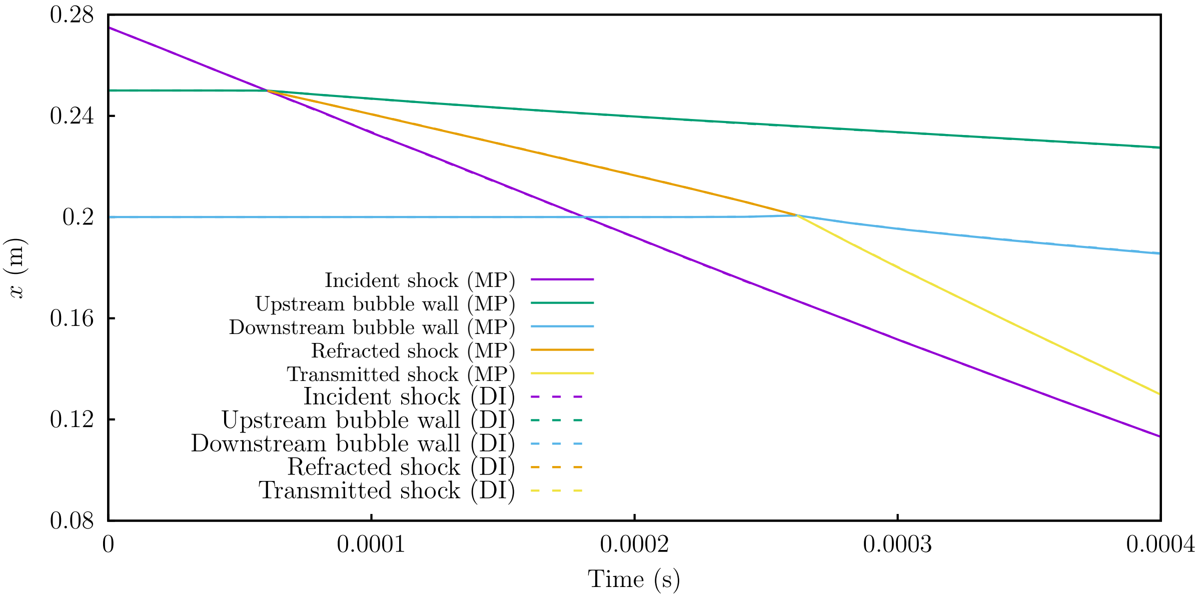


Figure 5.13: Space-time locations of various features in the shocked R22 bubble test. Refer to figure 5.12 for definitions of the tracked features. Velocities measured from this data are shown in table 5.3.

Velocity (ms^{-1})	A	B	C	D	E	F
Experiment [129]	415	73	78	240	540	153
Kokh & Lagoutierre (Lagrange-remap) [122]	411	65	86	243	525	-
Shyue & Xiao (THINC) [34]	410	65	87	244	538	-
Shyue & Xiao (anti-diffusion) [34]	410	64	100	244	532	-
Quirk & Karni (diffuse interface) [6]	420	74	116	254	560	-
Our results (DI method)	**412**	**65.2**	**100**	**244**	**532**	**147**
Our results (MP method)	**412**	**65.0**	**101**	**243**	**536**	**149**

Table 5.3: A comparison of the velocities of various features in the shocked R22 bubble test to those from the literature. See figure 5.12 for an explanation of the feature labels. There is some ambiguity over the measurement location of feature C, possibly explaning the variation in that measurement.

5.3.6 Richtmyer-Meshkov instability

In this test problem we simulate the Richtmyer-Meshkov instability through collision of a planar shock with a sinusoidal material interface. The computational domain is the region $[0,4] \times [0,0.5]$, with transmissive boundary conditions applied to the left and right hand sides, and reflective boundary conditions applied to the top and bottom. Both materials are modelled by the ideal gas equation of state, with air on the right having $\gamma_1 = 1.4$ and SF6 on the left having $\gamma_2 = 1.093$. The material interface is a vertical sine wave, given by the equation:

$$x = 2.9 - \epsilon \sin(2\pi(y + 0.25)).$$

The parameter ϵ determines the amplitude of the initial interface perturbation. The initial conditions are:

$$\begin{aligned}
x \geq 3.2 &: \mathbf{W} = (1.411, 0, -0.39, 0, 1.628, 0, 1)^T, \\
\left.\begin{array}{r} x < 3.2, \\ x \geq 2.9 - \epsilon \sin(2\pi(y + 0.25)) \end{array}\right\} &: \mathbf{W} = (1, 0, 0, 0, 1, 0, 1)^T, \\
\left.\begin{array}{r} x < 3.2, \\ x < 2.9 - \epsilon \sin(2\pi(y + 0.25)) \end{array}\right\} &: \mathbf{W} = (0, 5.04, 0, 0, 0, 1, 0)^T.
\end{aligned}$$

We use a 1600×400 grid for all calculations. A comparison of solutions obtained from the MP method and DI method for an initial perturbation amplitude of $\epsilon = 0.2$ is presented in figure 5.14. This shows the advantage of the MP method in resolving vortical structures.

We now quantitatively compare our numerical results against the impulsive model originally developed by Richtmyer [131], in which it is assumed that an initially sinusoidal interface is accelerated impulsively, with constant pressure subsequently. In this model, the growth rate of the instability amplitude, a, is given by:

$$\frac{\mathrm{d}a}{\mathrm{d}t} = k \cdot \Delta u \cdot a_0 \cdot \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2}, \tag{5.47}$$

where k is the wavenumber of the initial perturbation, Δu is the difference between shocked and unshocked mean interface velocities, a_0 is the initial perturbation amplitude after collision with the shock, and ρ_1 (ρ_2) is the post-shocked density of the fluid on the left (right) of the interface. The model is only valid in the $ka_0 \ll 1$ regime.

The simulation is now run with a smaller initial perturbation of $\epsilon = 0.02$. These initial conditions yield values of $k = 2\pi$, $\Delta u = 0.27$, $a_0 = 0.016$, $\rho_1 = 8.78$ and $\rho_2 = 1.54$, giving a predicted growth rate of 0.019, and Atwood number of 0.70. Our results for the perturbation amplitude and growth rate over the time interval $t \in [0, 4]$ are show in figure 5.15. These results show little difference between the two numerical methods,

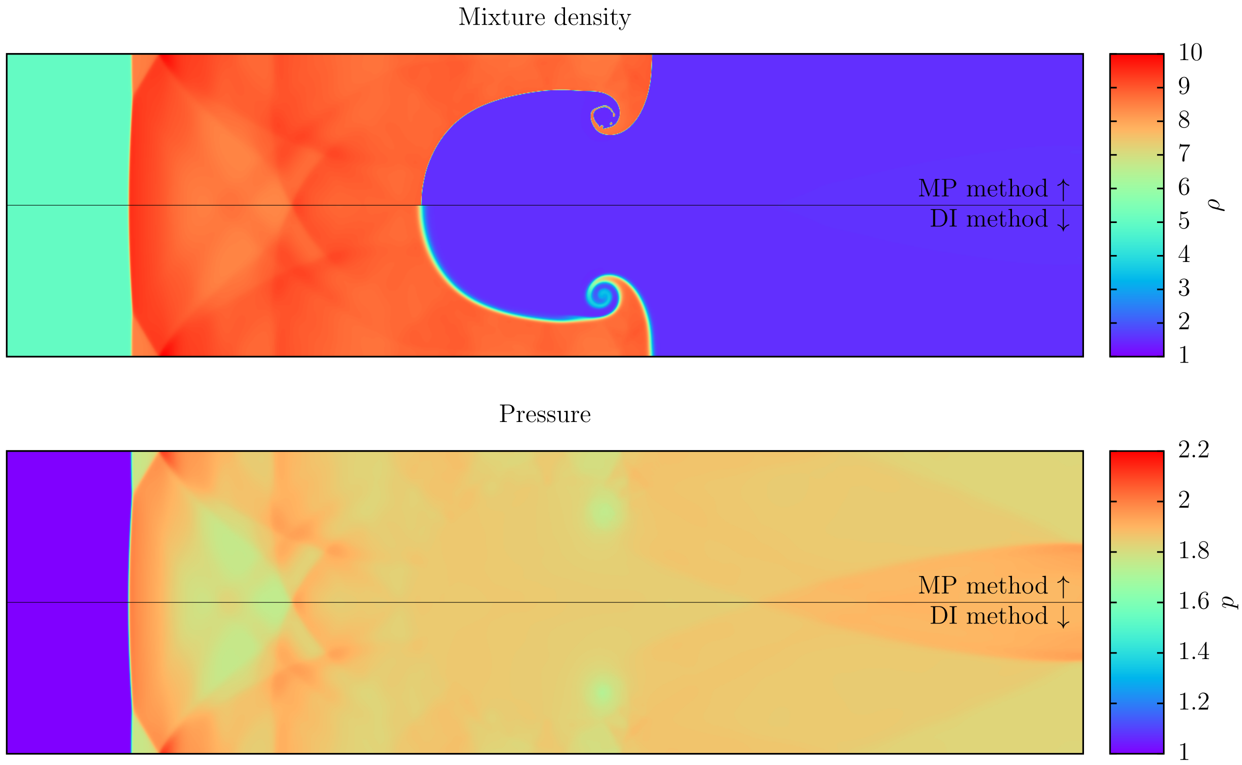


Figure 5.14: Numerical results for the Richtmyer-Meshkov instability test case with an initial perturbation amplitude of 0.2 on a 1600 × 200 grid at $t = 4$.

and excellent agreement with the predicted theoretical value. This is further evidence that the MP method yields correct shock-interface interaction dynamics. Note that with $ka_0 \approx 0.1 \ll 1$ this perturbation amplitude is inside the region of validity of the impulsive theory. Similar results obtained with the ghost fluid method are presented in [121].

5.3.7 Underwater shocked bubble

We now present a more challenging air-water test case. A bubble of air submerged in water is hit by a planar shockwave, causing complete bubble collapse. The computational domain for this test is $[0, 12] \times [0, 12]$, with transmissive boundary conditions applied to all four edges. The stiffened gas equation of state is used for both materials, with water having $\gamma_1 = 7.15$ and $p_{\infty 1} = 3.309 \times 10^8$, and air having $\gamma_2 = 1.4$ and $p_{\infty 2} = 0$. The initial conditions are

$$x < 2.4 : \mathbf{W} = (1.31, 0, 67.32, 0, 19000, 0, 1)^T,$$

$$\left.\begin{array}{r} x \geq 2.4, \\ (x-6)^2 + (y-6)^2 \geq 3^2 \end{array}\right\} : \mathbf{W} = (1, 0, 0, 0, 1, 0, 1)^T,$$

$$\left.\begin{array}{r} x \geq 2.4, \\ (x-6)^2 + (y-6)^2 < 3^2 \end{array}\right\} : \mathbf{W} = (0, 0.0012, 0, 0, 0, 1, 0)^T.$$

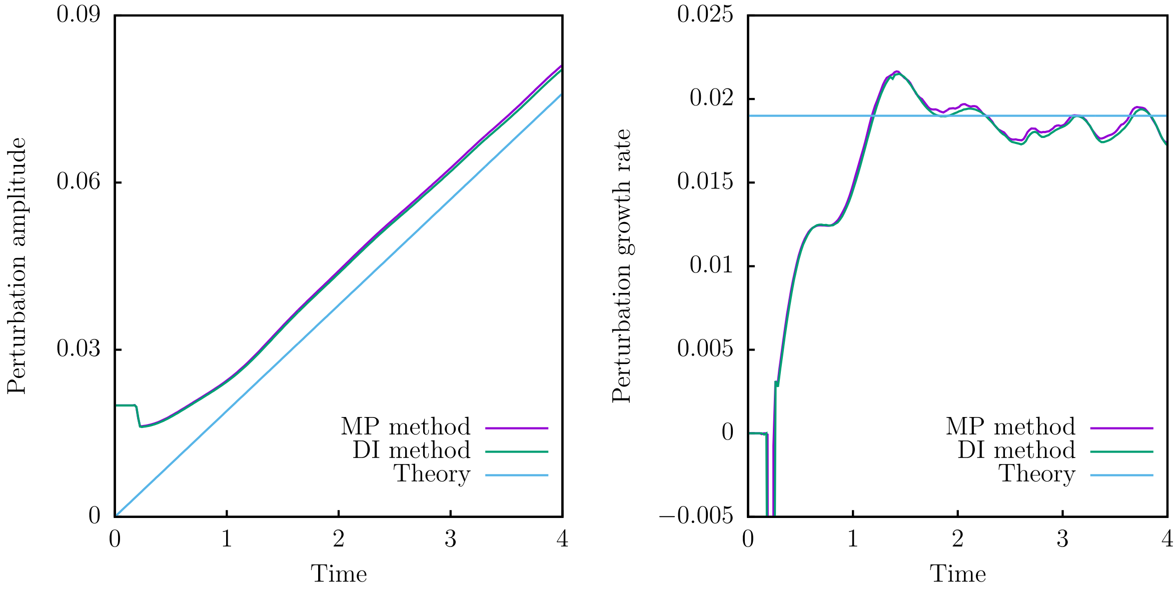


Figure 5.15: Perturbation amplitude and growth rate on the Richtmyer-Meshkov instability test case, computed using a 1600 × 400 grid and an initial perturbation amplitude of 0.02. Our numerical results are compared to the predicted theoretical value obtained using Richtmyer's impulsive theory [131].

A sequence of numerical Schlieren images of density are presented in figure 5.16, computed on a 400 × 400 grid. This case demonstrates the robustness of the MP method, since the novel internal energy update prevents negative internal energies which otherwise develop if the original Miller and Puckett energy update is applied.

5.3.8 Tin implosion

The final numerical experiment which we present is an extremely challenging test case, featuring large pressure gradients (pressure ratio of 10^6 across the initial shock) and changes in material properties across the interface (density ratio of approximately 10^4 across the material interface). These results are included to simply show the robustness of the MP method – they are not expected to be physically meaningful since the equations of state are not valid in the extreme conditions present. This test simulates a semi-circle of tin imploding onto air, with initially sinusoidal interface perturbations. The computational domain is $[0, 25] \times [-25, 25]$, with a reflective boundary condition on the left edge, and transmissive boundary conditions applied elsewhere. The stiffened gas equation of state is used for both materials, with tin having $\gamma_1 = 3.27$ and $p_{\infty 1} = 149,500$, and air having $\gamma_2 = 1.4$ and $p_{\infty 2} = 0$. Using polar coordinates, the equation of the material interface is

$$r = 20 + 0.4\cos(22\theta).$$

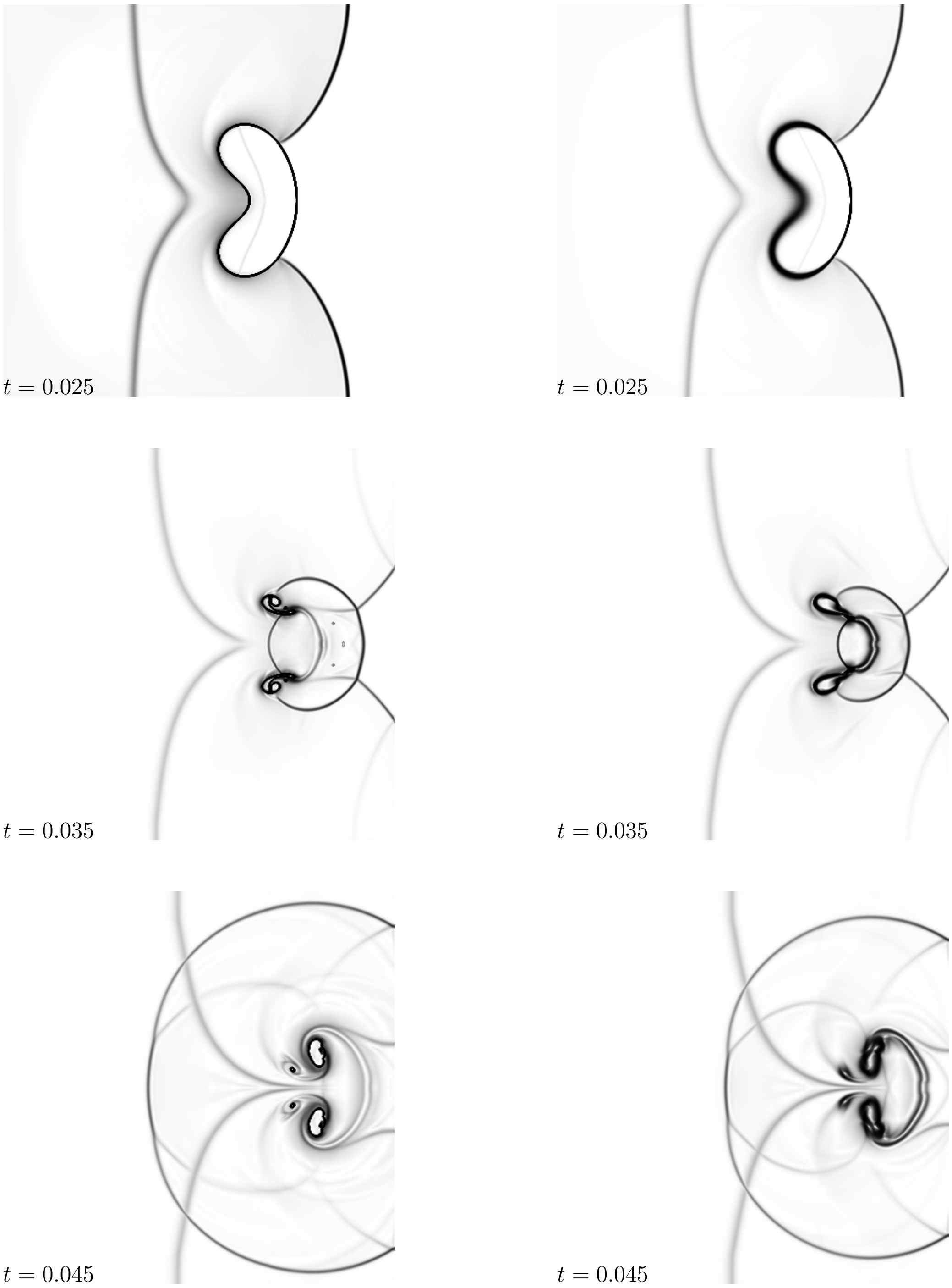


Figure 5.16: Numerical results for the underwater shocked bubble test on a 400 × 400 grid. Each image is a numerical Schlieren plot of density over the region $[0, 12] \times [0, 12]$. Results from the MP method are shown in the left column of images, and results using the DI method are shown in the right column of images.

The initial conditions are:

$$\begin{aligned} r \geq 24 : \mathbf{W} &= \left(11.84, 0, 0, 0, 10^6, 0, 1\right)^T, \\ \left.\begin{array}{r} r < 24, \\ r \geq 20 + 0.4\cos(22\theta) \end{array}\right\} : \mathbf{W} &= (7.28, 0, 0, 0, 1, 0, 1)^T, \\ \left.\begin{array}{r} r < 24, \\ r < 20 + 0.4\cos(22\theta) \end{array}\right\} : \mathbf{W} &= (0, 0.001, 0, 0, 0, 1, 0)^T. \end{aligned}$$

Numerical solutions computed with the MP method are shown in figure 5.17.

5.4 Conclusions

We now conclude our presentation of the six-equation sharp interface method. Our developments to Miller and Puckett's method are based on two ideas– employing modern Riemann solvers developed for the five-equation model, and evolving the internal energy of each material in mixed cells with an implicit update formula. We have presented results from the new numerical method on a wide range of two-material problems, demonstrating a significant improvement in resolution at the interface compared to the five-equation model. The method has proven to be very robust, and was shown to handle pressure ratios of 10^6, and density ratios of 10^4 across the interface in the tin implosion test case. We note that the ghost fluid method presented in chapter 4 was unable to run this test case.

The accuracy of the numerical method was measured both quantitatively and qualitatively. In one-dimensional shock tube problems with known exact solutions, the MP method gave significantly improved accuracy compared to the five-equation diffuse interface method. In quantitative measurements of various shock-interface interaction phenomena, the two methods gave virtually identical results. This verifies that the MP method results in correct shock-interface interactions, since the five-equation model is very well-studied. The MP method appears to result in qualitatively comparable accuracy to other interface sharpening approaches for the five-equation model, such as THINC. The advantage which is unique to our method is that any VOF-based interface tracking method may be applied, and indeed one promising extension to the work in this chapter is the extension to couple the method with the EMOF or MOF interface tracking methods.

As noted previously, a drawback of the MP method is that discrete conservation is lost, although the magnitude of the conservation error was shown to be small. We now outline a set of possible modifications to make the method fully conservative in mass, momentum, and total energy of the mixture.

Obtaining discrete conservation in mass and momentum is relatively straightforward.

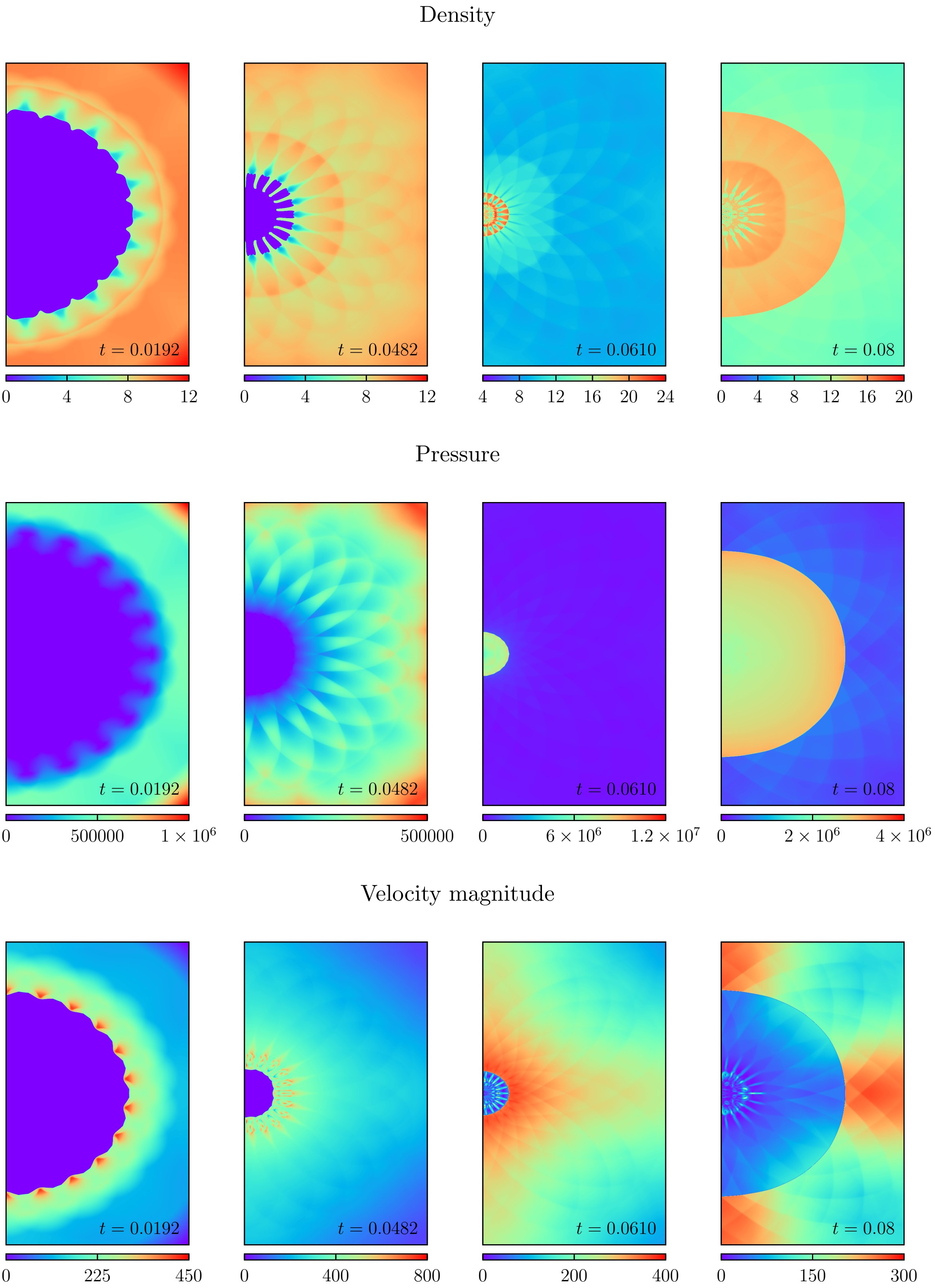


Figure 5.17: Numerical results for the tin implosion test case on a 1000 × 2000 grid.

These quantities are conserved by the update formulae used locally to the interface (equations 5.15b, 5.15c, 5.15d), and are conserved by the Godunov method used in pure material regions. The task is simply to make the flux computation single-valued where the update type changes. Since the updates are equal in the advection-only regime, the differences will generally be very small, and we propose using a simple mean value of the two fluxes on the cell edge where the update type switches from single-material to multi-material.

Obtaining total energy conservation is less trivial due to the source terms present in the six-equation model. Fortunately, a path towards regaining conservation of the six-equation system has already been developed in [126] for diffuse interface methods. The idea is to evolve a redundant seventh equation for total energy of the mixture

$$\frac{\partial E}{\partial t} + \frac{\partial(u(E+p))}{\partial x} = 0, \tag{5.48}$$

and reinitialise the individual internal energies to be consistent with this conservative total energy equation. Using the volume fractions obtained after the pressure relaxation step, and the mixture internal energy obtained from the conservative total energy equation, the mixture equation of state is used to compute mixture pressure. For the stiffened gas case,

$$p = \frac{\rho e - \left(\frac{z^{(1)} p_\infty^{(1)} \gamma^{(1)}}{\gamma^{(1)}-1} + \frac{z^{(2)} p_\infty^{(2)} \gamma^{(2)}}{\gamma^{(2)}-1} \right)}{\frac{z^{(1)}}{\gamma^{(1)}-1} + \frac{z^{(2)}}{\gamma^{(2)}-1}}. \tag{5.49}$$

Then the individual material equations of state are applied to compute an internal energy using this mixture pressure.

CHAPTER 6

Conclusions and future work

This thesis has addressed computational methods for simulating multi-material flow governed by the Euler equations, with a focus on algorithms which employ an interface tracking method. It is comprised of three independent but related areas of study. In this chapter the progress which was achieved in each area is summarised, and an indication of future work is given, in a presentation which is designed to be more self-contained than the short conclusion sections at the end of previous chapters.

6.1 Interface tracking

In chapter 3 the subject of interface tracking methods was explored. The anticipated future application of this work was to highly deformational flows, and as a result we focussed on volume tracking methods, taking advantage of their inherent conservation properties and ability to handle topology changes. The present state-of-the-art in this area is the moment-of-fluid method [82], which tracks material volumes and centroids, but we also considered simpler and less computationally expensive algorithms. Our efforts to improve the accuracy and efficiency of these methods have led to three separate contributions.

1. First, an improvement to the interface reconstruction accuracy of the VOF method was presented. This was achieved by performing a linear least-squares fit to marker particles located on the interface, and was shown to offer better accuracy than conventional VOF reconstruction schemes such as Parker and Youngs [73] and LVIRA [71] on interface advection problems. The common drawbacks of marker particle methods are avoided through a simple re-seeding procedure which is carried out every time step. The computational cost of the method was approximately double that of Parker and Youngs, but we have shown that to obtain a given accuracy on the interface advection test problems it was cheaper to run the new particle-VOF method.

2. The moment-of-fluid method [82] suffers from high computational cost due to the numerical optimisation problem which must be solved during every interface reconstruction. A reformulation of the algorithm was presented in which each objective function evaluation in the optimisation problem was made trivially cheap by pre-computing certain values and mapping the objective function domain onto these pre-computed values. We presented results showing that the accuracy loss due to this approximation is negligible, and obtained a factor of 3 decrease in the run-time of the method, using a 328MB pre-computed table. Due to the way that the objective function is mapped onto pre-computed values, this method is only applicable to grids in which all cell shapes can be expressed as a linear mapping of some base geometry. Although this rules out curvilinear grids, it is applicable to most two-dimensional finite volume grids, including Cartesian, rectilinear, and triangular. The implementation of this approach is simple, and it was used to solve fully-coupled fluid dynamics simulations using the ghost fluid method developed in chapter 4.

3. In our final contribution to interface tracking, the volume fraction update was reframed as a quadratic programming problem. We derived an efficient solution to this problem using the active set method. This mathematical framework results in a physically consistent update regardless of any initial guess, and offers a path towards novel couplings between the VOF method and other interface representations. As an example, we applied this framework to couple the VOF and level set methods in a radically different way to the previous approach [55]. Although the accuracy results obtained with our novel coupling method were significantly better than the traditional approach, this example is flawed in that we were unable to find a satisfactory way to transfer information back from the volume fractions to the level set field. As a result, in highly deformational flows when the level set field suffers from spurious mass loss, it will fail to provide accurate results. Despite this, we hope that the underlying quadratic programming-based update may find future application in other hybrid methods.

6.2 Ghost fluid methods

The ghost fluid method of [38] is a particularly simple approach to the simulation of multi-material flow, in which two separate single-material grids are evolved and a level set field is used to determine which material type exists at each point. Investigating this approach in chapter 4, we first presented evidence that the conservation properties of the ghost fluid method as a whole were significantly affected by the choice of interface tracking method. Motivated by this finding, we reformulated the ghost fluid algorithm such that the solution to the extrapolation equation (which requires a level set field) was eliminated. This then allows volume fraction-based interface tracking methods (which posses superior conservation properties to the level set method) to be applied. This extrapolation-free approach sets ghost states using an inverse-distance weighted average, and reduced the cost of the method by approximately 25% in our implementation. The efficient moment-of-fluid interface tracking method developed in chapter 3 was coupled, and results from shock-interface interaction problems were presented. As well as being cheaper to run, this method was shown to improve the conservation properties of the ghost fluid method by an order of magnitude relative to tracking the interface with a fifth-order accurate level set method. However, the late-time solutions of problems with complicated and unresolvable interface geometries obtained with the efficient moment-of-fluid method were qualitatively similar to those obtained with other conservative, but less accurate methods such as Parker and Youngs' VOF and the CLSVOF method. This suggests that for such flows, choosing a conservative interface tracking method is more important than the precise accuracy of the method.

6.3 Six-equation mixture model methods

In the hierarchy of two-material models derived from the full seven-equation Baer-Nunziato system [26], the five-equation model [28] which assumes pressure and velocity equilibrium between each material is particularly widely used. In the numerical solution of this system the material interface, which behaves like a contact discontinuity, will continuously diffuse outwards unless some form of interface sharpening is applied such as [32–34]. In chapter 5 we presented a numerical method which uses a VOF interface tracking method to maintain a sharp interface indefinitely. Based on a long-neglected approach by Miller and Puckett [35], this method evolves six-equations and then employs a pressure relaxation step to ensure convergence to the five-equation system. We thoroughly overhauled Miller and Puckett's algorithm, deriving a robust positivity-preserving internal energy update and simplifying the algorithm with approximate Riemann solvers developed for the five-equation system. Numerical results are presented on a wide range of one- and two-dimensional test problems with fluids governed by the stiffened gas equation of state. These results confirm the superior interface resolution of the method compared to the diffuse interface five-equation model, and demonstrate the robustness of the method when confronted with large jumps in material properties across the interface. Although non-conservative, the discrete update formula for mixed cells is based on quantities which are computed using conservation laws for total mass, momentum, and energy, and we demonstrate that shock speeds are correctly predicted, and that the conservation error is negligible in practice.

6.4 Future work

The future extensions to the methods presented in this thesis revolve mainly around the MP method used to solve the six-equation system in chapter 5.

Although it was shown that the MP method correctly predicts shock speeds and that the conservation errors were acceptably small, in future work the method will be made fully conservative by implementing the procedure described in section 5.4. Future work will also aim to improve the method by applying the efficient moment-of-fluid interface tracking method developed in chapter 3. This coupling is not trivial due to problems related to tracking the positions of material centroids over the course of the differential compression experienced by materials with different bulk moduli, but we believe it would be worthwhile since moment-of-fluid methods offer significantly improved interface resolution compared to Parker and Youngs.

Our work on interface tracking methods in chapter 3 may be extended in various ways. It would be particularly worthwhile to implement the efficient moment-of-fluid method in three dimensions, as we anticipate that the efficiency improvement would be even more significant in this case. It would also be interesting to apply the quadratic programming update framework to a piecewise-quadratic interface representation, a significant improvement over the current standard piecewise-linear interface reconstruction technique. Since the quadratic interface would no longer be required to exactly satisfy a volume conservation constraint, it may be possible to significantly simplify the operation of this class of interface reconstruction.

Bibliography

[1] Demetrios Christodoulou. The Euler equations of compressible fluid flow. *Bulletin of the American Mathematical Society*, 44(4):581–602, 2007.

[2] L Euler. Principes generaux du mouvement des fluides. *Memoires de l'Academie des Sciences de Berlin*, 11:274–315, 1757.

[3] C E Brennen. *Fundamentals of multiphase flow.* Cambridge University Press, 2005.

[4] S. Balachandar and John K. Eaton. Turbulent Dispersed Multiphase Flow. *Annual Review of Fluid Mechanics*, 42(1):111–133, 2010.

[5] J. M. Picone and J. P. Boris. Vorticity generation by shock propagation through bubbles in a gas. *Journal of Fluid Mechanics*, 189:23–51, 1988.

[6] J.J. Quirk and S Karni. On the dynamics of a shock-bubble interaction. pages 129–163, 1994.

[7] F K Röpke and W Hillebrandt. Full-star type Ia supernova explosion models. *Astronomy & Astrophysics*, 431:635–645, 2005.

[8] R. P J Town and A. R. Bell. Three-dimensional simulations of the implosion of inertial confinement fusion targets. *Physical Review Letters*, 67(14):1863–1866, 1991.

[9] K. K. Haller, Y. Ventikos, D. Poulikakos, and P. Monkewitz. Computational study of high-speed liquid droplet impact. *Journal of Applied Physics*, 92(5):2821–2828, 2002.

[10] Philip T Barton, B Obadia, and Dimitris Drikakis. A conservative level-set based method for compressible solid/fluid problems on fixed grids. *Journal of Computational Physics*, 230(21):7867–7890, 2011.

[11] Young S. Shin. Ship shock modeling and simulation for far-field underwater explosion. *Computers and Structures*, 82(23-26):2211–2219, 2004.

[12] Zhi rong ZHANG, Hui LIU, Song ping ZHU, and Feng ZHAO. Application of CFD in ship engineering design practice and ship hydrodynamics. *Journal of Hydrodynamics*, 18(3 SUPPL.):315–322, 2006.

[13] William C. Moss, Douglas B. Clarke, John W. White, and David A. Young. Hydrodynamic simulations of bubble collapse and picosecond sonoluminescence. *Physics of Fluids*, 6(9):2979–2985, 1994.

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

[15] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1):146–159, 1994.

[16] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335–354, 1992.

[17] C.W Hirt, J.L Cook, and T.D Butler. A lagrangian method for calculating the dynamics of an incompressible fluid with free surface. *Journal of Computational Physics*, 5(1):103–124, 1970.

[18] R Whitehurst. A free Lagrange method for gas dynamics. *Monthly Notices of the Royal Astronomical Societ*, 277(2):655–680, 1995.

[19] CW Hirt, AA Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 253:227–253, 1974.

[20] J. J. Monaghan. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–274, 1992.

[21] Joe J Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005.

[22] S. A. Silling and E. Askari. A meshfree method based on the peridynamic model of solid mechanics. *Computers and Structures*, 83(17-18):1526–1535, 2005.

[23] Shiyi Chen and GD Doolen. Lattice Boltzmann method for fluid flows. *Annual review of fluid mechanics*, (Kadanoff 1986), 1998.

[24] Bernard Larrouturou. How to preserve the mass fractions positivity when computing compressible multi-component flows. *Journal of Computational Physics*, 95(1):59–84, 1991.

[25] Richard Saurel and R Abgrall. A simple method for compressible multifluid flows. *SIAM Journal on Scientific Computing*, 21(3):1115–1145, 1999.

[26] M. R. Baer and J. W. Nunziato. A two-phase mixture theory for the deflagration-to-detonation transition (ddt) in reactive granular materials. *International Journal of Multiphase Flow*, 12(6):861–889, 1986.

[27] A. K. Kapila, R. Menikoff, J. B. Bdzil, S. F. Son, and D. S. Stewart. Two-phase modeling of deflagration-to-detonation transition in granular materials: Reduced equations. *Physics of Fluids*, 13(10):3002–3024, 2001.

[28] Grégoire Allaire, Sébastien Clerc, and Samuel Kokh. A Five-Equation Model for the Simulation of Interfaces between Compressible Fluids. *Journal of Computational Physics*, 181(2):577–616, sep 2002.

[29] Angelo Murrone and Hervé Guillard. A five equation reduced model for compressible two phase flow problems. *Journal of Computational Physics*, 202(2):664–698, 2005.

[30] Marie Billaud Friess and Samuel Kokh. Simulation of sharp interface multi-material flows involving an arbitrary number of components through an extended five-equation model. *Journal of Computational Physics*, 273:488–519, 2014.

[31] Baolin Tian, E. F. Toro, and C. E. Castro. A path-conservative method for a five-equation model of two-phase flow with an HLLC-type Riemann solver. *Computers and Fluids*, 46(1):122–132, 2011.

[32] K K So, X Y Hu, and N A Adams. Anti-diffusion interface sharpening technique for two-phase compressible flow simulations. *Journal of Computational Physics*, 231(11):4304–4323, 2012.

[33] Ratnesh K Shukla, Carlos Pantano, and Jonathan B Freund. An interface capturing method for the simulation of multi-phase compressible flows. *Journal of Computational Physics*, 229(19):7411–7439, 2010.

[34] Keh-Ming Shyue and Feng Xiao. An Eulerian interface sharpening algorithm for compressible two-phase flow: The algebraic THINC approach. *Journal of Computational Physics*, 268:326–354, 2014.

[35] GH Miller and EG Puckett. A high-order Godunov method for multiple condensed phases. *Journal of Computational Physics*, 128(1):134–164, 1996.

[36] J. M. McGlaun, S. L. Thompson, and M. G. Elrick. CTH: A three-dimensional shock wave physics code. *International Journal of Impact Engineering*, 10(1-4):351–360, 1990.

[37] David J Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer methods in Applied mechanics and Engineering*, 99(2):235–394, 1992.

[38] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152(2):457–492, jul 1999.

[39] T. G. Liu, B. C. Khoo, and K. S. Yeo. Ghost fluid method for strong shock impacting on material interface. *Journal of Computational Physics*, 190(2):651–681, 2003.

[40] C. W. Wang, T. G. Liu, and B. C. Khoo. A Real Ghost Fluid Method for the Simulation of Multimedium Compressible Flow. *SIAM Journal on Scientific Computing*, 28(1):278–302, 2006.

[41] R GAFFNEY, JR and H Hassan. Euler calculations for wings using cartesian grids. In *25th AIAA Aerospace Sciences Meeting*, page 356, 1987.

[42] Christiane Helzel, Marsha J Berger, and Randall J LeVeque. A high-resolution rotated grid method for conservation laws with embedded geometries. *SIAM Journal on Scientific Computing*, 26(3):785–809, 2005.

[43] Nandan Gokhale, Nikos Nikiforakis, and Rupert Klein. A dimensionally split Cartesian cut cell method for hyperbolic conservation laws. *Journal of Computational Physics*, 364:186–208, 2018.

[44] D M Causon, D M Ingram, and C G Mingham. A Cartesian cut cell method for shallow water flows with moving boundaries. *Advances in water resources*, m:899–911, 2001.

[45] X. Y. Hu, B. C. Khoo, N. a. Adams, and F. L. Huang. A conservative interface method for compressible flows. *Journal of Computational Physics*, 219:553–578, 2006.

[46] Charles S Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.

[47] Salih Ozen Unverdi and Grétar Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of computational physics*, 100(1):25–37, 1992.

[48] James Glimm, John W. Grove, Xiao Lin Li, Keh-ming Shyue, Yanni Zeng, and Qiang Zhang. Three-Dimensional Front Tracking. *SIAM Journal on Scientific Computing*, 19(3):703–727, 1998.

[49] Cyril W Hirt and Billy D Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981.

[50] W.F Noh and P Woodward. SLIC (simple line interface calculation). *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*, pages 330–340, 1976.

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

[52] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[53] Elin Olsson and Gunilla Kreiss. A conservative level set method for two phase flow. *Journal of Computational Physics*, 210:225–246, 2005.

[54] R. R. Nourgaliev and T. G. Theofanous. High-fidelity interface tracking in compressible flows: Unlimited anchored adaptive level set. *Journal of Computational Physics*, 224(2):836–866, 2007.

[55] Mark Sussman and Elbridge Gerry Puckett. A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, 162:301–337, 2000.

[56] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83–116, 2002.

[57] Jacques Massoni, Richard Saurel, Boniface Nkonga, and Rémi Abgrall. Proposition de méthodes et modèles eulériens pour les problèmes à interfaces entre fluides compressibles en présence de transfert de chaleur: Some models and eulerian methods for interface problems between compressible fluids with heat transfer. *International journal of heat and mass transfer*, 45(6):1287–1307, 2002.

[58] Gael Guennebaud and Benoit Jacob. Eigen v3, 2010.

[59] Leonardo Dagum and Ramesh Menon. OpenMP: an industry standard API for shared-memory programming. *Computational Science & Engineering, IEEE*, 5(1):46–55, 1998.

[60] Steven G. Johnson. The NLopt nonlinear-optimization package.

[61] S K Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 47(3):271–306, 1959.

[62] E F Toro, M Spruce, and W Speares. Restoration of the Contact Surface in the HLL {Riemann} Solver. *Shock Waves, Vol 4*, pages 25–34, 1994.

[63] PL Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 372:357–372, 1981.

[64] P. Colella and PR Woodward. The piecewise-parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics*, 54(1):174–201, 1984.

[65] Bram van Leer. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. *Journal of Computational Physics*, 14(4):361–370, 1974.

[66] Chi-Wang Shu. High-order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD. *International Journal of Computational Fluid Dynamics*, 17(2):107–118, 2003.

[67] E Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction.* Springer-Verlag: Berlin Heidelberg, 2nd ed edition, 1999.

[68] Randall J Leveque. *Finite volume methods for hyperbolic problems.* Cambridge University Press, 2002.

[69] G Strang. On the Construction and Comparison of Difference Schemes. *SIAM Journal on Numerical Analysis*, 5(3):506 – 517, 1968.

[70] William F Noh and Paul Woodward. SLIC (simple line interface calculation). In *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*, pages 330–340. Springer, 1976.

[71] E G Puckett. A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction. In *Proceedings of the Fourth International Symposium on Computational Fluid Dynamics*, pages 933–938, 1991.

[72] James Edward Pilliod Jr and Elbridge Gerry Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199(2):465–502, 2004.

[73] B Parker and D Youngs. *Two and three dimensional Eulerian simulation of fluid flow with material interfaces.* Atomic Weapons Establishment, 1992.

[74] G D Weymouth and Dick K-P Yue. Conservative Volume-of-Fluid method for free-surface simulations on Cartesian-grids. *Journal of Computational Physics*, 229(8):2853–2865, 2010.

[75] Glenn Robert Price. *A Piecewise Parabolic Volume Method for the Numerical Simulation of Interfacial Flows.* PhD thesis, University of Calgary, 2000.

[76] S. V. Diwakar, Sarit K. Das, and T. Sundararajan. A Quadratic Spline based Interface (QUASI) reconstruction algorithm for accurate tracking of two-phase flows. *Journal of Computational Physics*, 228(24):9107–9130, 2009.

[77] Dalton J E Harvie and David F Fletcher. A new volume of fluid advection algorithm: the defined donating region scheme. *International Journal for Numerical Methods in Fluids*, 35(2):151–172, 2001.

[78] Ruben Scardovelli and Stephane Zaleski. Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection. *International Journal for Numerical Methods in Fluids*, 41(3):251–274, 2003.

[79] J López, J Hernández, P Gómez, and F Faura. A volume of fluid method based on multidimensional advection and spline interface reconstruction. *Journal of Computational Physics*, 195(2):718–742, apr 2004.

[80] Qinghai Zhang. On a family of unsplit advection algorithms for volume-of-fluid methods. *SIAM Journal on Numerical Analysis*, 51(5):2822–2850, 2013.

[81] Raphaël Comminal, Jon Spangenberg, and Jesper Henri Hattel. Cellwise conservative unsplit advection for the volume of fluid method. *Journal of Computational Physics*, 283:582–608, 2015.

[82] Vadim Dyadechko and Mikhail Shashkov. Moment-of-fluid interface reconstruction. *Los Alamos report LA-UR-05-7571*, 836, 2006.

[83] F Xiao, Y Honma, and T Kono. A simple algebraic interface capturing scheme using hyperbolic tangent function. *International Journal for Numerical Methods in Fluids*, 48(9):1023–1040, 2005.

[84] MACIEJ Marek, WOJCIECH Aniszewski, and A Boguslawski. Simplified volume of fluid method (SVOF) for two-phase flows. *TASK QUARTERLY*, 12(3):255–265, 2008.

[85] 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. *Journal of Scientific Computing*, 54(2-3):454–491, 2013.

[86] Akio Kawano. A simple volume-of-fluid reconstruction method for three-dimensional two-phase flows. *Computers and Fluids*, 134-135(2):130–145, 2016.

[87] Zhaoyuan Wang, Jianming Yang, and Frederick Stern. A simple and conservative operator-splitting semi-Lagrangian volume-of-fluid advection scheme. *Journal of Computational Physics*, 231(15):4981–4992, 2012.

[88] W. Aniszewski, T. Ménard, and M. Marek. Volume of Fluid (VOF) type advection methods in two-phase flow: A comparative study. *Computers & Fluids*, 97:52–73, jun 2014.

[89] Eugenio Aulisa, Sandro Manservisi, and Ruben Scardovelli. A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows. *Journal of Computational Physics*, 188(2):611–639, 2003.

[90] J. López, J. Hernández, P. Gómez, and F. Faura. An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. *Journal of Computational Physics*, 208:51–74, 2005.

[91] J M Gablonsky and C T Kelley. A Locally-Biased form of the DIRECT Algorithm. *Journal of Global Optimization*, 21(1):27–37, sep 2001.

[92] Hyung Taek Ahn and Mikhail Shashkov. Multi-material interface reconstruction on generalized polyhedral meshes. *Journal of Computational Physics*, 226(2):2096–2132, 2007.

[93] Yisen Guo, Yongsheng Lian, and Mark Sussman. Numerical Simulation of Droplet Impact on Dry Solid Surfaces Using the Moment of Fluid Method. *Bulletin of the American Physical Society*, 58, 2013.

[94] Hyung Taek Ahn, Mikhail Shashkov, and Mark A Christon. The moment-of-fluid method in action. *communications in Numerical Methods in Engineering*, 25(10):1009–1018, 2009.

[95] Matthew Jemison, Mark Sussman, and Mikhail Shashkov. Filament Capturing with the Multimaterial Moment-of-Fluid Method. *Journal of Computational Physics (submitted)*, 2013.

[96] Ryan N Hill and Mikhail Shashkov. The Symmetric Moment-of-Fluid interface reconstruction algorithm. *Journal of Computational Physics*, 249:180–184, 2013.

[97] M. J. D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, 7(1998):287, 1998.

[98] Matthew Jemison. *An asymptotically preserving method for multiphase flow*. PhD thesis, The Florida State University, 2014.

[99] QINGHAI Zhang and AARON Fogelson. MARS: An analytic framework of interface tracking via mapping and adjusting regular semi-algebraic sets. *SIAM J. Numer. Anal., submitted*, 2014.

[100] Raphaël Comminal, Jon Spangenberg, and Jesper Henri Hattel. Cellwise conservative unsplit advection for the volume of fluid method. *Journal of Computational Physics*, 283:582–608, feb 2015.

[101] Qinghai Zhang. On a family of unsplit advection algorithms for volume-of-fluid methods. *SIAM J. Numer. Anal., submitted*, 84103(5):2822–2850, 2013.

[102] Jorge Nocedal and Stephen J Wright. Quadratic programming. *Numerical optimization*, pages 448–492, 2006.

[103] Guang-Shan Jiang and Danping Peng. Weighted eno schemes for hamilton–jacobi equations. *SIAM Journal on Scientific computing*, 21(6):2126–2143, 2000.

[104] David Adalsteinsson and James a. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.

[105] Yen-Hsi Richard Tsai, Li-Tien Cheng, Stanley Osher, and Hong-Kai Zhao. Fast Sweeping Algorithms for a Class of Hamilton–Jacobi Equations. *SIAM Journal on Numerical Analysis*, 41(2):673–694, 2003.

[106] William J. Rider and Douglas B. Kothe. Reconstructing Volume Tracking. *Journal of Computational Physics*, 141(2):112–152, apr 1998.

[107] Steven T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335–362, 1979.

[108] Charles A Doswell III. A kinematic analysis of frontogenesis associated with a nondivergent vortex. *Journal of the atmospheric sciences*, 41(7), 1983.

[109] E. F. Toro and V. A. Titarev. ADER schemes for scalar non-linear hyperbolic conservation laws with source terms in three-space dimensions. *Journal of Computational Physics*, 202(1):196–215, 2005.

[110] Liang Xu and Tiegang Liu. Accuracies and conservation errors of various ghost fluid methods for multi-medium Riemann problem. *Journal of Computational Physics*, 230(12):4975–4990, 2011.

[111] Shiv Kumar Sambasivan and HS Udaykumar. Ghost fluid method for strong shock interactions part 1: Fluid-fluid interfaces. *Aiaa Journal*, 47(12):2907–2922, 2009.

[112] X. Y. Hu and B. C. Khoo. An interface interaction method for compressible multifluids. *Journal of Computational Physics*, 198(1):35–64, 2004.

[113] Liang Xu, Chengliang Feng, and Tiegang Liu. Practical Techniques in Ghost Fluid Method for Compressible Multi-Medium Flows. *Communications in Computational Physics*, 20(03):619–659, 2016.

[114] Rémi Abgrall and S Karni. Computations of compressible multifluids. *Journal of Computational Physics*, 169(2):594–623, may 2001.

[115] Charbel Farhat, Arthur Rallu, and Sriram Shankaran. A higher-order generalized ghost fluid method for the poor for the three-dimensional two-phase flow computation of underwater implosions. *Journal of Computational Physics*, 227(16):7674–7700, 2008.

[116] Bruno Lombard and Rosa Donat. The explicit simplified interface method for compressible multicomponent flows. *SIAM Journal on Scientific Computing*, pages 1–23, 2005.

[117] Duc Nguyen, Frederic Gibou, and Ronald Fedkiw. A Fully Conservative Ghost Fluid Method & Stiff Detonation Waves. *12th International Detonation Symposium*, pages 1–8, 2002.

[118] Wei Liu, Li Yuan, and Chi Wang Shu. A Conservative modification to the ghost fluid method for compressible multiphase flows. *Communications in Computational Physics*, 10(4):785–806, 2011.

[119] Alexandre Joel Chorin. Random choice solution of hyperbolic systems. *Journal of Computational Physics*, 22(4):517–533, 1976.

[120] Robert R. Nourgaliev, T. Nam Dinh, and Theo G. Theofanous. Adaptive characteristics-based matching for compressible multifluid dynamics. *Journal of Computational Physics*, 213(2):500–529, 2006.

[121] Wurigen Bo and John W. Grove. A volume of fluid method based ghost fluid method for compressible multi-fluid flows. *Computers & Fluids*, 90:113–122, feb 2014.

[122] S. Kokh and F. Lagoutière. An anti-diffusive numerical scheme for the simulation of interfaces between compressible fluids by means of a five-equation model. *Journal of Computational Physics*, 229(8):2773–2809, 2010.

[123] David I Ketcheson, Matteo Parsani, and Randall J LeVeque. High-order wave propagation algorithms for hyperbolic systems. *SIAM Journal on Scientific Computing*, 35(1):A351–A377, 2013.

[124] Daniel P. Garrick, Wyatt A. Hagen, and Jonathan D. Regele. An interface capturing scheme for modeling atomization in compressible flows. *Journal of Computational Physics*, 344:260–280, 2017.

[125] M. M. Francois, M. J. Shashkov, T. O. Masser, and E. D. Dendy. A comparative study of multimaterial Lagrangian and Eulerian methods with pressure relaxation. *Computers and Fluids*, 83:126–136, 2013.

[126] Richard Saurel, Fabien Petitpas, and Ray A. Berry. Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures. *Journal of Computational Physics*, 228(5):1678–1712, 2009.

[127] Daniel P Garrick, Mark Owkes, and Jonathan D Regele. A finite-volume HLLC-based scheme for compressible interfacial flows with surface tension. *Submitted to Journal of Computational Physics*, 339:46–67, 2016.

[128] P Colella, HM Glaz, and RE Ferguson. Multifluid algorithms for Eulerian finite difference methods. *Private Communication*, pages 1–24, 1989.

[129] J. F. Haas and B Sturtevant. Interaction of Weak Shock Waves With Cylindrical And Spherical Gas Inhomogeneities. *Journal of Fluid Mechanics*, 181:41–76, 1987.

[130] James J Quirk and S Karni. On the dynamics of a shock-bubble interaction. *Journal of Fluid Mechanics*, 318(1996):129–163, 1996.

[131] R D Richtmyer. Taylor Instabilities in Shock Acceleration of Compressible Fluids. *Communications on Pure and Applied Mathematics*, XIII:297–319, 1960.

APPENDIX A

Stiffened gas exact Riemann solver

The Riemann problem is the simplest non-trivial initial value problem for the Euler equations. It is necessary to find the star-state pressure, velocity and densities as well as all three wave speeds (and perhaps the state inside a rarefaction). For the initial states given by the vectors of primitive variables $\mathbf{W}_{L|R} = (\rho_{L|R}, u_{L|R}, p_{L|R})^T$, and stiffened gas parameters $\gamma_{L|R}$, $p_{\infty L|R}$, the star-state pressure can be derived by using either the Rankine-Hugoniot conditions or the isentropic law to connect both initial states to the star state. For reference to the notation used in the following derivation, or general background, see [67].

From the stiffened gas equation of state,

$$p = (\gamma - 1)\rho e - \gamma p_\infty, \tag{A.1}$$

we now obtain the isentropic law

$$\frac{1}{\rho}\mathrm{d}\rho = \frac{1}{\gamma}\frac{1}{p + p_\infty}\mathrm{d}p, \tag{A.2}$$

which leads to the sound speed

$$a = \sqrt{\frac{\gamma(p + p_\infty)}{\rho}}. \tag{A.3}$$

The star-state pressure is then given by the root of the algebraic equation

$$f(p, \mathbf{W}_L, \mathbf{W}_R) \equiv f_L(p, \mathbf{W}_L) + f_R(p, \mathbf{W}_R) + u_R - u_L = 0 \tag{A.4}$$

where

$$f_L(p,\mathbf{W}_L) = \begin{cases} (p-p_L)\left[\frac{A_L}{B_L+p+p_{\infty L}}\right]^{\frac{1}{2}} & \text{if } p > p_L \\ \\ \frac{2a_L}{\gamma_L-1}\left[\left(\frac{p+p_{\infty L}}{p_L+p_{\infty L}}\right)^{\frac{\gamma_L-1}{2\gamma_L}}-1\right] & \text{if } p \le p_L \end{cases} \tag{A.5}$$

$$f_R(p,\mathbf{W}_R) = \begin{cases} (p-p_R)\left[\frac{A_R}{B_R+p+p_{\infty R}}\right]^{\frac{1}{2}} & \text{if } p > p_R \\ \\ \frac{2a_R}{\gamma_R-1}\left[\left(\frac{p+p_{\infty R}}{p_R+p_{\infty R}}\right)^{\frac{\gamma_R-1}{2\gamma_R}}-1\right] & \text{if } p \le p_R \end{cases} \tag{A.6}$$

$$A_{L|R} = \frac{2}{(\gamma_{L|R}+1)\rho_{L|R}} \tag{A.7}$$

$$B_{L|R} = \frac{\gamma_{L|R}-1}{\gamma_{L|R}+1}(p_{L|R}+p_{\infty L|R}). \tag{A.8}$$

Equation A.4 is solved numerically using a Newton-Raphson iterative procedure, where the initial guess is obtained using the solution to the linearised primitive-variable form of the Euler equations:

$$p_0 = \max\left(10^{-6},\ \frac{1}{2}(p_L+p_R) - \frac{1}{8}(u_R-u_L)(\rho_L+\rho_R)(a_L+a_R)\right). \tag{A.9}$$

The star state velocity is then

$$u_* = \frac{1}{2}(u_L+u_R) + \frac{1}{2}(f_R(p_*) - f_L(p_*)). \tag{A.10}$$

The value of the star-state densities depends on the character of the 1st and 3rd waves in the solution.

$$\rho_{*L} = \begin{cases} \rho_L\left[\frac{2\gamma_L p_{\infty L}+(\gamma_L+1)p_*+(\gamma_L-1)p_L}{2(p_L+\gamma_L p_{\infty L})+(\gamma_L-1)p_*+(\gamma_L-1)p_L}\right] & \text{if } p_* > p_L \\ \\ \rho_L\left(\frac{p_*+p_{\infty L}}{p_L+p_{\infty L}}\right)^{\frac{1}{\gamma_L}} & \text{if } p_* \le p_L \end{cases}, \tag{A.11}$$

$$\rho_{*R} = \begin{cases} \rho_R\left[\frac{2\gamma_R p_{\infty R}+(\gamma_R+1)p_*+(\gamma_R-1)p_R}{2(p_R+\gamma_R p_{\infty R})+(\gamma_R-1)p_*+(\gamma_R-1)p_R}\right] & \text{if } p_* > p_R \\ \\ \rho_R\left(\frac{p_*+p_{\infty R}}{p_R+p_{\infty R}}\right)^{\frac{1}{\gamma_R}} & \text{if } p_* \le p_R \end{cases}. \tag{A.12}$$

To determine the state on the $\frac{x}{t}=0$ characteristic we must compute all wave speeds in the solution, and the state inside a rarefaction fan. The central contact wave speed (u_*)

is already known. The shock speeds are

$$S_L = u_L - \frac{1}{\rho_L}\left(\frac{B_L + p_{\infty L} + p_*}{A_L}\right)^{\frac{1}{2}}, \tag{A.13}$$

$$S_R = u_R + \frac{1}{\rho_R}\left(\frac{B_R + p_{\infty R} + p_*}{A_R}\right)^{\frac{1}{2}}. \tag{A.14}$$

The speeds of a left rarefaction head and tail are

$$S_{HL} = u_L - a_L, \tag{A.15}$$

$$S_{TL} = u_* - a_{*L}, \tag{A.16}$$

and for a right rarefaction:

$$S_{HR} = u_R + a_R, \tag{A.17}$$

$$S_{TR} = u_* + a_{*R}, \tag{A.18}$$

where

$$a_{*L|R} = a_{L|R}\left(\frac{p_* + p_{\infty L|R}}{p_{L|R} + p_{\infty L|R}}\right)^{\frac{\gamma_{L|R}-1}{2\gamma_{L|R}}}. \tag{A.19}$$

Finally, the state inside a rarefaction fan is given by

$$\mathbf{W}_{LFAN} = \begin{pmatrix} \rho_L\left[\frac{2}{\gamma_L+1} + \frac{\gamma_L-1}{(\gamma_L+1)a_L}\left(u_L - \frac{x}{t}\right)\right]^{\frac{2}{\gamma_L-1}} \\ \\ \frac{2}{\gamma_L+1}\left[a_L + \frac{\gamma_L-1}{2}u_L + \frac{x}{t}\right] \\ \\ (p_L + p_{\infty L})\left[\frac{2}{\gamma_L+1} + \frac{\gamma_L-1}{(\gamma_L+1)a_L}\left(u_L - \frac{x}{t}\right)\right]^{\frac{2\gamma_L}{\gamma_L-1}} - p_{\infty L} \end{pmatrix}, \tag{A.20}$$

$$\mathbf{W}_{RFAN} = \begin{pmatrix} \rho_R\left[\frac{2}{\gamma_R+1} - \frac{\gamma_R-1}{(\gamma_R+1)a_R}\left(u_R - \frac{x}{t}\right)\right]^{\frac{2}{\gamma_R-1}} \\ \\ \frac{2}{\gamma_R+1}\left[-a_R + \frac{\gamma_R-1}{2}u_R + \frac{x}{t}\right] \\ \\ (p_R + p_{\infty R})\left[\frac{2}{\gamma_R+1} - \frac{\gamma_R-1}{(\gamma_R+1)a_R}\left(u_R - \frac{x}{t}\right)\right]^{\frac{2\gamma_R}{\gamma_R-1}} - p_{\infty R} \end{pmatrix}. \tag{A.21}$$