Stony Brook University

Enhanced 3D Front Tracking Method with Locally Grid Based Interface Tracking

A Dissertation Presented

by

Yuanhua Li

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

August 2007

Stony Brook University

The Graduate School

Yuanhua Li

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

Xiaolin Li - Dissertation Advisor
Professor
Department of Applied Mathematics and Statistics

James Glimm - Chairman of Defense
Professor
Department of Applied Mathematics and Statistics

Yan Yu - Member
Research Professor
Department of Applied Mathematics and Statistics

Foluso Ladeinde - Outside Member
Professor
Mechanical Engineering

Roman Samulyak - Outside Member
Scientist
Computational Science Center
Brookhaven National Laboratory

This dissertation is accepted by the Graduate School.

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Enhanced 3D Front Tracking Method with Locally Grid Based Interface Tracking

by

Yuanhua Li

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

2007

We present a new interface tracking algorithm for 3D front tracking called Locally Grid Based tracking (LGB), which is demonstrated to be a significant improvement to the existing front tracking method. It combines the best features of two previous 3D interface tracking algorithms. To be specific, it combines the robustness of Grid Based tracking with the accuracy of Grid Free tracking. We report the implementation of this algorithm and the comparison study with publicly distributed interface codes (the level set method), with published performance results (VOF and other methods) and with previous versions of front tracking. We also explore the application of this algorithm in the study of mean curvature flow and 3D chaotic fluid mixing problems.

To my parents, sisters and my loving wife

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my profound gratitude to my advisor, Dr. Xiaolin Li, for suggesting this important and exciting thesis topic and for his advice, support and guidance toward my Ph.D. degree. He taught me not only the way to do scientific research, but also the way to become a professional scientist.

I am deeply indebted to Dr. James Glimm for his mentorship, guidance and sharing his wisdom on conducting scientific research. His scientific vigor and dedication makes him a lifetime role model for me.

I would like to thank all my groupmates and friends, with whom I collaborated on most of my research works. Especially, I want to acknowledge Dr. Xinfeng Liu whose work significantly improved the Locally Grid Based Method and extended its application into fluid mixing problems.

I would also like to thank Dr. Roman Samulyak and Dr. Foluso Ladeinde for being on my dissertation committee. I would like to thank Dr. Yan Yu, Brian Fix and other Galaxy admins, who spend tremendous time to maintain the super computing facilities on which most of my work was performed.

Most of all, I would like to thank my dearly be loved wife Yue, who is willing to share happiness and sadness with me. She makes me the luckiest man on the earth.

Throughout my academic journey, the constant support from my families has always motivated me to strive forward. Their unconditional love has never been affected by the physical distance between us. My dissertation is dedicated to them.

Chapter 1

Introduction

We begin Chapter 1 with a background review of interface tracking methods, the front tracking in *FronTier*, then we introduce a new interface tracking algorithm for 3D front tracking.

1.1 Interface Tracking Methods

Numerical modelling of multi-phase and free-surface flows is an important topic with a variety of applications in different disciplines in engineering, applied sciences and fundamental physics. Many numerical methods have been proposed and used to model complex 2D and 3D flows exhibiting topology changes. Surveys of different interface tracking methods can be found in [37, 89, 104, 107, 114]. These interface tracking methods can be roughly divided into three groups: Lagrangian methods (or sometimes called front tracking methods), Eulerian methods (or sometimes called front capturing methods) and Hybrid Methods.

Lagrangian methods usually maintain an explicit interface and evolve the interface by following the flow along characteristics, either using surface markers [44, 46, 96, 115, 116] or marker particles with volume particles possessing an identity or color [18, 100]. Generally, Lagrangian methods can maintain filamentary structures

better than Euler method and conserve adequately mass even without explicit volume conservation. On the other hand they could be rather expensive as the number of volume particles increases, or require complex algorithms to resolve topological changes in the explicitly represented interface.

Popular Eulerian front capturing methods, where the interface is defined in some way on a fixed grid, include volume tracking or volume-of-fluid (VOF), level set and phase field methods. They are simple to implement since they are based on a single scalar function defined on the computational domain and do not require explicit tracking of the interface geometries as in Lagrangian methods. Topological changes in the interface could be handled naturally. However, they generally suffer from excessive numerical dispersion or mass loss. For detailed comparison study of several interface tracking methods, we refer to the papers [101, 102] by Rider and Kothe and several other papers [30, 35, 66].

In order to take advantages of different tracking methods, a number of hybrid methods also appeared [5, 35, 111]. The Locally Grid Based interface tracking method that we will discuss in detail in following sections falls into this group. In [111] a mixed Eulerian scheme combines the good mass conservation property of VOF methods with the accurate surface curvature representation via finite difference of the level set function. In [35] Lagrangian disconnected marker particles are randomly positioned near the interface and are passively advected by the flow in order to rebuild the level set function in underresolved zones, such as high curvature regions and near filaments, where the level set approach regularize excessively the interface structure and loses mass.

1.2 Front Tracking Method in *FronTier*

The front tracking method is an adaptive computational method that provides sharp resolution of wave fronts by tracking the interfaces between distinct materials. It represents interfaces explicitly as lower dimensional meshes moving through a volume-filling grid and treat them as interior moving boundaries. Early proposals for front tracking are introduced in Richtmyer and Morton [99], and its realization in one space dimension can be found in [65, 92, 99, 112]. Since early 1980s, extensive work on front tracking method and its application in two space dimensions has been done by J. Glimm and his coworkers [49, 60, 63]. This method has been implemented in a robust and validated code *FronTier*, which is characterized by its accuracy and high resolution for both solutions and discontinuity locations on significantly coarse grids. Front tracking has matured as a numerical scheme in one and two space dimensions in last few decades. Great effort has been added on its extension to three space dimensional flows [46, 47]. The major challenge in its extension to three dimensions lies in the maintenance of the dynamically evolving fluid interface, which requires the ability to detect and resolve changes in the topology of a moving front.

A primary objective for any front tracking code is to allow simulations that distinguish different fluid or material components. This is accomplished via the notion of an interface, which describes the topology and geometry of the surfaces that separate the fluids. Thus the interface tracking algorithm is one of the most critical parts of any front tracking code. We will give a brief introduction of the *FronTier* code in this section. Please refer to [46, 47] for details about *FronTier* code. Figure 1.1, which is taken from [46], shows the flow chart of *FronTier* front tracking.

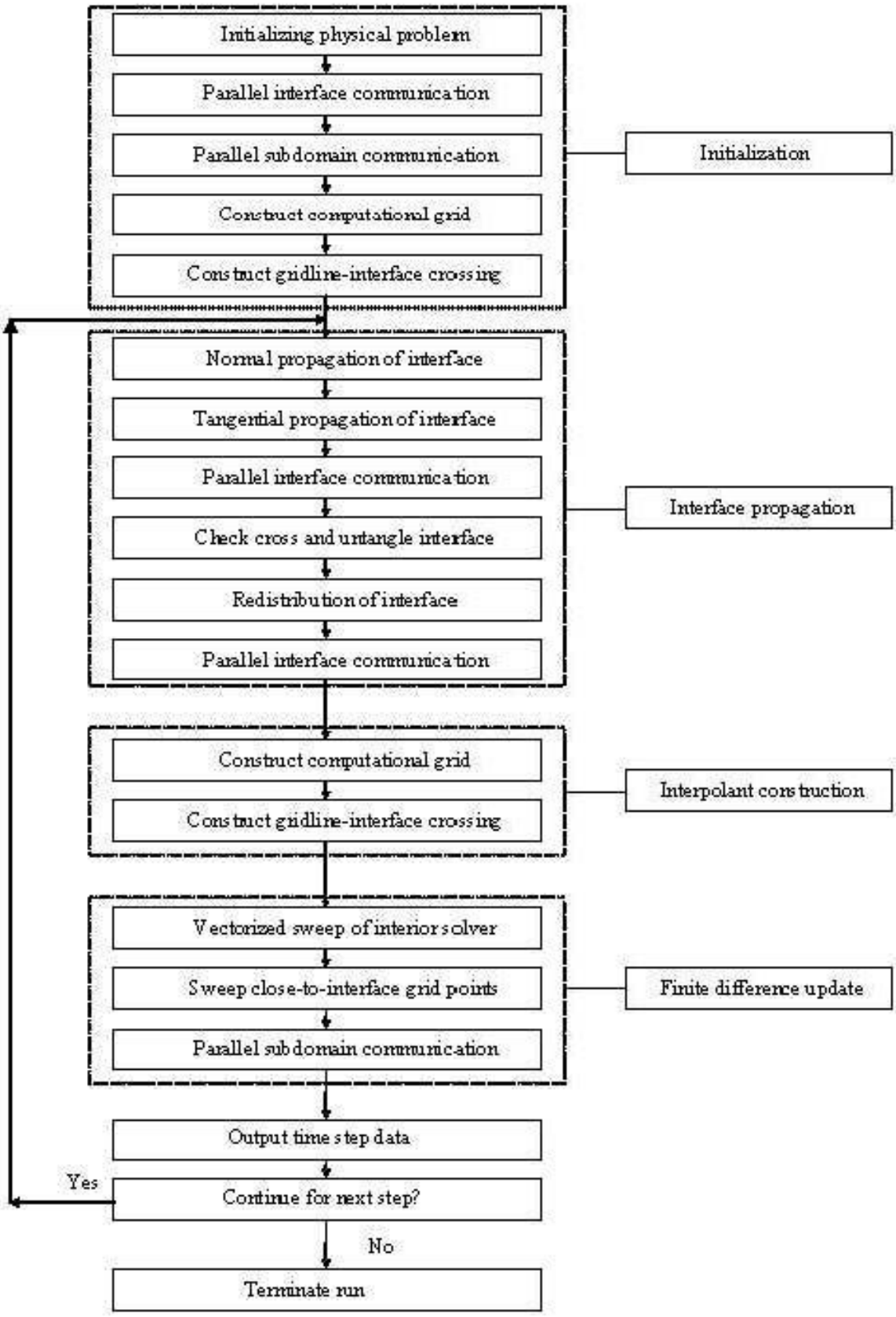


Figure 1.1: *FronTier* Flow Chart

1.2.1 Mathematical Formulation

FronTier models physical problems by solving systems of conservation laws. The basic physical principles of conservation of mass, momentum, and energy can be expressed mathematically in the system of conservation laws. Depending on the physical model, the system of conservation laws can be equations of almost any type, such as elliptic, parabolic, hyperbolic, or even mixtures of these types. The conservation principles state that for an extensive quantity throughout a domain, the total amount of quantity within the domain, which in general varies with time, must balance the flux through the boundary of the domain during the time period.

The system of hyperbolic conservation laws can be written in the following differential form:

$$U_t + \nabla \cdot F(U) = 0, \quad F = (F_1, ..., F_d) \ \ in \ \mathcal{D} = R^d \times R^+, \tag{1.1}$$

where the function U is defined to be $U : (\mathbf{x}, t) \in R^d \times R^+ \rightarrow U(\mathbf{x}, t) \in R^p$; $F_j(U) = (F_{1j}(U), ..., F_{pj}(U))^T \in R^p$.

If we integrate equation 1.1 with respect to x and t in domain $\mathcal{D}$, we obtain the integral form of conservation law

$$\frac{d}{dt} \int_{\mathcal{D}} U \ d\mathcal{D} = \int_{\partial \mathcal{D}} F(U) \cdot n \ dS, \tag{1.2}$$

where S is the boundary of $\mathcal{D}$, and n is the outward unit normal to S.

We shall study the numerical solution to the *Cauchy problem* for this system: Given an initial condition

$$U(\mathbf{x}, 0) = U_0(\mathbf{x}), \quad \mathbf{x} \in R^d \tag{1.3}$$

which is piecewise smooth, find a solution U for the system (1.1) which satisfies the initial condition.

Definition 1.2.1 *The function $U \in L^{\infty}_{loc}(R^d \times [0, +\infty))^p$ is called a weak solution of the Cauchy problem (1.1) and (1.3) if $U(x,t) \in \Omega$, where Ω is an open subset of R^p, satisfies*

$$\int_0^{\infty} \int_{R^d} (U \frac{\partial \phi}{\partial t} + \sum_{j=1}^{d} F_j(U) \frac{\partial \phi}{\partial x_j}) dx dt = - \int_{R^d} U_0(x) \phi(x,0) dx. \tag{1.4}$$

for every smooth ϕ with compact support.

We can see from the above definition that the weak solution can be discontinuous.

1.2.2 Representation of fronts

A geometry package, called the interface library, is used in *FronTier* for the description and manipulation of interfaces. Details regarding this geometry package of *FronTier* can be found in [46–48, 55, 61]. We will only give a brief summary of some basic terminology here.

One of the important features of front tracking methods is the usage of lower dimensional grids, called fronts, which fit the discontinuities in the numerical solutions. The location of a set of fronts at a given time is described by an interface. Geometrically an interface is a set of geometric objects which consists of discrete representations of points, curves and surfaces. Both surfaces and curves are examples of discrete manifolds with boundaries. An interface consists of a set of surfaces. The boundaries of surfaces are curves, while the boundaries of curves are called nodes. A curve is a connected oriented piecewise linear list of bonds, which are connectors

between adjacent points. A surface is a connected oriented piecewise linear collection of triangles, which are in turn connectors between three adjacent points. Both bonds and triangles are linking objects in the sense that they contain pointers to their neighbors. Each bond points to both the previous and following bonds that share its endpoints. Similarly, triangles share a pair of points along a common side with their neighbor and contain pointers to that neighbor's address. A valid interface is one where each surface and curve is piecewise smooth; surfaces intersect only along curves; and curves intersect only at nodes. We say that such an interface is untangled. During front propagation, front intersections are produced due to wave interactions and require special treatment to resolve the interaction and untangle the interface.

For the three-dimensional flows considered here, we assume the interface is embedded in a compact subset of $\mathbf{R}^3$ (in fact a rectangular domain) called the computational domain. The surfaces divide this domain into a set of connected components. The surfaces together with the fixed boundaries of the computational domain form the boundaries of the fluid components. A surface is oriented and a component label is assigned to each of its sides. Topological consistency requires that this label be identical for every surface side bounding a given connected domain.

The front tracking method has showed its advantage in the computation of several important physical problems, such as the study of fluid interface instabilities [43, 52, 62, 69, 118], providing the first or the only physically validated simulation for some important fluid instability problems.

1.2.3 Propagation of Interface

There are three different interface tracking methods implemented in *FronTier*: Grid Free method, Grid Base method and newly implemented Locally Grid Base method. They propagate interface points in the same way but differ in how they

reconstruct the new interface and resolve tangles in the new interfaces. We will give more details about these three tracking methods in Chapter 2.

The interface points are first propagated normally. By computing the solution to the local Riemann problem with initial states being those on either side of the interface point and using the method of characteristics, a wave speed and a new position for the interface point are determined.

The interface states are then updated by a tangential sweep, which uses a chosen interior solver with a stencil centered at the new interface point.

After interface propagation, we get a new interface at new time level. Interface tracking algorithms is responsible for resolving the possible tangles in the new interface and optimization of the interface mesh.

1.2.4 Interior Solver

A connected region in the domain separated by the interface is represented by a component. Therefore, each grid node is associated with a specific component in addition to the state variables. The interior states are updated by finite difference schemes.

Computations near the fluid interface use ghost cells [51, 54] to avoid crossing the interface, keeping the different fluid computations entirely separate. The ghost cell method avoids the interpolation across the interface. A new conservative front tracking scheme has been implemented in *FronTier* recently [82], which eliminates the need for ghost cells.

1.3 Locally Grid Based Interface Tracking

In order to combine the advantages of both Lagrangian methods and Eulerian methods, we propose a new interface tracking method for 3D front tracking. This method is called the Locally Grid Based Method. It is a hybrid method inherited from two existing tracking methods in *FronTier*.

1.4 Dissertation Organization

The rest of my thesis is organized as follows: In Chapter 2 we first briefly describe two existing tracking methods (Grid Free tracking and Grid Based tracking) and then present the newly implemented interface tracking method (Locally Grid Based tracking). In Chapter 3 we extend the comparison study of Rider and Kothe to our Locally Grid Based method. In Chapter 4 we explore the applications of our enhanced front tracking method in the numerical study of Mean Curvature Flow and Chaotic Mixing problems. In Chapter 5, we present the study of numerical simulation with limited mass diffusion using an enhanced front tracking method.

Chapter 2

Locally Grid Based Interface Tracking

2.1 Two Existing 3D Interface tracking Methods

We have two existing 3D interface tracking methods in *FronTier* : Grid Free tracking (Lagrangian), and Grid Based tracking (Eulerian).

2.1.1 Grid Free Method

The Grid Free tracking resembles boundary integral simulations [6, 70, 85, 97], in that they both describe the moving surface by markers which are treated as Lagrangian points. However essentially they are different since they use different methods to evolve the interface. More details of the comparison of these two numerical methods can be found in [74, 119]. The grid-free front tracking interface method consists of two parts, surface retriangulation and untangling a set of intersecting interfaces.

Grid-free retriangulation

As in the ALE methods, periodic retriangulation of the interface is used to maintain uniformity of the triangles that form the surfaces. At a frequency specified

by the user, the interface elements (triangles) with size out of range (either too large or too small) or with a bad aspect ratio with respect to a set of user defined tolerances are put into a queue. Several basic operations (splitting, flipping, deleting) are applied iteratively to the triangles in the queue until no triangles need to be retriangulated. During each iteration: (1) Large triangles are divided by splitting them along their longest side. At the same time the neighboring triangle (or bond if the split side lies on a surface boundary curve) is also split. The split triangles are removed from the queue, and the resulting new triangles are added to the processing queue. (2) The adjacent pairs of small triangles with a common short side will be identified and the indicated side is collapsed and the degenerate triangles are deleted. (3) An additional elementary operation is to flip the interior edge of two adjacent triangles (which thus form a diamond), so that the diamond with its four vertices is not changed, but its triangularization is reversed. In practice the processing queue will be empty after a relatively few number of iterations.

Grid-free untangle

The second part of the grid-free algorithm is to resolve any intersections that have been produced during front propagation. It contains three main steps: (1) use a robust and efficient triangle intersection detection algorithm [64] to find the intersecting triangle pairs. Topological grid is introduced to avoid the check for intersects of triangles that are spatially distant from each other. This considerably speeds up the algorithm. (2) retriangulate intersected triangles. For each intersecting triangle on the two intersecting surfaces, the intersection bonds divide it into two polygonal parts each bounded by the bonds of the crossing curve and the original triangle sides. We triangulate these polygonal parts using the constrained planar Delauney triangulation method of Chew [22] as implemented for the case of a single

polygon by Li [78]. For the details of this algorithm please refer to the paper of Chew [22]. (3) delete the unphysical surfaces. After the first two steps, we can get an untangled interface that satisfies all the requirements for a valid interface except for the consistency of its embedding into the computational domain. That is because for the surfaces meeting along the intersection curves, the component numbers of the common side of the surfaces meeting at this curve are inconsistent. It is thus impossible to assign components to the regions of the computation domain defined by this interface. The only thing needed to get a consistent untangled interface is to delete the unphysical surfaces. Methods are designed to identify the unphysical surfaces and delete them [47, 49].

2.1.2 Grid Based Method

Another method to resolve changing interface topology is to reconstruct the interface using microtopology within each rectangular grid block cell on a user specified lattice. The scheme is divided into three steps: (1) Compute the crossings of the interface and the grid block edges. At each crossing of the interface and the grid block edges we assign labels recording the components on both sides of the interface. (2) Determine component values at the grid block corners and eliminate inconsistent crossings. The crossings divide each edge into a set of subintervals. In regions where the interface is tangled, some subintervals will have different component labels at their opposite endpoints. We process each grid block edge to eliminate crossings that produce inconsistencies in the assignment of component values to the subintervals. (3) Reconstruct a new interface using the remaining consistent grid block edge and interface crossings. The reconstruction process consists of two steps: The reconstruction of an interface segment within a grid cell and the assembling of the single block surface elements into global surfaces. Detailed description about this method can be

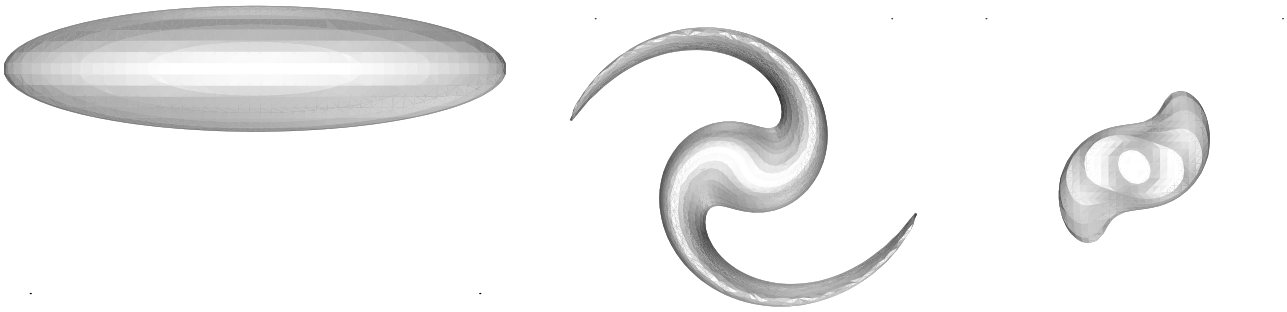


Figure 2.1: *Comparison of GF and GB interface propagation using a common* 80^3 *mesh. Left: Common initial initial conditions (an elongated ellipsoid). Center: Lagrangian (grid free) propagation. Right: Eulerian (grid based) propagation. The plots are after 500 steps in a spiraling velocity field. which is after about 1.6 rounds of revolution at the center and 2.9 rounds of revolution at the outer edge (the angular velocity is a linear function of* r*).*

found in [47].

2.1.3 Grid Free vs. Grid Based Method

Both the grid-free and the grid-based method described above have advantages as well as deficiencies. The grid-free method produces a high quality distribution of triangle sizes and shapes and accurately controls numerical diffusion. It suffers from being complex and subject to failure when the interface is complex. The grid-based method is over-diffusive, which is manifested as an over-smoothing of the interface. It also tends to produce poorly conditioned surface triangles due to the constraint of reconstructing the surface elements within a single grid block. On the other hand, this method appears to be quite robust and always reconstructs a topologically valid interface.

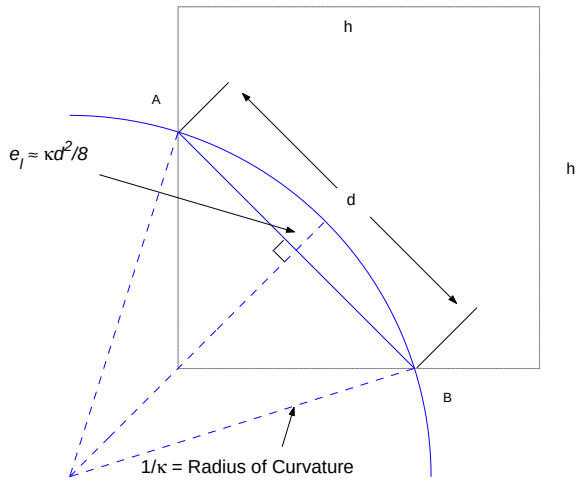


Figure 2.2: *Grid based interface reconstruction introduces a truncation error of the interface position which is of* $O(h^2/\Delta t)$

In Fig. 2.1, we show a comparison of an interface under purely Lagrangian GF propagation with the same problem solved via GB tracking. We start with the same initial interface, an ellipsoid, and propagate with a spiraling velocity field whose angular velocity is described by

$$\omega(r) = \omega_0 + kr \tag{2.1}$$

where $r = \sqrt{x^2 + y^2}$ is the distance to the spiral center. From the comparison we see that the propagation with Eulerian reconstruction introduces a large error in regions where the curvature of the surface is large.

The one step error due to reconstruction of the interface can be analyzed via Fig. 2.2. Let the local error in the front position be e_l, the local mean curvature be κ_l, the grid spacing $\Delta x = \Delta y = \Delta z = h$, and let d represent the distance between two crossings of the interface with the grid edges. The replacement of the interface within the grid block by a linear segment results in an $\mathrm{O}(h^2)$ error in the position of the front:

$$e_l = 1/\kappa_l - \sqrt{(1/\kappa_l)^2 - (d/2)^2} \approx \frac{\kappa_l d^2}{8} \leq \frac{\kappa_l h^2}{4} \tag{2.2}$$

when $\frac{1}{\kappa_l} \gg \frac{d}{2}$. Assuming $\kappa^n_{\max} = \max(\kappa_l)$ is the upper bound of the local mean curvature at time step n, we obtain an L_∞ error bound of the reconstruction at this time step:

$$E^n_\infty = \frac{\kappa^n_{\max} h^2}{4} \tag{2.3}$$

Let $P^n(x)$ be the position of the interface at time step n, so that $P^0(x)$ is the position of the initial interface. Let Δt being the time step. From equation above, we have

$$\left\| P^{n+1}(x) - P^n(x) \right\|_\infty \leq E^n_\infty \quad . \tag{2.4}$$

Therefore,

$$\left\| P^{n+1}(x) - P^0(x) \right\|_\infty \leq \sum_{i=1}^{n} E^i_\infty = \frac{h^2 \sum_{i=1}^{n} \kappa^i_{\max}}{4} = O\left(\frac{h^2}{\Delta t}\right). \tag{2.5}$$

This error is inversely proportional to Δt, which means that for a given grid size, reducing the time step will increase the truncation error. Assuming a CFL determined time step, the method is first order accurate.

The error can be assessed quantitatively using a velocity map for a certain time interval followed by its inverse (the negative of v in place of v). The analytic solution is then the initial conditions (taken to be a sphere) and errors computed by an L_1 norm are easy to determine. In Table 2.1, we compare grid based propagation errors to locally grid based propagation errors (with fourth order propagation), and a velocity

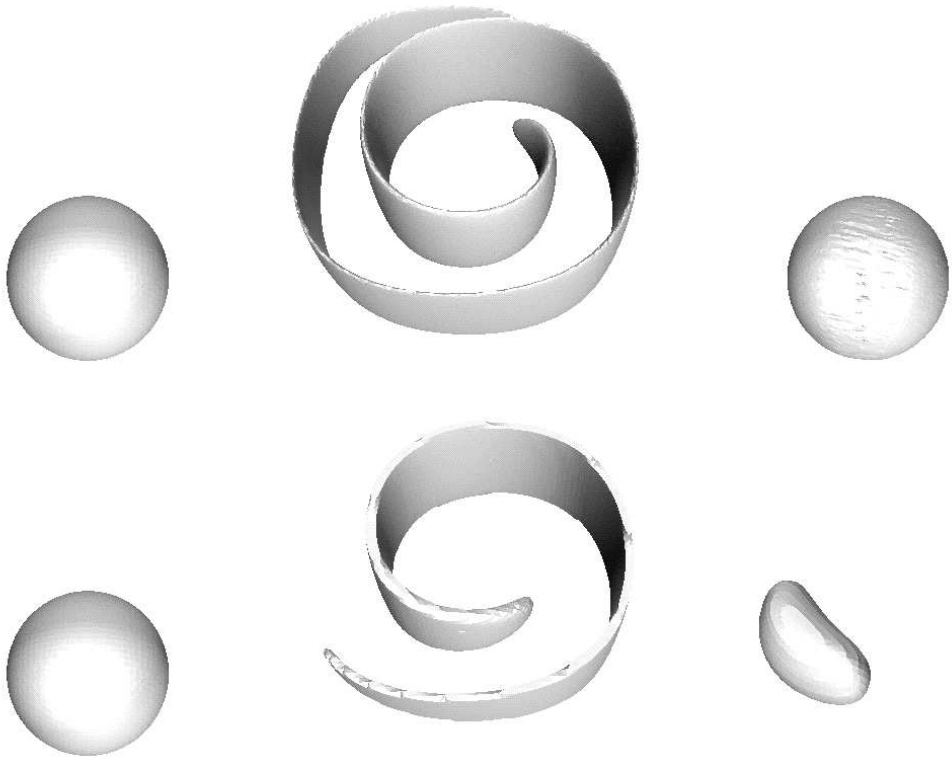


Figure 2.3: *Comparison of LGB and GB in a single vortex velocity field with reversal. using a common* 100^3 *mesh. The initial interface is sphere of radius 0.15 centered at* $(0.5, 0.75, 0.5)$. *The upper row shows the locally grid based tracking and the lower row shows the grid based tracking. From left to right are* $t = 0, 4, 8$ *respectively.*

field given by a single vortex:

$$\begin{cases} u(x,y,z) &= -\sin(2\pi y)\sin^2(\pi x)\cos\left(\frac{\pi t}{T}\right) \\ v(x,y,z) &= \sin(2\pi x)\sin^2(\pi y)\cos\left(\frac{\pi t}{T}\right) \\ w(x,y,z) &= 0.0. \end{cases} \tag{2.6}$$

The initial interface is a sphere of radius 0.15 centered at (0.5,0.75,0.5) and the computational domain is a $1 \times 1 \times 1$ cube. The result of the comparison is shown in Table 2.1. We observe that the local grid based error is better than one order of magnitude smaller than the grid based error.

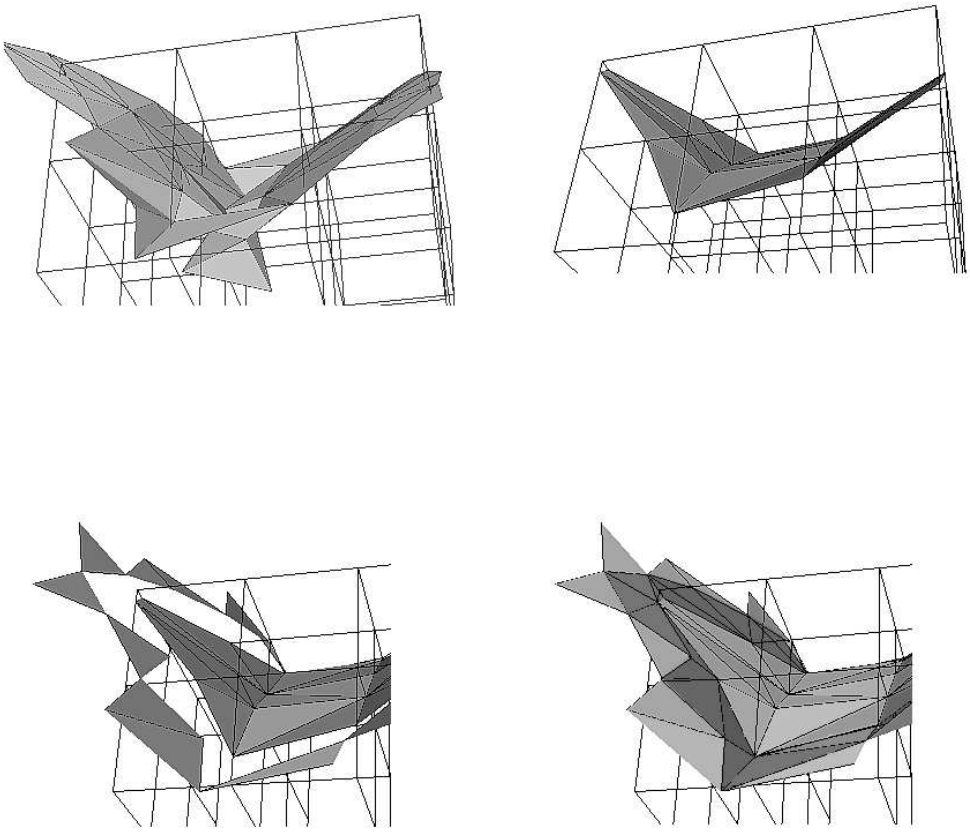


Figure 2.4: *Steps to reconstruct a tangled section of the three dimensional interface. From left to right and top to bottom: (1) assemble blocks which contain un-physical edges, (2) delete triangles attached to the box and rebuild the interface through the grid-based method, using the grid-based method to reconstruct the interface topology inside the box, (3) align triangles inside the box and outside the box, (4) relink the interface topology for triangles inside and outside the box, and thereby obtain the final interface with new topology.*

Case	Grid Based	Locally Grid Based
$T = 4$	5.70×10^{-3}	2.47×10^{-4}
$T = 8$	1.22×10^{-2}	3.37×10^{-4}

Table 2.1: Comparison of the L_1 error for the 3D Front Tracking method with grid based and locally grid based algorithms. The simulation is in a single vortex velocity field Eq. (2.6) followed by a reversal to the initial conditions. The simulations are performed in a $1 \times 1 \times 1$ domain with a 100^3 computational mesh. The initial interface is sphere of radius 0.15 centered at $(0.5, 0.75, 0.5)$.

2.2 Locally Grid Based Method

To reduce the GB interface interpolation error, we introduce a new method, LGB, or the locally grid based tracking, which combines the advantages of both methods. We use the fully Lagrangian GF method to propagate the interface to obtain an accurate solution of the interface position. Eulerian GB reconstruction of the interface is only used in small regions where topological bifurcation is detected. The detection of topological changes is through a fast algorithm which walks through the Eulerian grid to check the consistency of the indices assigned to grid nodes of each subdomain and the corresponding side of the interface. In the first step of the procedure, the intersections between the interface and cell edges of the Eulerian grid are inserted and the index of every subdomain is assigned to each corner point of the Eulerian grid. We then check the consistency between the indices of each crossing point and the node point it faces. If inconsistency is detected, the corresponding mesh block is recorded. This intersection detection algorithm is approximate in that it will miss bifurcations totally internal to a single mesh cell.

The construction advances through four steps.

1. Those recorded blocks will be assembled to form boxes and overlapping boxes will be merged.

2. Surgery is performed within each box. Triangles crossing the box boundaries are recorded for later use. All the triangles inside the box or attached to the box will be deleted leaving only the crossing points of the interface and grid edges. A grid-based reconstruction is followed to build the new section of the interface inside the box.

3. The triangles totally outside the box form the exterior interface.

4. The region between the exterior and interior interfaces is re-triangulated to join the two smoothly.

Fig. 2.4 shows the procedural steps of the local reconstruction of the interface. The reconnection step 4 is the most crucial step in the LGB method. We modify the triangles recorded in step 2 by a series of steps to make the triangles in the reconnection region also grid based relative to the box boundary. These substeps of step 4 are summarized as follows:

4.1 We split triangles crossing the box boundary which are recorded in step 2 using the intersection points between triangles and box boundaries. There are two types of intersection points: one is the intersection points between triangles and grid cell edges on the box boundary (TYPE I), the other is the intersection points between the sides of triangles and the box boundaries (TYPE II). We first divide each triangle by recursively inserting the TYPE I intersection points (if any) inside the triangle. Each triangle will be divided into three smaller triangles by joining the TYPE I intersection point with the vertices of the triangle. Then we insert the TYPE II intersection points recursively. Each triangle is divided into two smaller triangles by joining the TYPE II intersection point and the vertex opposite to the side containing this intersection point. After this step,

the triangles which crossed the box boundaries are split into triangles lying either entirely inside the box or entirely outside the box. We keep only the triangles entirely outside the box. These triangles meet the box surface along a closed curve which is actually a piece-wised linear curve connecting all the intersection points.

4.2 The curve gives an ordering to these triangles, and in this ordering, we merge triangles whose vertices are TYPE II intersection points until there is no TYPE II intersection points in the curve. After this step it is sufficient to assume that each triangle meets the box boundary only as a line joining adjacent grid edge. i.e. the triangle meets the box in a grid based manner.

After this operation, all triangles will meet the box edge only as a line joining adjacent grid edges. That is, all triangles, inside and outside, meet the box only in a grid based manner. The reconnection between the outside triangles and the newly reconstructed triangles inside the box is then a simple match of the triangle sides at the box boundaries.

This method reduces the use of the Eulerian reconstruction to a minimum. It is particularly useful for the computation of interface motion in which the interface has regions of large curvature. It reduces interface interpolation errors and minimizes the unphysical disappearance of the fragmented components of the material after bifurcation.

Chapter 3

Numerical Implementation and Comparison Study

Systematic comparisons of algorithms for the numerical modelling of interfaces were performed by Rider and Kothe [101, 102]. Tests show superior performance for particle methods, followed by PLIC-VOF, level sets and capturing, in that order. The purpose of this chapter is to extend this comparison to front tracking, to present new (3D) front tracking algorithms and to compare the new to previous front tracking algorithms. In this extended comparison, front tracking is comparable or superior to particle methods, and superior to the others.

3.1 Comparison with the Level Set Method

Comparison of the Lagrangian front tracking method and the Eulerian level set method has focused on some important geometrical properties of the interface. The truncation error of the level set method has following sources (1) interpolation of the level set function, (2) re-initialization of the level set function, and (3) extrapolation of the velocity field when the velocity is a function of the geometry (normal direction, curvature) only. Such errors are amplified in the neighborhood of a highly curved segment of the front. The resolution of the level set method is dependent on the Eulerian mesh size and the order of the PDE solver for the level set equation. We

have conducted two tests used by Fedkiw et al. The level set code comes from

`http://www.cs.ubc.ca/ mitchell/ToolboxLS`

3.1.1 Rotating Slotted Disk

A common test for the quality of front propagation is the rigid rotation of a slotted disk. Two sets of comparison are made between the FronTier code and the level set ToolBox package. The first comparison uses the first order Euler forward scheme for the point propagation in the front tracking method and the first order scheme for the solution of the level set functions. The radius of the disk is $R = 0.5$, the width and depth of the slot in the disk are $W = 0.1$ and $H = 0.4$ respectively. A CFL condition is applied to both runs. After one circulation, the slot in the level set solution is completely flattened and the disk has shrunk by about 15 percent. In the front tracking computation, the radius of the radius of the disk has expanded by 5 percent, while its slot is still well maintained.

In the second comparison, we used the fifth order WENO scheme for the convection of the level set function, while for the front tracking code, we used the fourth order Runge-Kutta method for the point propagation. The difference after the first round of circulation was not evident. So we continued the computation for 13 revolutions. The fourth order Runge-Kutta method appears to be extremely accurate in the front tracking simulation. Even after 13 rounds of rotation, the change of both radius of the disk and the slot are invisible. The level set computation began to show edge smoothing after the second rotation. At the end of the 13th circulation, the slot is closed at the top resulting in a topologically incorrect bifurcation.

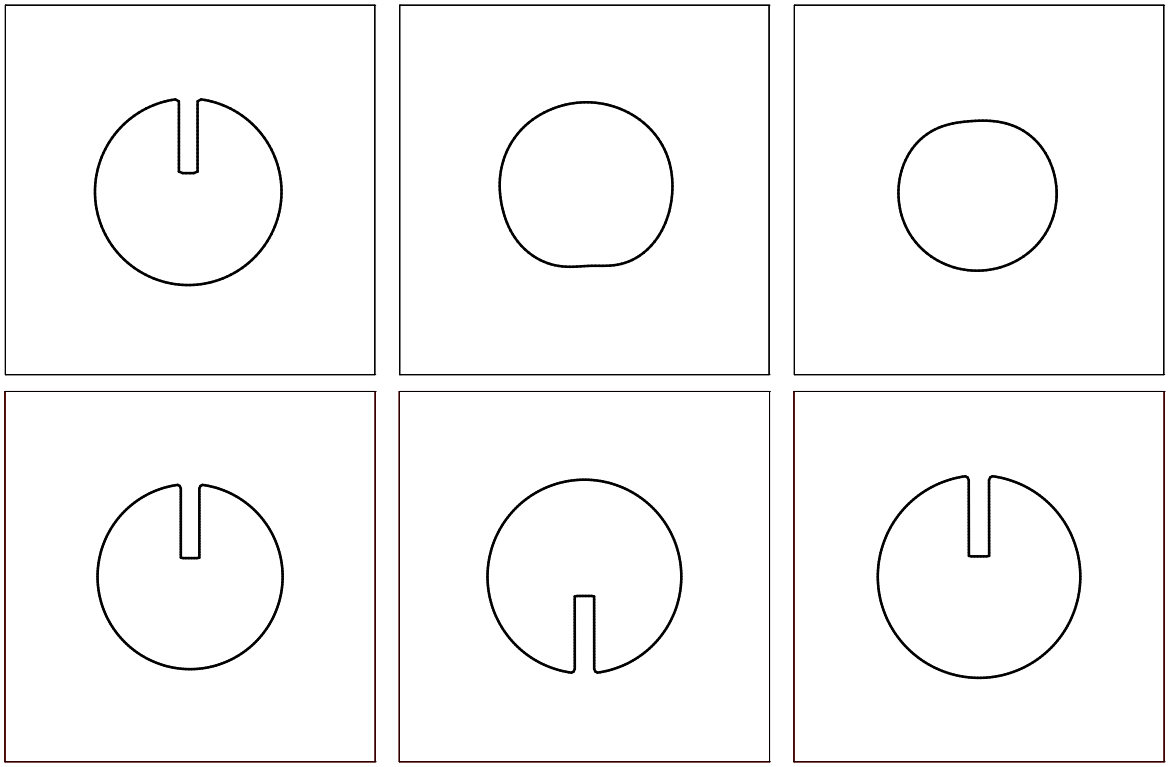


Figure 3.1: Comparison of slotted disk simulation using first order methods. The upper sequence shows the result of the level set method while the lower sequence shows the result for the front tracking method. The grid for both is 100×100.

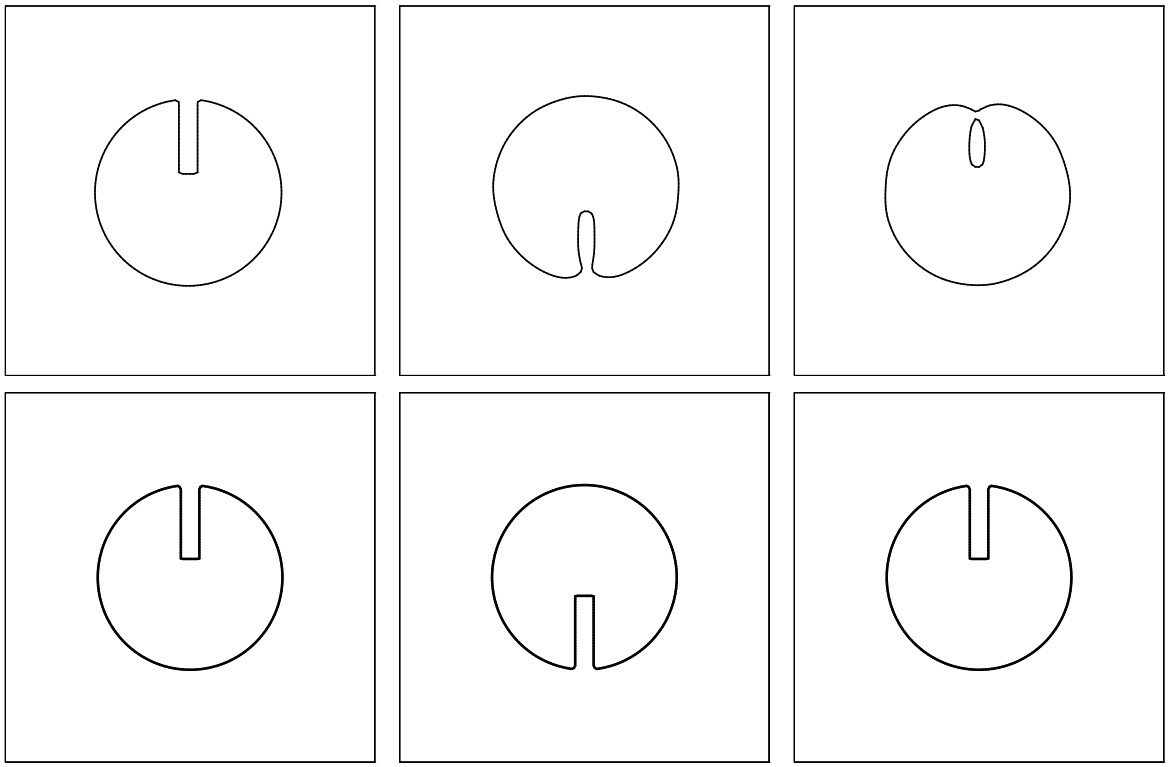


Figure 3.2: Comparison of slotted disk simulation using high order methods. The upper sequence shows the result of the level set method using the fifth order WENO scheme and the lower sequence shows the result of front tracking using the fourth order Runge-Kutta method.

3.1.2 Swirling Ellipsoid and Reversal

Another set of tests comparing the resolution between level set method and the front tracking method is to place an ellipsoid (ellipse in 2D) in a swirling velocity field and then reverse the velocity direction when it reaches a preset time T. In a 2D example, we place a ellipse in a dipole velocity field:

$$\mathbf{v} = \frac{\mathbf{k} \times (\mathbf{r} - \mathbf{r_1})}{|\mathbf{r} - \mathbf{r_1}|^2} w_1 - \frac{\mathbf{k} \times (\mathbf{r} - \mathbf{r_2})}{|\mathbf{r} - \mathbf{r_2}|^2} w_2 \tag{3.1}$$

where w_1 and w_2 are the strength of the vortices, and $\mathbf{r}$, $\mathbf{r_1}$ and $\mathbf{r_2}$ are the position vectors of the point, the center of the first vortex and the center of the second vortex. In the example shown in Fig. 3.3, we placed the center of the ellipse at the origin, while the two radii of the ellipse are $a = 0.4$ and $b = 0.2$ respectively. The center of the two vortices are at $(-0.3, 0)$ and $(0.3, 0)$ and the strengths of the two vortices are $w_1 = w_2 = 0.1$.

If we set the reversal time at $T = 10$, the swirling vortex tails become so thin that the level set method makes an incorrect topological bifurcation. On the reversal, this bifurcation cannot be recovered. The front tracking solution also looses some resolution, but it maintains the correct topology.

3.2 PLIC-VOF Comparisons

To compare with the multiple interface methods studied in [81, 101, 102], we have simulated several test problems. These tests were done with exactly the same initial conditions and velocity fields as used by Rider and Kothe. These tests include

(1) the single vortex velocity field in two dimensions,

(2) the deformation velocity field in two dimensions,

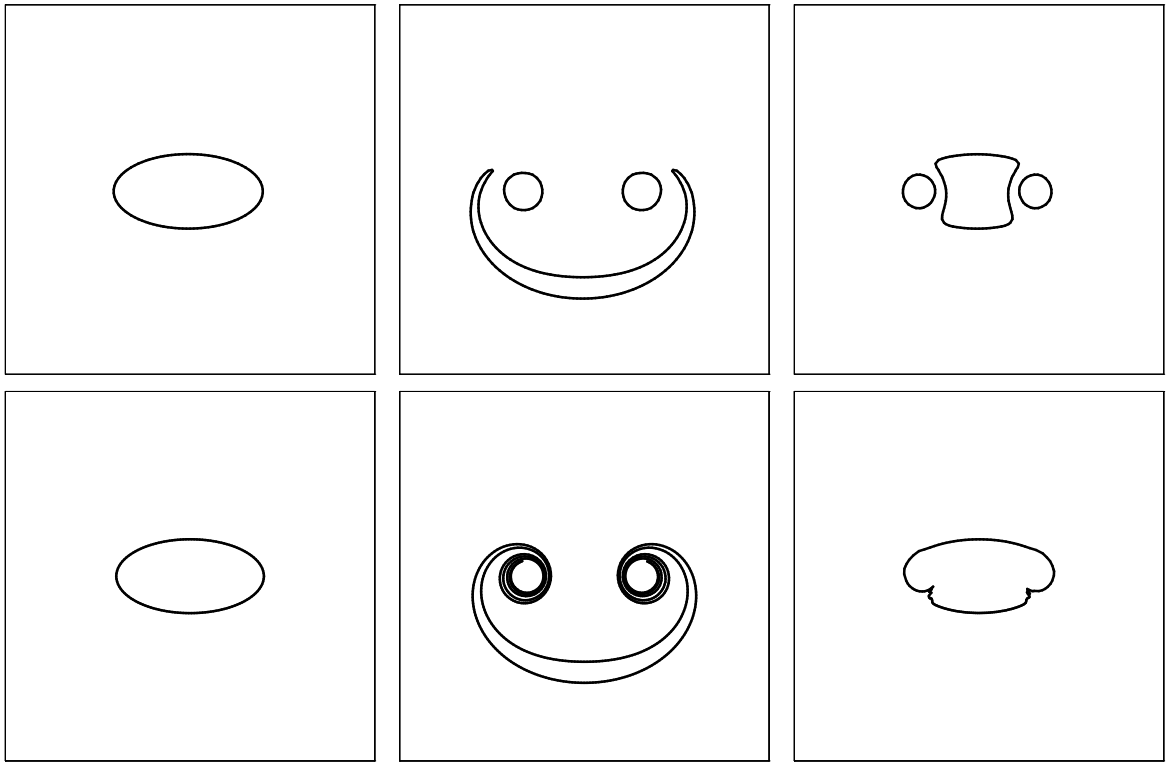


Figure 3.3: Comparison between level set and front tracking on velocity reversal. The upper sequence shows the results of the level set method using a fifth order WENO integration while the lower sequence shows the results of front tracking using the fourth order Runge-Kutta method for the point propagation. The ellipse is placed in a dipole velocity field. The velocity is reversed at t = 10.

(3) the deformation velocity field in three dimensions,

(4) the vortex shearing flow velocity field in three dimensions.

The computational domains and initial conditions for (1) and (2) are identical. It is a circle of radius 0.15 centered at (0.50, 0.75) within a 1×1 computational domain. For each test, we plotted the interface and calculated the L_1 norm to compare with the counterparts in [101, 102]. We also performed convergence tests of (1) in comparison with that in [101].

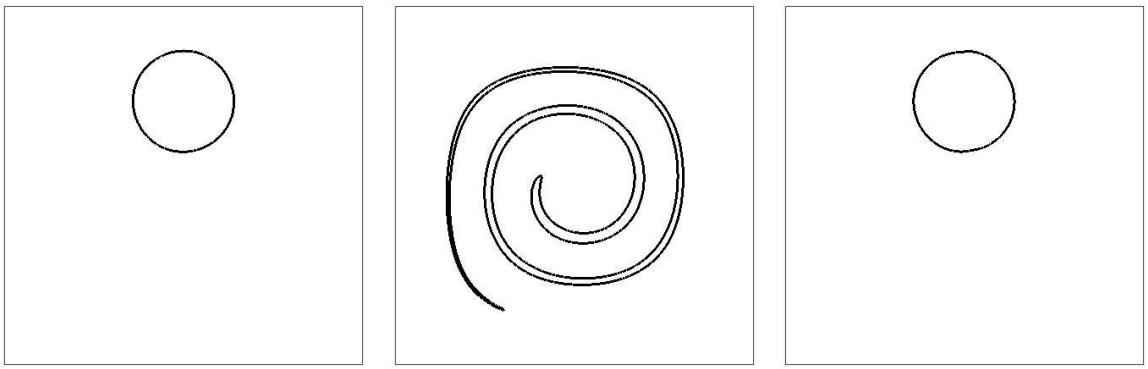
3.2.1 Two Dimensional Single Vortex Simulation

The single vortex problem has the velocity field described by the stream function [109]

$$\Psi = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y). \tag{3.2}$$

To compare the results in [101, 102], we let the simulation continue to $t = 3$ and then with a reversed velocity field to $t = 6$. A CFL number of 1.0 is used. Figure 3.4 shows the interface evolution. The left plot shows the comparison of the initial circle and the interface after reversal, and the right plot shows the interface at $t = 3$. Topologically, the resolution of the vortex tail of the interface matches the best plot of that in [101, 102].

We tested the convergence of the L_1 norm of the error. In L_1 norm tests, we multiplied Eq. (3.2) by a factor of $\cos(\pi t/T)$ so that the interface begins to reverse at $t = T/2$ and is fully reversed at $t = T$. The convergence tests are done for $T = 2,\ 4,\ 6$ respectively and are summarized by Table 3.1. Although the point propagation used the 4th order Runge-Kutta method, the order of convergence is only approximately equal to 2. The reason for the reduced order is due to the redistribution of the front points which is linear.



$t = 0$ $t = 3$ $t = 6$

Figure 3.4: *Reversal test of a 2D interface in the single vortex velocity field. The computation is performed in a* 128^2 *computational mesh. In comparison with Figure 4 of [101], the resolution of the interface matches the best results by the Marker Particle methods.*

Case	32^2	64^2	128^2
$T = 2$	1.03×10^{-3}	2.39×10^{-4}	9.54×10^{-5}
$T = 4$	2.12×10^{-3}	5.22×10^{-4}	9.35×10^{-5}
$T = 6$	2.79×10^{-3}	9.42×10^{-4}	1.19×10^{-4}

Table 3.1: Convergence test of the 2D front tracking method under the single vortex velocity field with $CFL = 1.0$.

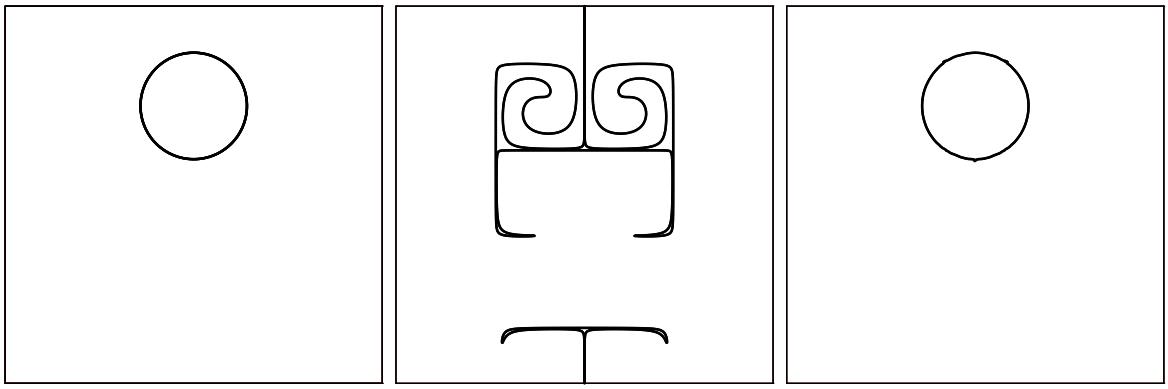


Figure 3.5: *Reversal test of a 2D interface in the 2D deformation velocity field. The figure is visually superior to all comparison solutions, including the particle methods in [101, 102].*

3.2.2 Two Dimensional Deformation Test

The stream function for the two dimensional deformation velocity field is given by

$$\Psi = \frac{1}{4\pi} \sin\left(4\pi\left(x + \frac{1}{2}\right)\right) \cos\left(4\pi\left(y + \frac{1}{2}\right)\right) \cos\left(\frac{\pi t}{T}\right). \tag{3.3}$$

In this experiment, we have $T = 2$. For the purpose of comparison, the interface plot of the FronTier advection to $t = 1$ is on a 128^2 grid which is shown in Fig. 3.5. The plot is visually superior to all comparison solutions of [101, 102], including the particle methods. We also performed a 64^2 reversal simulation to $t = 2$ to compare the L_1 norm with that in [101]. The L_1 norm the of front tracking simulation is $L_1 = 2.25 \times 10^{-3}$. This result is better than all methods in Table 4 of [101] except the Marker Particle (4) and Marker Particle (16) methods.

Mesh	LGB	Order	CVTNA	Youngs
32^3	5.72×10^{-3}	3.72	7.41×10^{-3}	7.71×10^{-3}
64^3	4.33×10^{-4}	1.82	1.99×10^{-3}	2.78×10^{-3}
128^3	1.23×10^{-4}	N/A	3.09×10^{-4}	7.58×10^{-4}

Table 3.2: L_1 norms at $t = 3$ for the LGB method in the three dimensional deformation simulation compared to the two interface methods used in [81] with $CFL = 0.5$.

3.2.3 Three Dimensional Deformation Field Test

The velocity field in this experiment is described by the equations

$$u(x,y,z) = 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos(\pi t/T) \tag{3.4}$$

$$v(x,y,z) = -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\cos(\pi t/T) \tag{3.5}$$

$$w(x,y,z) = -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)\cos(\pi t/T) \quad . \tag{3.6}$$

The interface evolves dynamically from an initial sphere of radius 0.15 centered at (0.35, 0.35, 0.35) to $t = 1.5$. The velocity field will then reverse its direction. At $t = 3.0$, the interface comes back to its initial state. The error comparison with the two PLIC methods in [81] is given in Table 3.2, and shows superior performance for LGB Front Tracking.

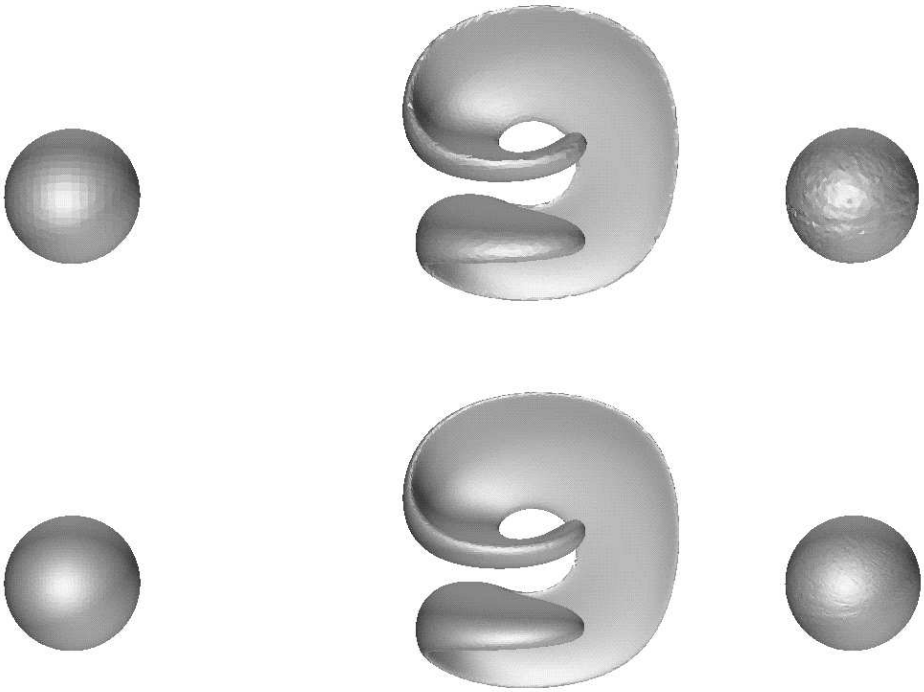


Figure 3.6: *Reversal test of a 3D interface in deformation velocity field with* $CFL = 0.5$. *The sequence above has the mesh of* 64^3, *and the sequence below has the mesh of* 128^3. *From left to right are* $t = 0,\ 1.5,\ 3$ *respectively.*

Method	$T = 3$	$T = 6$	$T = 9$
LGB	6.28×10^{-4}	9.07×10^{-4}	1.19×10^{-3}
CVTNA	6.20×10^{-4}	3.64×10^{-3}	8.01×10^{-3}
Youngs	9.99×10^{-4}	4.38×10^{-3}	8.59×10^{-3}

Table 3.3: L_1 norms for $T = 3$, $T = 6$, and $T = 9$ for the comparison of the LGB method with the two other methods in [81] in the three dimensional deformation simulation in the shear flow velocity field. The computational mesh is $64 \times 64 \times 128$ and $CFL = 1.0$ is used.

3.2.4 Three Dimensional Shear Flow

The shear flow velocity field is described by

$$u = \sin(2\pi y)\sin^2(\pi x)\cos\left(\frac{\pi t}{T}\right) \tag{3.7}$$

$$v = -\sin(2\pi x)\sin^2(\pi y)\cos\left(\frac{\pi t}{T}\right) \tag{3.8}$$

$$w = \left(1 - \frac{r}{R}\right)^2 \cos\left(\frac{\pi t}{T}\right), \tag{3.9}$$

where $r = \sqrt{(x - x_0)^2 - (y - y_0)^2}$, $R = 0.5$, $x_0 = y_0 = 0.5$.

The initial sphere is centered at $(0.5, 0.75, 0.25)$ with radius 0.15. In Figure 3.7 we plotted the interfaces of two runs using the LGB method at $t = T/2$ and $t = T$. In comparison with [81], The LGB method is visually superior. It maintains a singly connected topology throughout the simulation while the simulations in [81] using two PLIC methods showed bifurcation of the interface. Table 3.3 gives the comparison of the L_1 norms, and again shows the advantage of the LGB method.

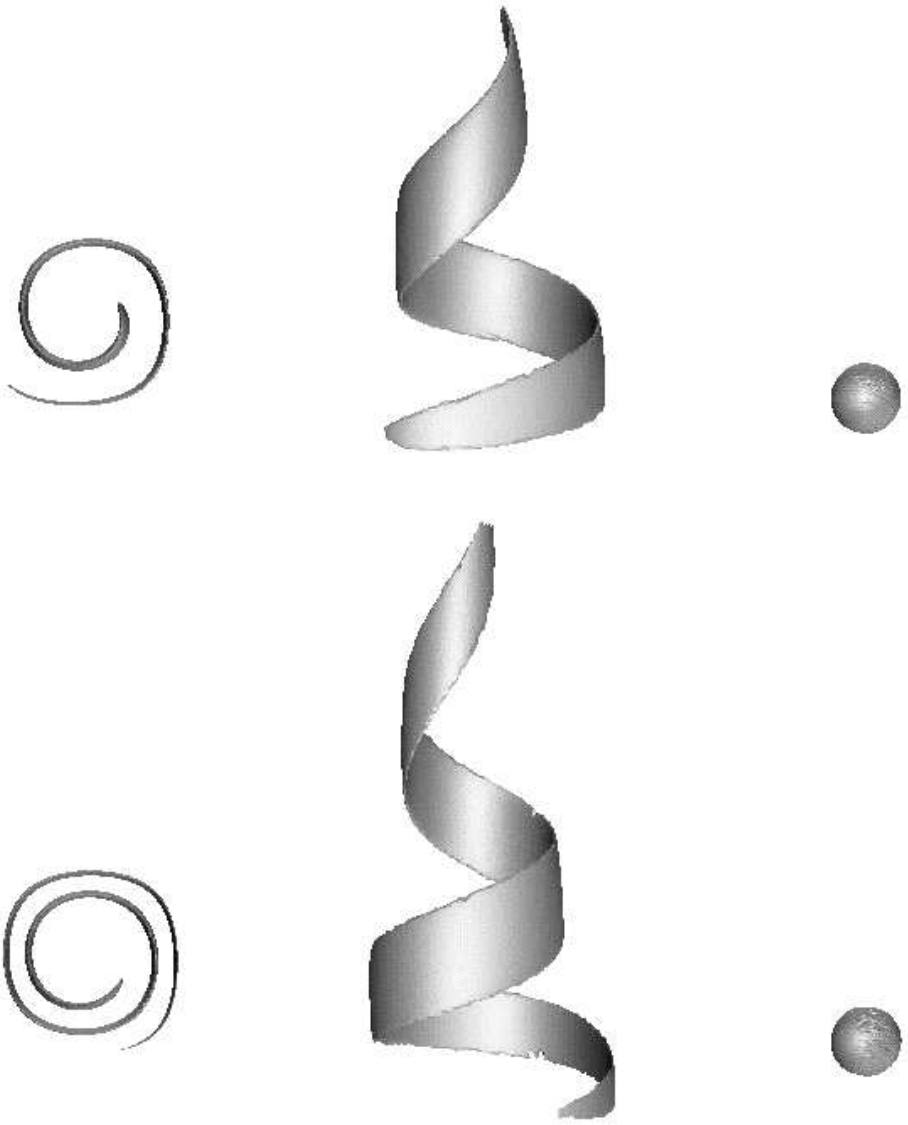


Figure 3.7: Profiles at maximum deformation for the 3D shearing flow using the locally grid-based front tracking. The mesh is $64 \times 64 \times 128$ and $CFL = 1.0$. The first row is for $T = 6$ and the second row is for $T = 9$. The first column is the xy-plane view and the second column is the yz-plane view of the interface at $T/2$. The third column gives the interfaces after complete reversal $t = T$.

Chapter 4

Applications to the Numerical Study of Physical Problems

4.1 Numerical Study of Mean Curvature Flow

Mean curvature flow evolves hypersurfaces in their normal direction with speed equal to the mean curvature at each point. Front Tracking method has achieved great success in study of mean curvature dependent motion due to the accurate tracking for the interface position [12, 14, 38, 39]. However the complexity in handling topology changes has made Front Tracking less applicable to the mean curvature problems in more than two dimensions. In this chapter we will explain the difficulties and illustrate how we can conquer these difficulties with the enhanced Front tracking method.

4.1.1 Introduction

In many physical problems (flame propagation, crystal growth, minimal surface), interfaces move with speed that depends on the local curvature.

Analytically, the problem has been studied widely by many authors from different points of view. In 1978, Brakke studied mean curvature flow in the framework of singular surfaces, so-called integral varifolds [11]. In the early eighties, Gage and

Hamilton showed that convex curves evolve smoothly, becoming asymptotically round as they disappear. Grayson extended the smoothness to general embedded curves in [57, 59]. Angenent has shown how to evolve immersed curves in [2–4]. Huisken [71, 72]and Sinestrari [73] and White [117] studied asymptotic behavior of singularities in the mean convex case. Comprehensive estimates of graphs evolving by mean curvature appear in Ecker-Huisken [33, 34]. Alklen and Cahn conjectured mean curvature motion as the singular limit of a reaction-diffusion (phase-field) equation [1]. This idea has been developed by de Mottoni and Schatzman [24, 25], Bronsard-Kohn [13], and many others. Following Osher-Sethian's numerical work [94], Chen-Giga-Goto [19] and Evans-Spruck [36] introduced the level-set approach. Ilmanen [75] further established the link between level-set flow and Brakke's varifold solution framework. For a list of bibliography, we refer to the book [32].

Explicit solutions seldom exist. Several numerical methods have been developed. Most of these can be divided into two groups: direct, or front tracking methods, where the motion of the interface is explicitly considered; and indirect methods, where the interface is given implicitly as the level set of some function. A survey of numerical formulations can be found in [7].

Several direct or front tracking methods have been proposed. In [14, 39], direct discretization of the evolution equation is used for each interface. They are quite efficient for curves that never cross. When line or planar segments interact, however, decisions must be made as to whether to inset or delete segments. Brakke's surface evolver minimizes surface energy to produce a variety of motions including motion by mean curvature. For three dimensional problems, however, it requires user intervention when topological changes occur [12].

Other direct methods include: Phase field methods [15], Front tracking coupled with the boundary integral method used for study of surface diffusion flow [88], Front

tracking coupled with finite elements [31] and the Allen-Cahn formulation [95].

The Hamilton-Jacobi level set method of Osher and Sethian [94] naturally handles topological merging and breaking in any number of spatial dimensions. Unfortunately, this method can be excessively slow in three dimensions [103] and also suffers from excessive dispersion. Variations of the level set approach include the MBO-method [91], and the Spectrum-Method based on the MBO-method [103].

Two major issues have made Front Tracking less applicable in the study of mean curvature flows in 3D:

First, handling of topological changes. Complicated topological changes often occur in 3D mean curvature flow. Many front tracking algorithms break down due to the lack of a robust algorithm for handling topological changes.

Second, accurate and stable estimation of the surface normal and curvatures at the vertices of the tracking interface. Direct discrete approximation to the interface geometries often suffers from an instability when the points on the interface are very close to each other. Thus, either an extremely small time step is required or some re-gridding mechanism is employed which might change the physics of the problem [94].

In the following sections we discuss how Locally Grid Based Front tracking can achieve accurate and stable propagation of interface with mean curvature dependent velocity. This chapter would be organized as following: we introduce some basic notations from differential geometry in Section 4.1.2 and then introduce our formulation in Section 4.1.3. We will discuss our algorithm to estimate the vertex normal and curvature in Section 4.1.4 and present the numerical results in Section 4.1.6.

4.1.2 Notations from Differential Geometry

Definition 4.1.1 *A hypersurface M of class C^2 is a subset of R^n which is locally the graph of a function u of class C^2*

$$M = \{(x', x_n) = (x_1, x_2, \ldots, x_{n-1}, x_n) \in R^n | x_n = u(x')\}$$

In 3D, we have the following definitions: A *normal curve* is the intersection of a surface with a plane containing the normal n. For a given direction d in the tangent plane there is a unique normal curve, obtained by intersecting the plane spanned by n and d with the plane. The curvature of a normal curve is called *sectional curvature.* The *principal curvature* (κ_1, κ_2) are the maximal and minimal sectional curvatures. *The principal curvature directions* are the directions in the tangent plane for which the maximum and minimum are attained. these directions are perpendicular to each other. The sectional curvature corresponding to the direction in the tangent plane forming an angle α with the first principal curvature direction is given by

$$\kappa = \kappa_1 cos^2\alpha + \kappa_2 sin^2\alpha$$

Gaussian curvature is also called intrinsic curvature, and is defined as

$$\kappa_G = \kappa_1\kappa_2$$

If the surface is isometrically deformed (i.e. the distances between points on the surface along the surface do not change) the Gaussian curvature is preserved.

Mean Curvature is defined by

$$\kappa_H = \frac{\kappa_1 + \kappa_2}{2}$$

Given a parameterized surface $r(u,v)$: $R^2 \to R^3$, let $r_u = \frac{\partial r}{\partial u}, r_v = \frac{\partial r}{\partial v}$, and the unit normal $n = \frac{r_u \times r_v}{|r_u \times r_v|}$. Also define

$$\begin{aligned} E = (r_u \cdot r_u), F = (r_u \cdot r_v), G = (r_v \cdot r_v) \\ L = (n \cdot r_{uu}, M = (n \cdot r_{uv}), N = (n \cdot r_{vv}) \end{aligned} \tag{4.1}$$

Then the formula for the Gaussian curvature is

$$\kappa_G = \frac{L \cdot N - M^2}{E \cdot G - F^2}$$

and for mean curvature it is

$$\kappa_H = \frac{L \cdot G - 2F \cdot M + E \cdot N}{2(E \cdot G - F^2)}$$

4.1.3 Basic equations

The formulation of the mean curvature flow problem for hypersurfaces is taken from Huisken's work [71].

Definition 4.1.2 *A family $(M(t))_{t\geq 0}$ of hypersurfaces evolves, from a hyper surface M_0, according to the mean curvature flow if*

$$\begin{aligned} \frac{\partial F}{\partial t}(s,t) &= H(F(s,t))N(F(s,t)) \forall s \in M, \forall t > 0, \\ F(s,0) &= F_0(s), \end{aligned} \tag{4.2}$$

where H denotes the mean curvature of $M(t)$, N the innernormal to $M(t)$ and F_0 defines M_0 equal to $M(\cdot, 0)$.

For the special case of a uniformly convex initial surface, Huisken has proved the following existence result.

Theorem 4.1.3 *Suppose that M_0 is uniformly convex (that is the principal curvatures are positive everywhere). Then 4.1.2 has a smooth solution on $[0, T[$. Moreover, the hypersurfaces $M(\cdot, t)$ converge spherelike to a point when t increases to T.*

For general cases, analytic solutions are hard to find and topological changes are possible.

We discretize the Equation using the Euler Predict-Correct (EPC, Heun's) scheme, which is 2nd order accurate and unconditionally stable.

$$\begin{aligned} \tilde{y}_{n+1} &= y_n + \Delta t \cdot H(t_n, y_n) \\ y_{n+1} &= y_n + \frac{1}{2}\Delta t \cdot (H(t_n, y_n) + H(t_{n+1}, \tilde{y}_{n+1})) \end{aligned} \tag{4.3}$$

The numerical method to compute the normal and curvature is critical to get a accuracy and stable propagation. A numerical study of normal and curvature estimation in *FronTier* was primarily done by my colleague Dr. Jian Du. We would discuss these results in details in next section.

4.1.4 Calculation of Normal and Curvature

Many methods have been proposed to estimate normals and curvatures on the vertices of discrete mesh. These methods can be grouped as either discrete method or fitting methods. Discrete methods usually estimate vertex normals (curvatures) using a weighted average of face normals (curvatures), such as area weighting, angle weighting and spherical weighting [26, 87, 113]. Fitting methods usually estimate normals (curvatures) by fitting the one-ring or two-ring neighbor of a vertex quadratically or

cubically [16, 56, 90]. A comparison study of different vertex normal computation can be found in [77] by Jin et al. A number of methods for curvatures were discussed in Gatzke and Grimm's paper [40]. For list of bibliographies we refer to [76] by Jiao et al.

Estimation of Vertex Normal

After careful study, we chose to use Max' Sine weighted normal algorithm [87]. It is easy to implement, stable, convergent and accurate.

By Max' Sine weighted formula, the vertex normal is expressed as a weighted sum of the normals to the facets surrounding the vertex. The key point is to assign large weights for smaller facets; it is especially suitable when the facets surrounding the vertex differ greatly in size. Both this new algorithm and our previous one use only points adjacent to the point at which the normal is evaluated, and so both have a stencil radius of Δx. Our curvature algorithm also has approximately this same stencil.

Suppose O is a vertex of the polyhedron which is located at the origin and has adjacent vertices $\mathbf{V_0}, \mathbf{V_1}, \ldots, \mathbf{V_{n-1}}$. The normal of the triangle $\mathbf{V_i} O \mathbf{V_{i+1}}$ can be calculated as

$$\mathbf{N_i} = \mathbf{V_i} \times \frac{\mathbf{V_{i+1}}}{|\mathbf{V_i}||\mathbf{V_{i+1}}| \sin \alpha_i} \tag{4.4}$$

where α_i is the angle between $\mathbf{V_i}$ and $\mathbf{V_{i+1}}$. The Max weight of the $\mathbf{N_i}$ in the calculation of $\mathbf{N}$ is $\sin \alpha_i / (|\mathbf{V_i}||\mathbf{V_{i+1}}|)$. That is

$$\mathbf{N}(O) = \sum_{i=1}^{K} \frac{\mathbf{N_i} \sin \alpha_i}{|\mathbf{V_i}||\mathbf{V_{i+1}}|} = \sum_{i=1}^{K} \mathbf{V_i} \times \frac{\mathbf{V_{i+1}}}{|\mathbf{V_i}|^2 |\mathbf{V_{i+1}}|^2} \tag{4.5}$$

This new algorithm for calculating the interface normal has been compared with

the old area weighted algorithm

$$\mathbf{N}(O) = \frac{1}{A}\sum_{i=1}^{K} \mathbf{N_i} A_i \tag{4.6}$$

where A_i is the area of the i-th surrounding triangle and A is the summation of A_i's. Table 4.1 shows the results of the comparison. We note that comparison results are sensitive to surface regularity, and that the Max algorithm is specifically superior for regular (nearly spherical) surfaces.

Estimation of Vertex Curvature

We estimate the vertex curvature by fitting a local quadratic surface. First, we reconstruct a local coordinate system. Let the vertex O be the origin of the new coordinate system and let the z direction be along the normal direction of the vertex at O. We approximate the surface as a quadratic surface

$$z = Ax^2 + Bxy + Cy^2 + Dx + Fy \tag{4.7}$$

To determine the coefficients, at least five neighboring vertices must be included. If the number of neighboring points is less than five, we add the closest next to the adjacent points. If the number of neighboring points is greater than five, least square fitting is used to determine the coefficients A, B, C, D, E. The curvature of the quadratic surface is used as the approximation for the surface curvature at O. The improved definition for the normal yields an improved curvature. See Table 4.1, from which we infer a second order rate of convergence for κ, whereas the previous method appears to be $\mathcal{O}(1)$, *i.e.,* bounded but not convergent.

Sphere Radius = 0.1			
Method	Mesh size	$\| \kappa - \kappa_e \|_1 / \| \kappa_e \|$	$\| \overrightarrow{n} - \overrightarrow{n_e} \|_1$
Max with Quadratic Fitting	40x40x40	0.05700	1.1e-5
	80x80x80	0.01440	2.8e-5
Area Weighted Integral Method	40x40x40	0.10885	0.0863
	80x80x80	0.13996	0.0506
Sphere Radius = 0.3			
Method	Mesh size	$\| \kappa - \kappa_e \|_1 / \| \kappa_e \|$	$\| \overrightarrow{n} - \overrightarrow{n_e} \|_1$
Max with Quadratic Fitting	40x40x40	0.0061	3.6e-5
	80x80x80	0.0015	2.3e-5
Area Weighted Integral Method	40x40x40	0.1149	0.0323
	80x80x80	0.1487	0.0154

Table 4.1: Comparison of mean curvature and normal between the Nelson Max method and the area weighted method.

4.1.5 Interface Optimization

We have observed from our study that the mesh quality plays a very important role in constructing an accurate estimation of the vertex normal and curvature. Thus we employed the Grid Free retriangulation procedure introduced in 2.1.1 to optimize the interface.

4.1.6 Numerical Results

Shrinking Sphere Test

We first test the accuracy and convergence of our algorithm by simulating a shrinking sphere for which the analytical solution is known. A sphere of radius 0.3 centered at $(0.5, 0.5, 0.5)$ is placed in a $1 \times 1 \times 1$ domain. The sphere shrinks under the velocity field $\overrightarrow{v(x,y,z,t)} = -0.1 \cdot H(x,y,z,t) \cdot \overrightarrow{n(x,y,z,t)}$, where $H(x,y,z,t)$ and $\overrightarrow{n(x,y,z,t)}$ are the mean curvature and outward unit normal of surface point (x,y,z) at time t respectively. The radius $R(t)$ of the sphere evolves over time according to

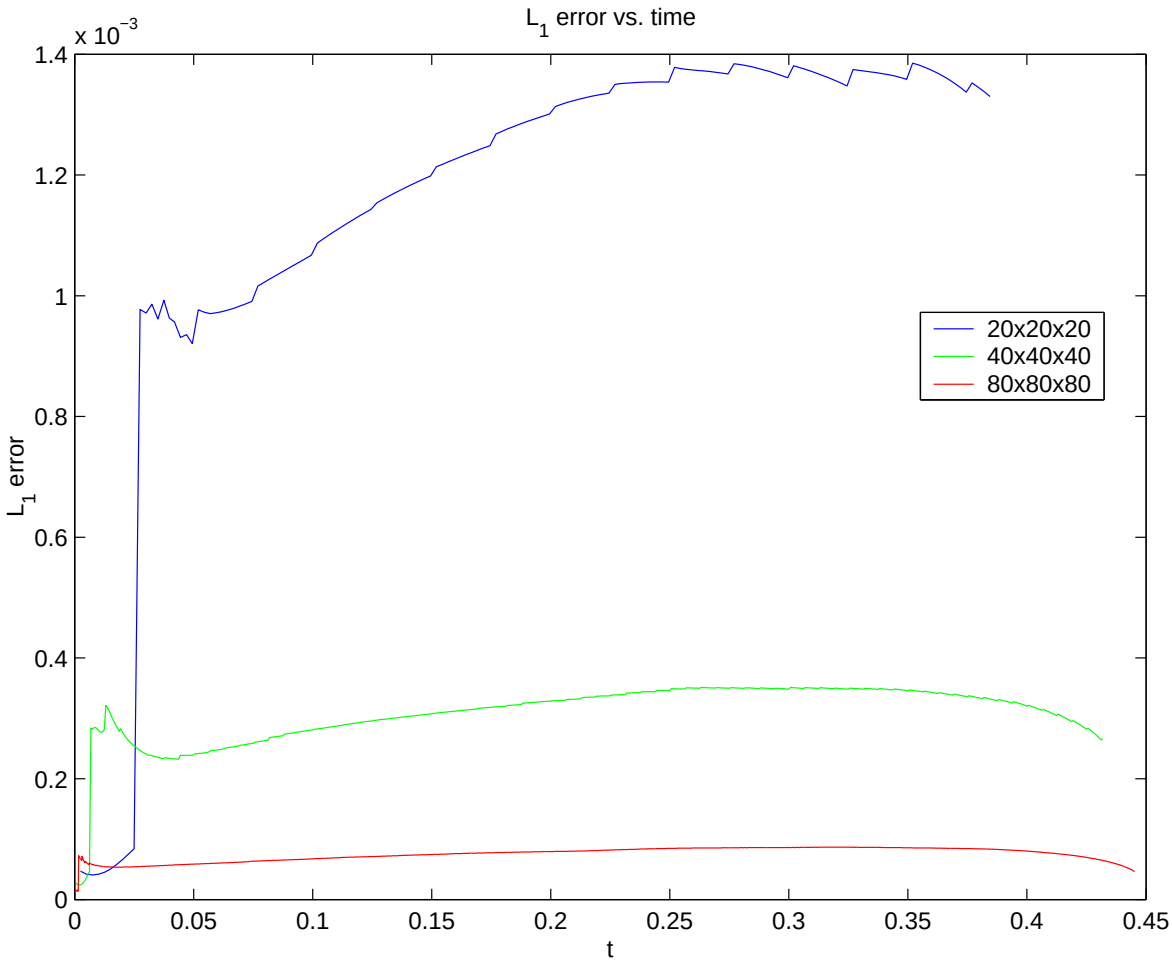


Figure 4.1: L_1 error norm of the shrinking sphere under $20^3, 40^3$ and 80^3 meshs.

$\frac{dR}{dt} = -0.1/R$. The analytical solution of this motion is:

$$R(t) = \sqrt{R_0^2 - 0.2t},$$

where R_0 is the initial radius of the sphere. By Theorem 4.1.3, the sphere will shrink smoothly to a single point at $t = 0.45$.

We performed tests using three different mesh sizes: $20^3, 40^3$ and 80^3. A CFL condition was set to ensure $dt = o(\triangle\, x^2)$. We plotted the L_1 error norm in Figure 4.1 and tabulated the maximum L_1 error norm in Table 4.2. Our tests showed accurate

Mesh	20^3	40^3	80^3
Maximum L_1 norm	1.4×10^{-3}	3.5×10^{-4}	8.6×10^{-5}

Table 4.2: Maximum amount of the L_1 norm of the error of the shrinking sphere evolving under mean curvature

convergence to the analytical solution as the mesh size was refined.

Dumbbell Test

We simulate a well-known example (collapse of a dumbbell under mean curvature)[105] to demonstrate how Locally Grid Based interface tracking could handle formulation of singularities and topological changes present in 3D mean curvature flow. This example was used by Grayson [58] to show that non-convex shapes in three dimensions may in fact not shrink to one sphere.

Consider a dumbbell in a $2 \times 1 \times 1$ domain. The dumbbell is made up of two spheres centered along x-axis, each of radius 0.3 and connected by a cylindrical handle of radius 0.15. This dumbbell shape initial interface evolves under mean curvature velocity field $\overrightarrow{v(x,y,z,t)} = -H(x,y,z,t) \cdot \overrightarrow{n(x,y,z,t)}$, where $H(x,y,z,t)$ and $\overrightarrow{n(x,y,z,t)}$ is the mean curvature and outward unit normal of surface point (x,y,z) at time t respectively. We use a $100 \times 50 \times 50$ mesh. The handle narrows as the surface shrinks, until it pinches off and the dumbbell separates into two pieces. These two pieces continue to shrink while acquiring a more spherical shape. Figure 4.1.6 shows the interface at four different times.

Despite the difficulties with traditional front tracking method in simulating mean curvature driven motions, the above examples demonstrate that the enhanced front tracking with Locally Grid Based tracking is well suited for complex mean curvature driven interface motions, including those that involve topological changes.

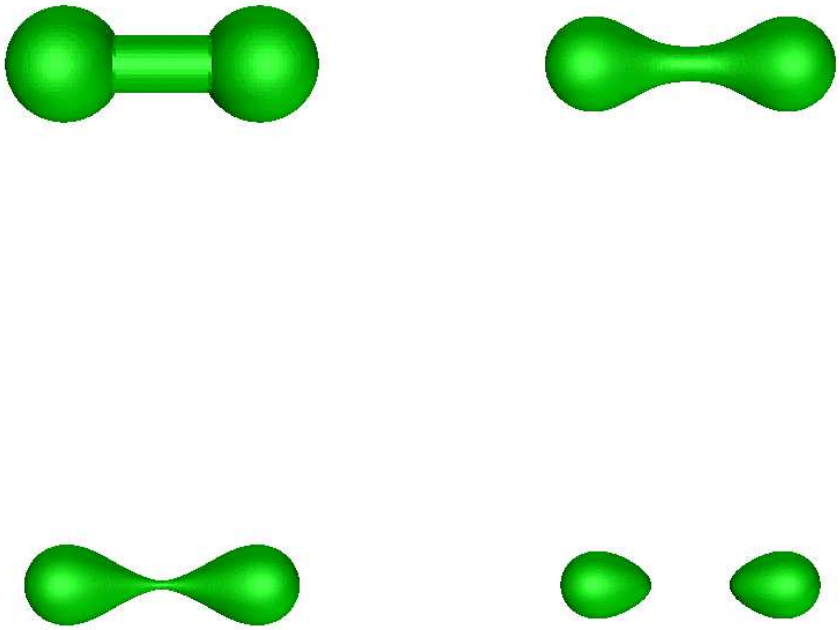


Figure 4.2: Dumbbell shape at times: 0, 0.015, 0.024, 0.03 respectively

4.2 Numerical Study of 3D Rayleigh Taylor Growth Mixing

Turbulent mixing is an important but unfinished subject. Acceleration driven mixing has been the subject of intense investigation over the past 50 years [17, 106]. Idealized cases of steady acceleration (Rayleigh-Taylor or RT) and impulsive (Richtmyer-Meshkov or RM) mixing have been studied by theory [20, 93], experiment [27, 98, 108, 110], and numerical simulations [28, 43, 68, 93], as documented in the proceedings of biannual conferences, *e.g.* [10, 80]. The experiments occur in vastly different time and energy scales, the theories are generally ideal, with an absence of length scales, and the numerical simulations have scales set by mesh resolution, and ultimately by computer budgets and the decade in which the simulation is performed. Furthermore, not all relevant experimental scales (specifically the amplitude of the long wave length initial perturbations) were recorded. In this context, efforts to compare theory, simulation, and experiment have been generally successful within a factor of two, but the absence of better agreement has led to alternate explanations regarding unresolved differences.

Rayleigh-Taylor phenomena occurs on a variety of scales, from laser fusion to turbulent combustion in a supernova. Figure 4.3 is a mosaic image, one of the largest ever taken by NASA's Hubble Space Telescope of the Crab Nebula, a six-light-year-wide expanding remnant of a star's supernova explosion. RT instabilities are especially obvious in the Crab Nebula, in which hot gas from the explosion is ramming into the surrounding interstellar medium, and they give rise to the familiar clumpy appearance of material in these and several other astronomical objects.

Our group conducted a new series of 3D Rayleigh-Taylor chaotic mixing simulations based on the enhanced 3D front tracking with Locally Grid Based tracking in *FronTier*. Detailed results were published in a series of papers [42, 83, 84].

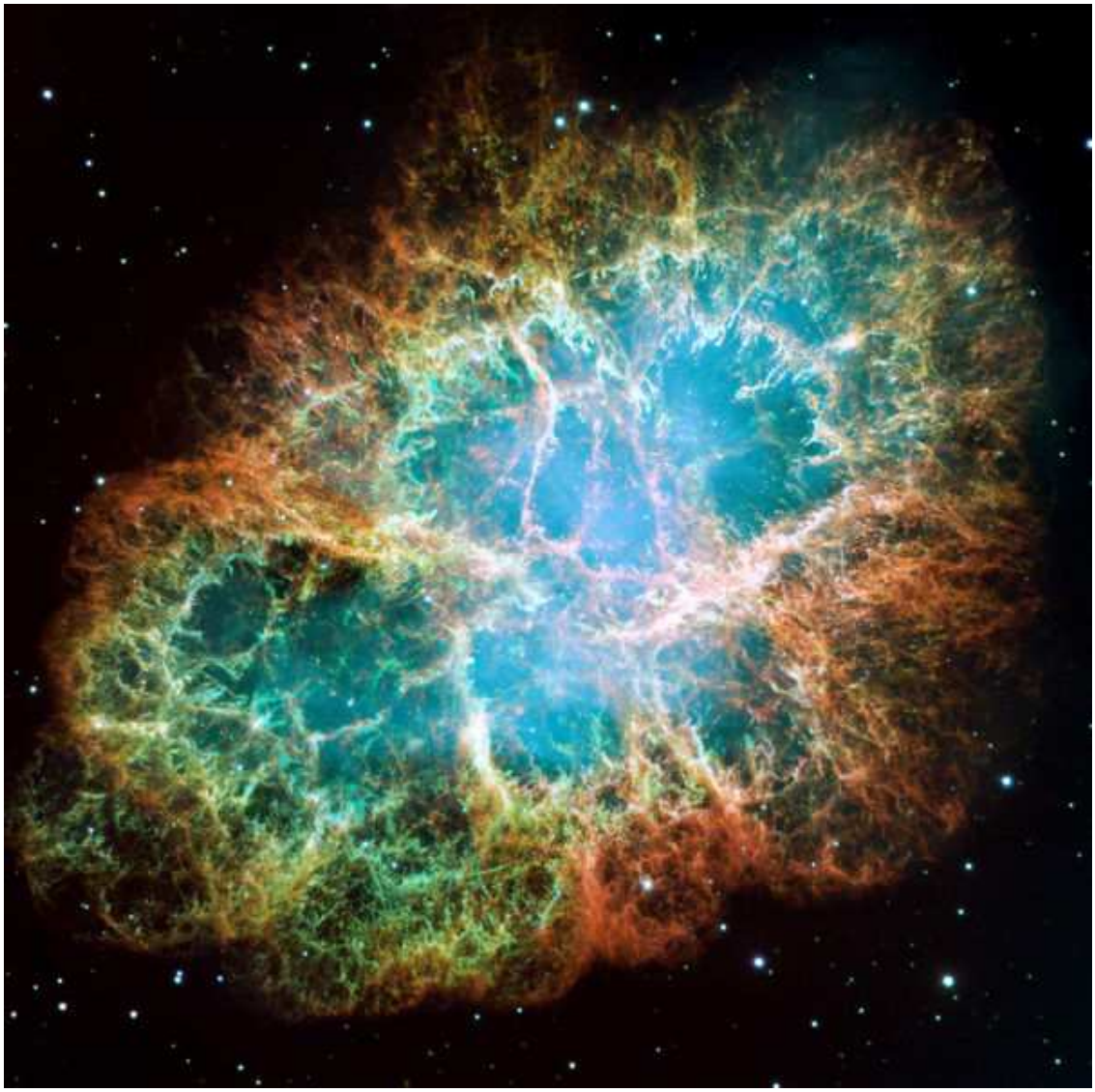


Figure 4.3: Rayleigh Taylor instabilities evident in Crab Nebula. The material was created for NASA by STScI under Contract NAS5-26555 and for ESA by the Hubble European Space Agency Information Centre.

We only summarize the main results here. We achieved agreement with experimental data. Our simulations also reveal the sensitivity of Rayleigh-Taylor mixing rates to physical and numerical scale breaking phenomena and to scale breaking diffusive or interface smoothing artifacts. Thus accurate numerical tracking to control numerical mass diffusion and accurate modelling of physical scale breaking phenomena (surface tension, physical mass diffusion or viscosity) were identified to be significant contributors to turbulent RT mixing rates.

Chapter 5

Front Tracking with Limited Mass Diffusion

The advantage of front tracking over capturing methods for the numerical simulation of discontinuity surfaces in fluid flow is to eliminate the numerical diffusion of mass across a fluid interface. For accurate modelling of an interface between miscible fluids, however, physical mass diffusion may be significant, and must be added to the calculation.

In this chapter, we introduce a new front tracking algorithm in which the physical diffusion of mass across a tracked interface is included. The accuracy and convergence properties of this algorithm are discussed. The new algorithm is a subgrid algorithm in the sense that the asymptotic fine grid behavior is identical to conventional untracked (capturing) methods while the coarse grid behavior is improved. It is thus most suitable for modelling small values of physical mass diffusion, for which adequate numerical resolution is computationally demanding. The mixing rates for the 3D Rayleigh-Taylor instability of miscible fluids based on this algorithm agree with experimental values.

This is a joint work with Dr. Xinfeng Liu under the supervision of Dr. James Glimm and Dr. Xiaolin Li. The design and implementation of subgrid model was done by Dr. Xinfeng Liu.

5.1 Introduction

The conservation laws

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} &= \Delta \cdot \nu \rho \ , \\ \frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \rho \mathbf{v}\mathbf{v} &= -\nabla P + \rho g + \nabla \cdot \nu \mathbf{v} \nabla \rho \ , \\ \frac{\partial (\rho E)}{\partial t} + \nabla \cdot ((\rho E + P)\mathbf{v}) &= \rho \mathbf{v} \cdot g + \nabla \cdot \nu E \nabla \rho \ , \end{aligned} \tag{5.1}$$

model the mixing of compressible miscible fluids with physical mass diffusion, where ρ is density, $\mathbf{v}$ is velocity, P is pressure, g is gravity, E is total specific energy, and ν is the coefficient of mass diffusion.

The difficulty in solving this system is to eliminate numerical diffusion across an interface between distinct fluids while allowing the correct amount of physical mass diffusion, in the limit where ν is small relative to the affordable grid resolution. The front tracking method [9, 21, 29, 63, 67, 86] totally eliminates numerical diffusion, but until now it has not allowed the inclusion of small amounts of physical mass diffusion. In this chapter, we introduce a new algorithm, building on the front tracking method, to add small amounts of physical mass diffusion while preserving the elimination of numerical mass diffusion across an interface. The improved algorithm is based on the analytical solution of the diffusion equation in one dimension.

Front tracking provides sharp resolution of wave fronts through the active tracking of interfaces between distinct materials. It is a numerical method that represents interfaces explicitly as a lower dimensional mesh moving through a volume filling grid. The time step of the front tracking code *FronTier* consists of two parts, the finite difference interior solver for the states defined on the volume filling rectangular grid and front propagation for the front position and states defined on each side of the front, see [21, 45, 46, 50, 51, 61]. This propagation of the front points and front

state variables distinguishes front tracking from other numerical methods. We use a directionally split method, which breaks the front propagation into normal and tangential steps. For the normal front propagation, we project Eq. (5.1) to the front normal direction to yield the one dimensional system

$$\begin{aligned}
\tfrac{\partial \rho}{\partial t} + \tfrac{\partial \rho v_N}{\partial N} &= \nu \tfrac{\partial^2 \rho}{\partial N^2}, \\
\tfrac{\partial \rho v_N}{\partial t} + \tfrac{\partial (\rho v_N^2 + P)}{\partial N} &= \rho g_N + \nu \tfrac{\partial v_N}{\partial N}\tfrac{\partial \rho}{\partial N} + \nu v_N \tfrac{\partial^2 \rho}{\partial N^2}, \\
\tfrac{\partial \rho v_T}{\partial t} + \tfrac{\partial \rho v_N v_T}{\partial N} &= \nu \tfrac{\partial v_T}{\partial N}\tfrac{\partial \rho}{\partial N} + \nu v_T \tfrac{\partial^2 \rho}{\partial N^2}, \\
\tfrac{\partial \rho E}{\partial t} + \tfrac{\partial (\rho E v_N + P v_N)}{\partial N} &= \rho g_N v_N + \nu \tfrac{\partial E}{\partial N}\tfrac{\partial \rho}{\partial N} + \nu E \tfrac{\partial^2 \rho}{\partial N^2} \ ,
\end{aligned} \tag{5.2}$$

where $\frac{\partial}{\partial N}$ is the directional derivative in the normal direction $\vec{N}$, $v_N = \mathbf{v}\cdot\vec{N}$, $g_N = g\cdot\vec{N}$, and $v_T = \mathbf{v} - v_N\vec{N}$ is the tangential velocity component.

Our primary results are to introduce the new, limited diffusion tracking algorithm, to test this algorithm in 1D examples, and to discuss its extension to 3D. Sample 3D simulation results with experimental validation will be summarized also. In §5.2 the new algorithm to solve the continuity equation with physical diffusion in one dimension is introduced, making use of the front tracking approach. In §5.3 we present computational verification evidence for the proposed algorithm. In §5.4, we extend the algorithm to higher dimensions. We validate the 3D algorithm by comparing 3D simulations of the Rayleigh-Taylor instability to experimental results.

5.2 Physical Mass Diffusion in One Dimension

The new algorithm, developed first in 1D, is based on the following ideas. Untracked contact discontinuities give rise to a blurred or smeared out front. We preserve the tracking of a sharp front and introduce physical mass diffusion through it as a perturbation. Thus the time step is split into two parts, the first being the usual non

diffusive front tracking [30, 48], and the second a pure physical mass diffusion step. Since the first step has been described previously, we only describe the second, physical mass diffusion, step. Conceptually, the front states stored on the tracked front represent the states at $\pm\infty$ relative to the scale of the diffusion layer and the diffusion correction of the interior states near the front will be carried along the characteristics. For the mass diffusion step at each time step, we compare two algorithms. The first is a subgrid algorithm which allows limited mass diffusion according to the analytic form of the solution for the diffusion equation. The second is a finite difference (FD) algorithm. It is conceptually simpler, and computes the desired diffusion across the front based on the interior (nonfront) states, with no regard for any tracked front which might occur within the difference stencil. After the diffusion layer has reached a width of $2\Delta x$, the first algorithm is turned off and replaced with the second. In referring to the finite difference algorithm, we understand the case in which it is used for all times, not just after the diffusion layer width is comparable to the mesh spacing. Both algorithms appear to give satisfactory results, but the subgrid algorithm is superior in computing the amount of mass diffused through the interface. These two algorithms are compared to a finite difference algorithm without tracking.

The subgrid algorithm starts with a reconstruction of the diffusion transition layer. Consider the convection equation with physical diffusion

$$\rho_t + u\rho_x = \nu\rho_{xx} \tag{5.3}$$

and the initial condition

$$\rho(x,0) = \begin{cases} \rho_{-\infty} & x < x_0 \\ \rho_{\infty} & x > x_0 \end{cases}$$

where ν is the physical diffusion coefficient. This initial value problem can be solved

exactly

$$\rho(x,t) = \rho_{-\infty} + \frac{\rho_\infty - \rho_{-\infty}}{\sqrt{\pi}} \int_{-\infty}^{(x-X)/\sqrt{4\nu t}} e^{-\xi^2} d\xi \ , \tag{5.4}$$

where $X = x_0 + ut$.

When Eq. (5.3) is solved numerically, the finite difference (or finite volume) equation is equivalent to the following modified equation

$$\rho_t + u\rho_x = (\nu + \tilde{\nu})\rho_{xx}, \tag{5.5}$$

where $\tilde{\nu}$ is the numerical diffusion. As a result, the numerical solution is approximately

$$\bar{\rho}(x,t) = \rho_{-\infty} + \frac{\rho_\infty - \rho_{-\infty}}{\sqrt{\pi}} \int_{-\infty}^{(x-X)/\sqrt{4(\nu+\tilde{\nu})t}} e^{-\xi^2} d\xi. \tag{5.6}$$

The diffusion widths for the exact solution and numerical solution are $d = \pi\sqrt{\nu t}$ and $\bar{d} = \pi\sqrt{(\nu + \tilde{\nu})t}$ respectively. Since $\tilde{\nu} \to 0$ as $\Delta x \to 0$, $\bar{d} \to d$ under the mesh refinement.

However, for a large scale computation in three dimensions, mesh refinement is expensive and requires large increases of memory and CPU time. For example, each doubling of the number of grid points in each dimension requires an 8 times increase of memory and a 16 times increase of CPU time. We wish to use finite and affordable computational mesh to simulate realistic physical mass diffusion when the physical diffusion coefficient is substantially smaller than the numerical diffusion coefficient.

Assume that we know the density transition, $\rho^n_{\pm\infty}$, at the current step, the location of the density transition (midpoint), X^n, at the current step, and the amount of mass which has diffused through the interface in either direction up to the current step, $\mathcal{M}^n_\pm$. The density transition and its location are stored as front states and front positions, and updated as part of the (non-diffusive) front and interior state update.

The diffused mass, $\mathcal{M}_{\pm}^{n}$, is a new variable. It is a diffusion progress variable, and it is updated incrementally within the mass diffusion step. On this basis, we define an analytic solution

$$\rho(x,t^n) = \rho_{-\infty} + \frac{\rho_{\infty}-\rho_{-\infty}}{\sqrt{\pi}} \int_{-\infty}^{(x-X^n)/\sqrt{4\nu t^n}} e^{-\xi^2} d\xi \ . \tag{5.7}$$

The solution of (5.7) depends on a time t^n, defined by the property that the integral of $\rho - \rho_{-\infty}$ from $-\infty$ to X^n is just $\mathcal{M}_{-}^{n}$.

Thus we define the current (pseudo) time t^n in terms of $\mathcal{M}_{\pm}^{n}$ and the current density states at infinity, through the formula

$$\begin{aligned} \mathcal{M}_{+}^{n} &= \int_0^{\infty}(\rho(x,t^n)-\rho_{+\infty}^{n})dx = (\rho_{-\infty}^{n}-\rho_{+\infty}^{n}) \int_0^{\infty} \left(1 - \tfrac{1}{\sqrt{\pi}} \int_{-\infty}^{\frac{x\sqrt{\pi}}{\sqrt{4\pi\nu t^n}}} e^{-\xi^2} d\xi \right) dx \ , \\ \mathcal{M}_{-}^{n} &= \int_{-\infty}^{0}(\rho(x,t^n)-\rho_{-\infty}^{n})dx = \tfrac{\rho_{+\infty}^{n}-\rho_{-\infty}^{n}}{\sqrt{\pi}} \int_{-\infty}^{0} \int_{-\infty}^{\frac{x\sqrt{\pi}}{\sqrt{4\pi\nu t^n}}} e^{-\xi^2} d\xi dx \ . \end{aligned} \tag{5.8}$$

Notice $\mathcal{M}_{-}^{n} = -\mathcal{M}_{+}^{n}$. The increased mass diffused through the middle point at step n is defined as

$$\Delta\mathcal{M}_{\pm}^{n} = \mathcal{M}_{\pm}^{n+1} - \mathcal{M}_{\pm}^{n} \ , \tag{5.9}$$

where $\mathcal{M}_{\pm}^{n+1}$ is defined the same as in (5.8), but with the pseudo time t^n replaced by $t^n + \Delta t$.

As a convention [23], the diffusion layer is reconstructed as a simple piecewise linear curve, defined by a line segment tangent to $\rho(x,t^n)$ through the transition mid point X^n and cutting the horizontal lines $\rho = \rho_{\pm\infty}$. The edges of the reconstructed transition layer are located at $X^n \pm d^n$, where

$$d^n = d(t^n) = \frac{|\rho_{\infty}-\rho_{-\infty}|}{2|\max_x \rho'(x,t^n)|} \approx \pi\sqrt{\nu t^n} = d^{n-1}\sqrt{1+\frac{\pi^2\nu\Delta t^n}{(d^{n-1})^2}}$$

and the max is computed within the transition zone.

$\mathcal{M}_{\pm}^{n}$ is defined incrementally from its previous value by adding the amount of mass to diffuse in the current step. This amount is defined by the analytic solution of (5.7), using the equivalent time t^n defined in terms of $\mathcal{M}_{\pm}^{n}$ and the current time step size Δt. The diffused mass is added to the interior state on one side of the front and subtracted from the interior state on the other.

The subgrid algorithm is applied as long as $d^n < \Delta x$. When $d^n \geq \Delta x$ or if the finite difference algorithm rather than the subgrid algorithm is being used, then the diffusion step is a pure centered second order finite difference step, applied to the interior states, with no regard for the front locations, and no use of ghost cells near the front. In either case, the diffusion step defines an amount of mass to be added to or subtracted from that located in the interior states at locations adjacent to the front.

Assume that the velocity and total energy are slowly varying through the transition layer, in comparison to the density variation. On this basis, we also treat the diffusion terms added to the momentum and energy equations as a diffusional correction, and define a parabolic update step for these equations.

To update the interior states, we use operator splitting to separate the hyperbolic from the parabolic terms. A regular stencil is one which does not meet the front. The diffusion term is solved by conventional centered finite differences. For an irregular stencil, if the front cuts a mesh cell not at the center of the stencil, we define ghost cell extrapolation of the states on the same side of the front as the center cell using the front states as in [53]. In case the front cuts the central cell of the stencil, *i.e.,* the cell that the stencil is updating, we use the new algorithm explained below. We denote by $\widetilde{\rho}_i^{n+1}, \widetilde{(\rho v)}_i^{n+1}, \widetilde{(\rho E)}_i^{n+1}$ the states at x_i^n after the hyperbolic update. Let $\Delta\mathcal{M}_{\pm}^{n}$ be defined as above. This mass must be added to the mass described by the

interior states. We distribute this increment of diffused mass to the two closest grid cells which lie on each side of the center of the layer. In order to do this, we detect the closest grid center points on either side of the front center point X_c^{n+1}, namely x_i and x_{i+1}. We add to these cells the mass diffused from the other side

$$\begin{aligned}
\rho_i^{n+1} &= \widetilde{\rho}_i^{n+1} + \tfrac{\Delta\mathcal{M}_-^n}{\Delta x} \ , \\
(\rho v)_i^{n+1} &= \widetilde{(\rho v)}_i^{n+1} + \tfrac{\tilde{v}_i^n \Delta\mathcal{M}_-^n}{\Delta x} \ , \\
(\rho E)_i^{n+1} &= \widetilde{(\rho E)}_i^{n+1} + \tfrac{\tilde{E}_i^n \Delta\mathcal{M}_-^n}{\Delta x} \ , \\
\rho_{i+1}^{n+1} &= \widetilde{\rho}_{i+1}^{n+1} + \tfrac{\Delta\mathcal{M}_+^n}{\Delta x} \ , \\
(\rho v)_{i+1}^{n+1} &= \widetilde{(\rho v)}_{i+1}^{n+1} + \tfrac{\tilde{v}_{i+1}^n \Delta\mathcal{M}_+^n}{\Delta x} \ , \\
(\rho E)_{i+1}^{n+1} &= \widetilde{(\rho E)}_{i+1}^{n+1} + \tfrac{\tilde{E}_{i+1}^n \Delta\mathcal{M}_+^n}{\Delta x} \ .
\end{aligned} \tag{5.10}$$

5.3 Validation

First we compare algorithms for a 1D pure diffusion and transport problem in Fig. 5.1. We set a constant velocity field $v = 0.5$, and study the mass diffusion across a density jump. We compare the exact solution (obtained from a fine grid untracked numerical method) to the subgrid algorithm, the tracked FD algorithm and an untracked FD algorithm. The later three algorithms are computed on a coarse grid, with the final time and the physical values of mass diffusion equivalent to that used in the 3D RT simulation for all figures of Sect. 5.3. We choose $\nu = 0.0008$ in all simulations shown in Sect. 5.3. On the scale of three or so mesh blocks, we see that the untracked algorithm is wrong, the tracked FD algorithm is good and the subgrid algorithm is excellent. The excessive mass diffusion in the untracked solution results from the transport within the Euler equation step and not from the computation of diffusion per se.

Fig. 5.2 compares density contours with different algorithms for the pure trans-

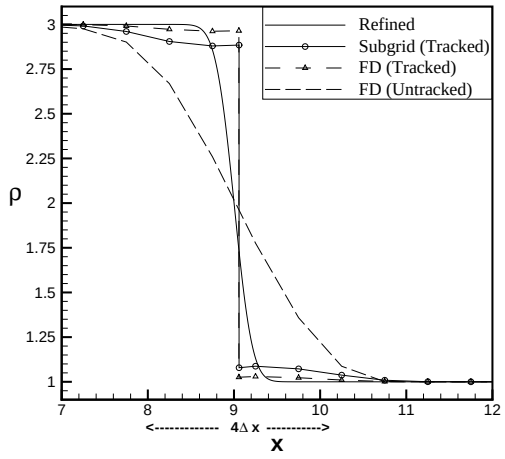


Figure 5.1: Comparison of algorithms for the 1D diffusion and transport problem. Plot of density *vs* time, displayed after 2000 coarse grid time steps and an equivalent physical time for the fine grid.

port and diffusion problem, from which we can see that the subgrid (tracked) algorithm is less diffused than the untracked algorithm.

Table 5.1 compares the L_1 errors for different algorithms for the diffusion and transport problem. We denote by $\rho_{\text{exact}}(x, t^n)$ the exact solution, which can be obtained by a numerical solution using a very fine grid and by $\rho(x, t^n)$ the numerical solution computed on a coarse grid, and interpreted as piecewise linear between grid points. For the two tracked algorithms, the grid cell which contains the front is divided into two fractional cells on either side of the front, and the front states provide the additional data to reconstruct a linear solution in each subcell. We calculate the L_1 error at time t^n by $\int_{-\infty}^{\infty} |\rho(x, t^n)_{\text{exact}} - \rho(x, t^n)| dx$.

Table 5.2 compares the mass diffused through the diffusion layer midpoint X^n at time t^n. For untracked simulations, X^n is defined as the midpoint location of the density within the transition layer. This location is, in general, not a grid point, and so the diffused mass calculation involves use of a fractional cell. The diffused

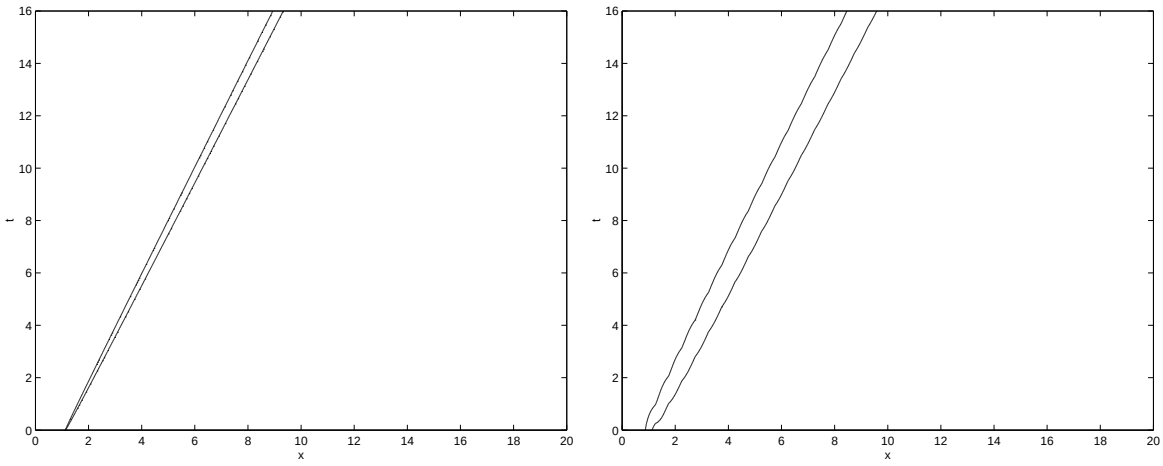


Figure 5.2: Comparison of coarse mesh space time density contours for different algorithms. Two contours are shown, for $\rho = 1.5$ and $\rho = 2.5$, that is 25% and 75% of the density change through the transition layer. To avoid stairstep plots of coarse grid piecewise constant functions, we use reconstruction of the transition layer as defined in Sec. 5.2. For the untracked FD algorithm, we interpolate data between adjacent grid points. Left: subgrid (tracked) algorithm. The diffusion layer has a width of about $2d^n = 0.8\Delta x$ at time $t = 16$. Right: FD (untracked) algorithm. The numerically computed diffusion layer has a width of about $3\Delta x$ at time $t = 16$.

Mesh	Subgrid (Tracked)	FD (Tracked)	FD (Untracked)
40	0.30	0.29	1.02
80	0.28	0.26	0.57
160	0.21	0.20	0.25

Table 5.1: L_1 error comparison for different algorithms.

Mesh	Subgrid (Tracked)	FD (Tracked)	FD (Untracked)
40	0.240	0.071	1.274
80	0.242	0.129	0.735
160	0.252	0.218	0.480
4000	0.264	0.264	0.264

Table 5.2: Comparison of diffused mass for different algorithms at the same time with different grid sizes. The coarsest mesh corresponds to the late time 3D simulation.

mass is computed by $2\int_{-\infty}^{X^n}|\rho(x,t^n)-\rho_{-\infty}|dx$. From this comparison, we see that the subgrid algorithm is far more accurate in this measure of convergence than the FD (tracked and especially untracked) algorithms for a coarse grid, and that the tracked (FD and subgrid) algorithm can eliminate the numerical diffusion across the interface completely. All algorithms agree with a fine grid.

Fig 5.3 shows the convergence of the subgrid and the tracked FD algorithms. Both algorithms have good L_1 error norms, even for the coarse grid, with a slight advantage for the FD algorithm.

Fig. 5.4 compares the algorithms for a shock-contact interaction, again with an imposed velocity field. The left frame is before and the right frame after the interaction of the shock with the contact. The same three algorithms are compared to a fine grid solution. Fig. 5.5 compares the numerical algorithms before (left) and after (right) a rarefaction wave interacts with the contact interface, parameters chosen as above. From these three plots, we conclude that the tracked subgrid algorithm and the tracked FD algorithm are nearly equivalent, and both are much better than the completely untracked solution.

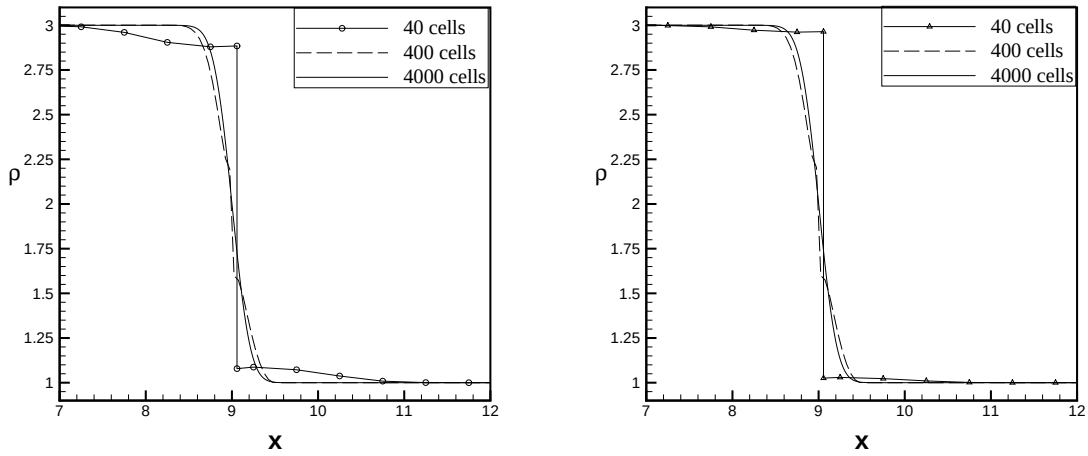


Figure 5.3: Convergence of the tracked subgrid and the tracked FD algorithms for the same 1D diffusion and transport problem. Left: the tracked subgrid algorithm; right: the tracked FD algorithm. The coarsest grid coincides with that of Fig. 5.1.

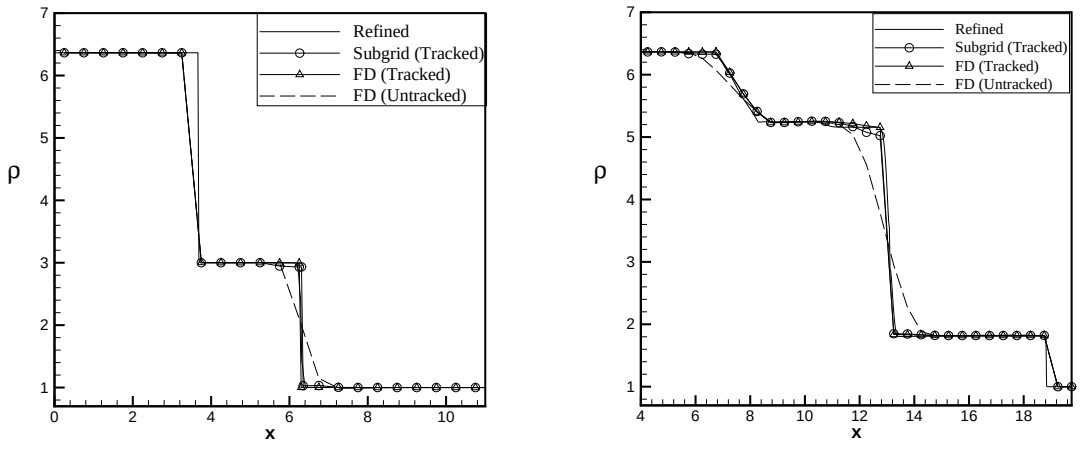


Figure 5.4: 1D comparison of algorithms for a shock contact interaction problem. Left: before the shock interacts with the contact; right: after the shock passes through the contact.

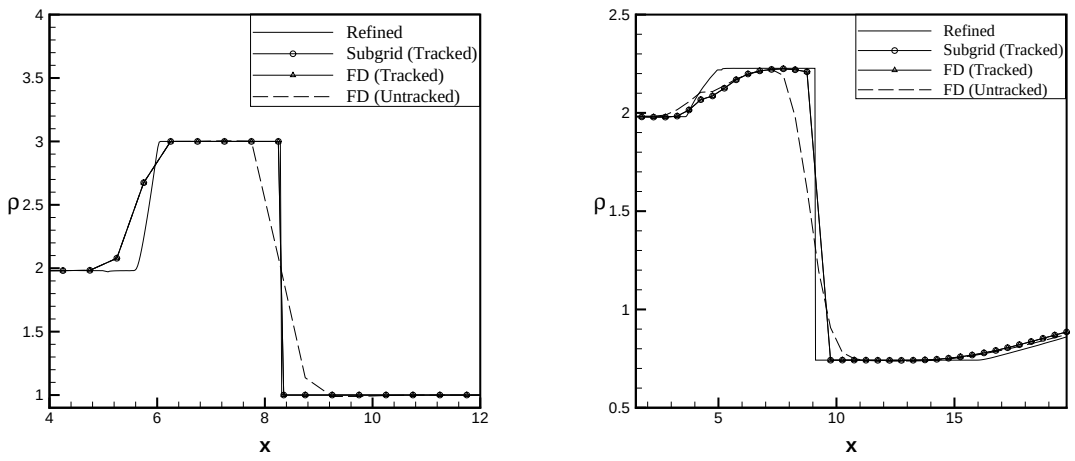


Figure 5.5: 1D comparison of algorithms for a rarefaction contact interaction problem. Left: before the rarefaction interacts with the contact; right: after the rarefaction passes through the contact.

5.4 Extension to Higher Dimensions

For the higher dimension case, we split the front into normal and tangential directions. The subgrid algorithm is applied along the normal direction, and no diffusion is added to the tangential sweep. In this way, the subgrid algorithm in one dimension can be extended easily and efficiently to the higher dimension. We do not present results from a 3D extension of the tracked FD algorithm, but we remark that it also must be solved in normal-tangential coordinates to avoid over-diffusing the mass. The subgrid algorithm is validated by comparing a 3D simulation of the Rayleigh-Taylor (RT) instability to laboratory experiments [8]. The RT instability occurs at a fluid interface whenever the density gradient is opposed to the acceleration gradient across the interface. The RT mixing rate is the dimensionless coefficient α in the equation

$$h = \alpha A g t^2 \tag{5.11}$$

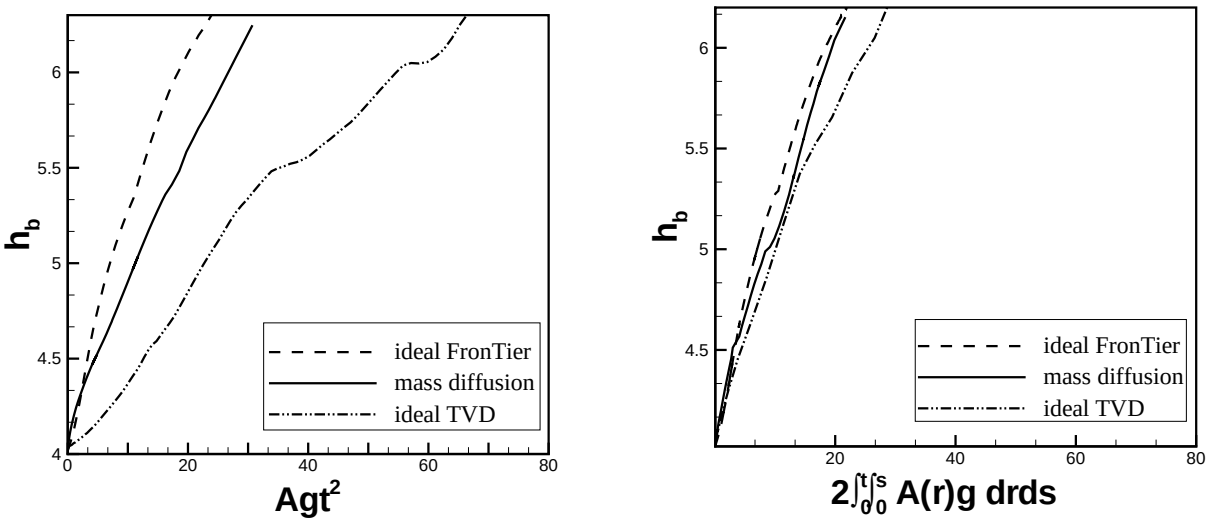


Figure 5.6: Left: Self similar growth of the mixing zone. Right: The same data plotted using a time dependent Atwood number, to remove the effects of numerical or physical mass diffusion.

for the height h of the bubbles, *i.e.,* the interpenetration distance h of the light fluid into the heavy fluid. Here $A = (\rho_h - \rho_l)/(\rho_h + \rho_l)$ is the Atwood number and t is the time. Acceptable experimental values for α are $\alpha = 0.06 \pm 0.01$ [8, 108]. See [106] for background information. To remove effects of mass diffusion (physical and/or numerical), we follow [41] to define a time dependent Atwood number $A(t)$, and the renormalized growth rate α_{ren},

$$\alpha_{\text{ren}} = \frac{h}{2\int_0^t \int_0^s A(r) g dr ds} \ .$$

Our 3D validation results are summarized in Table 5.3. See also Fig. 5.6. Details regarding this simulation have been published separately [83].

Experiment Simulation	Comment	α
Andrews	Miscible [8]	0.07
FronTier	Miscible	0.069
TVD	Untracked [41], Ideal	0.035
FronTier	Ideal	0.09
TVD	Untracked, ideal (renormalized)	0.076
FronTier	Miscible, ideal (renormalized)	0.089

Table 5.3: Mixing rates compared for an air-helium 3D Rayleigh-Taylor experiment and related simulations. The simulations compare physical mass diffusion to ideal physics (no diffusion) and they compare tracked to untracked algorithms. The agreement of the tracked simulation with physical mass diffusion with the experiment is excellent, while the ideal simulations do not agree with experiment, nor (because of the numerical mass diffusion in the untracked ideal simulation) with each other.

5.5 Discussion

We have introduced a new subgrid algorithm to combine physical mass diffusion with the tracking of an interface (to eliminate numerical mass diffusion between the two fluids). The algorithm has been tested in typical 1D wave interaction problems, extended to 3D and then compared to a laboratory experiment of RT mixing rates, with basically perfect agreement. We can also identify two other issues that could contribute opposite effects and at least in part cancel each other, subgrid or unresolved mass diffusion due to unresolved subgrid interfaces and long wave length initial conditions.

Viscosity will be very small for the air-helium experiment but large for other experiments, see the discussion in [83]. The extension of the subgrid algorithm to include the viscosity will definitely be useful and will be considered in our future studies.

We omit tangential mass diffusion as this effect is not directly coupled to the diffusion related decrease of bouyancy forces that is activated by the mass diffusion

normal to the interface. Similarly we regard curvature related corrections to the diffusion process as secondary. Such phenomena are not considered here but could be studied in following investigations.

This result can be compared to typical untracked simulations, which generally under-predict mixing in comparison to experiment by a factor of 2 [28]. In the study [41], we identified numerical mass diffusion as an explanation of the discrepancy between untracked simulations and experiment.

The total variation diminishing (TVD) algorithm [79] gives a value for the growth rate $\alpha = 0.035$, while for other simulation codes [28], α varies from 0.023 to 0.030. We list three differences between the untracked simulation [79] and the others [28] with the potential to explain these differences: [79] used 2 times the mesh resolution per mode, it used artificial compression to reduce mass diffusion and it used a different numerical algorithm (TVD).

In previous work, we obtained agreement with experimental data for RT mixing of immiscible fluids [42], namely, $\alpha_{\text{numerical}} = 0.067$, $\alpha_{\text{experiment}} = 0.063 \pm 0.013$.

Combining the present study and this previous one, we can state that improved numerical modelling of interfaces (via front tracking) and improved physics modelling (via inclusion of scale breaking phenomena) is an important factor to our agreement between numerical simulation and experiment. We are not aware of other 3D RT mixing simulations with comparable agreement with experiment.

Chapter 6

Conclusions

We have several main conclusions from this study.

1 Locally Grid Based interface tracking combines the advantages of Grid Free tracking and Grid Based tracking. It is accurate and robust and demonstrated superior performance through series of benchmark tests.

2 Locally Grid Based interface tracking conquered two major difficulties faced by traditional front tracking method while modelling mean curvature flow. It is well suited for complex mean curvature driven interface motions, including those that involve topological changes.

3 Our new series of 3D Rayleigh-Taylor chaotic mixing simulations based on the enhanced 3D front tracking with Locally Grid Based tracking achieved agreement with experimental data. Improved numerical modelling of interfaces (via front tracking) and improved physics modelling (via inclusion of scale breaking phenomena) are two important factors to our agreement between numerical simulation and experiment.

Bibliography

[1] S. Allen and J. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metall.*, 27:1084–1095, 1979.

[2] S. Angenent. Parabolic equations for curves on surfaces (i). cureves with p-integrable curvature. *Annals of Math.*, 132:451–483, 1990.

[3] S. Angenent. On the formation of singularities in the curve shortening flow. *J. Diff. Geom.*, 33:601–633, 1991.

[4] S. Angenent. Parabolic equations for curves on surfaces (ii). intersections, blow up and generalized solutions. *Annals of Math.*, 133:171–215, 1991.

[5] E. Aulisa, S. Manservisi, and R. Scardovelli. A surface marker algorithm coupled to an area-preserving marker redistribution method for three-dimensional interface tracking. *J. Comput. Phys.*, 197:555–584, 2004.

[6] G. Baker, D. Meiron, and S. Orszag. Vortex simulations of the Rayleigh-Taylor instability. *Phys. Fluids*, 23:28–64, 1980.

[7] C. Bandle, A. Brillard, G. Dziuk, and A. Schmidt. Course on mean curvature flow, 1994. Lecture Notes.

[8] A. Banerjee and M. J. Andrews. Statistically steady measurements of rayleigh-taylor mixing in a gas channel. *Phys. Fluids*, 18, 2006.

[9] M. Ben-Artzi. The generalized riemann problem for reactive flow. *J. Comp. Phys.*, 86:70–101, 1989.

[10] D. Besnard, C. Cavailler, L. Dagens, P. Figeac, M. de Gliniasty, J. F. Haas, P. A. Holstein, J. Montigny, C. Parisot, V. Rupert, B. Sitt, and N. Wilke, editors. *Proceedings of Third International Workshop on the Physics of Compressible Turbulent Mixing at Royaumont, France*. CEA DAM, 1991.

[11] K. A. Brakke. *The motion of a surface by its mean curvature*. Princeton University Press, New Jersey, 1978.

[12] K. A. Brakke. The surface evolver. *Experimental Mathematics*, 1:141–165, 1992.

[13] L. Bronsard and R. V. Kohn. Motion by mean curvature as the singular limit of ginzburg-landau dyanmics. *J. Diff. Eq.*, 90:211–237, 1991.

[14] L. Bronsard and B.T.R. Wetton. A numerical method for tracking curve networks moving with curvature motion. *J. Comp. Phys.*, 120:66–87, 1993.

[15] G. Caginalp. The dynamics of a conserved phase field system: Stephan-like, Hele-Shaw and Cahn-Hilliard models as asymptotic limits. *IMA J. applied Math.*, 44:77–94, 1990.

[16] F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Comput. Aid. Geom. Des.*, 22:121–146, 2005.

[17] S. Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability*. Oxford University Press, Oxford, 1961.

[18] S. Chen, D. Johnson, P. Raad, and D. Fadda. The surfer marker and micro cell method. *Int. J. Numer. Meth. Fluids*, 25:749–778, 1997.

[19] Y. G. Chen, Y. Giga, and S. Goto. Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations. *J. Diff. Geom.*, 33:749–786, 1991.

[20] B. Cheng, J. Glimm, and D. H. Sharp. A 3-D RNG bubble merger model for Rayleigh-Taylor mixing. *Chaos*, 12:267–274, 2002.

[21] I-L. Chern, J. Glimm, O. McBryan, B. Plohr, and S. Yaniv. Front tracking for gas dynamics. *J. Comp. Phys.*, 62:83–110, 1986.

[22] L. P. Chew. Constrained delaunay triangulations. *Algorithmica*, 4:97–108, 1989.

[23] A. Chorin and J. Marsden. *A Mathematical Introduction to Fluid Mechnics*. Springer Verlag, New York–Heidelberg–Berlin, 2000.

[24] P. de Mottoni and M. Schatzman. Evolution geometrique d'interfaces. *C. R. Acd. Sci. Paris*, 309:453–458, 1989.

[25] P. de Mottoni and M. Schatzman. Development of interfaces in rn. *Proc. Royal Soc. Edin.*, 116A:207–220, 1990.

[26] M. Desbrun, M. Meyer, P. Schroder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2cmanifolds, 2003. Visualization and Mathematics III, H.-C.Hege and K. Polthier, Ed.

[27] G. Dimonte and M. Schneider. Density ratio dependence of Rayleigh-Taylor mixing for sustained and impulsive acceleration histories. *Phys. Fluids*, 12:304–321, 2000.

[28] G. Dimonte, D. L. Youngs, A. Dimits, S. Weber, M. Marinsk, S. Wunsch, C. Garsi, A. Robinson, M. Andrews, P. Ramaprabhu, A. C. Calder, B. Fryxell, J. Bielle, L. Dursi, P. MacNiece, K. Olson, P. Ricker, R. Rosner, F. Timmes, H. Tubo, Y.-N. Young, and M. Zingale. A comparative study of the turbulent Rayleigh-Taylor instability using high-resolution three-dimensional numerical simulations: The alpha-group collaboration. *Phys. Fluids*, 16:1668–1693, 2004.

[29] J. Donea, S. Giuliani, and J. P. Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33:689–723, 1982.

[30] Jian Du, Brian Fix, James Glimm, Xiaolin Li, Yunhua Li, and Lingling Wu. A simple package for front tracking. *J. Comp. Phys.*, 213:613–628, 2006.

[31] G. Dziuk. An algorithm for evolutionary surfaces. *Numerische Mathematik*, 58:603–611, 1990.

[32] K. Ecker. *Regularity Theory for Mean Curvature Flow*. Springer, New York, 2004.

[33] K. Ecker and G. Huisken. Mean curvature evolution of entire graphs. *Annals of Math.*, 130:453–471, 1989.

[34] K. Ecker and G. Huisken. Interior estimates for hypersurfaces moving by mean curvature. *Inventiones Mathematicae*, 105:547–569, 1991.

[35] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.*, 183:83–116, 2002.

[36] L. C. Evans and J. Spruck. Motion of level sets by mean curvature i. *J. Diff. Geom.*, 33:635–681, 1989.

[37] J. M. Floryan and H. Rasmussen. Numerical methods for viscous flow with moving boundaries. *Appl. Mech. Rev.*, 42:323–340, 1989.

[38] V. E. Fradkov, M. E. Glicksman, M. Palmera, J. Nordberg, and K. Rajan. Topological rearrangements during 2d normal grain growth. *Physica D*, 66:50–60, 1993.

[39] H. Frost, C. Thompson, C. Howe, and J. Whang. A two-dimensional computer simulation of capillarity-driven grain growth: preliminary results. *Scripta Metallurgica*, 22:65–70, 1988.

[40] T. Gatzker and C. Grimm. Improved curvature estimation on triangular meshes, 2003. Eurographics Symposium on Geometry Processing.

[41] E. George and J. Glimm. Self similarity of Rayleigh-Taylor mixing rates. *Phys. Fluids*, 17:054101–1–054101–13, 2005. Stony Brook University Preprint number SUNYSB-AMS-04-05.

[42] E. George, J. Glimm, X. L. Li, Y. H. Li, and X. F. Liu. The influence of scale breaking phenomena on turbulent mixing rates. *Phys. Rev. Lett.*, 2005. In press. Stony Brook University Preprint number SUNYSB-AMS-05-11.

[43] E. George, J. Glimm, X. L. Li, A. Marchese, and Z. L. Xu. A comparison of experimental, theoretical, and numerical simulation Rayleigh-Taylor mixing rates. *Proc. National Academy of Sci.*, 99:2587–2592, 2002.

[44] J. Glimm, M. J. Graham, J. W. Grove, X.-L. Li, T. M. Smith, D. Tan, F. Tangerman, and Q. Zhang. Front tracking in two and three dimensions. *Comput. Math. Appl.*, 35(7):1–11, 1998.

[45] J. Glimm, J. W. Grove, X. L. Li, W. Oh, and D. H. Sharp. A critical analysis of Rayleigh-Taylor growth rates. *J. Comp. Phys.*, 169:652–677, 2001.

[46] J. Glimm, J. W. Grove, X.-L. Li, K.-M. Shyue, Q. Zhang, and Y. Zeng. Three dimensional front tracking. *SIAM J. Sci. Comp.*, 19:703–727, 1998.

[47] J. Glimm, J. W. Grove, X.-L. Li, and D. C. Tan. Robust computational algorithms for dynamic interface tracking in three dimensions. *SIAM J. Sci. Comp.*, 21:2240–2256, 2000.

[48] J. Glimm, J. W. Grove, X.-L. Li, and N. Zhao. Simple front tracking. In G.-Q. Chen and E. DiBenedetto, editors, *Contemporary Mathematics*, volume 238, pages 133–149. Amer. Math. Soc., Providence, RI, 1999.

[49] J. Glimm, J. W. Grove, W. B. Lindquist, O. McBryan, and G. Tryggvason. The bifurcation of tracked scalar waves. *SIAM Journal on Computing*, 9:61–79, 1988.

[50] J. Glimm, J. W. Grove, and Y. Zhang. Interface tracking for axisymmetric flows. *SIAM J. SciComp*, 24:208–236, 2002. LANL report No. LA-UR-01-448.

[51] J. Glimm, E. Isaacson, D. Marchesin, and O. McBryan. Front tracking for hyperbolic systems. *Adv. Appl. Math.*, 2:91–119, 1981.

[52] J. Glimm, X.-L. Li, R. Menikoff, D. H. Sharp, and Q. Zhang. A numerical study of bubble interactions in Rayleigh-Taylor instability for compressible fluids. *Phys. Fluids A*, 2(11):2046–2054, 1990.

[53] J. Glimm, D. Marchesin, and O. McBryan. Stable and unstable fluid interface surfaces in petroleum engineering. Technical Report preprint, Rockefeller Univ., 1980.

[54] J. Glimm, D. Marchesin, and O. McBryan. Subgrid resolution of fluid discontinuities II. *J. Comp. Phys.*, 37:336–354, 1980.

[55] J. Glimm and O. McBryan. A computational model for interfaces. *Adv. Appl. Math.*, 6:422–435, 1985.

[56] J. Goldfeather and V. Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Transactions On Graphics*, 23:45–63, 2004.

[57] M. A. Grayson. The heat equation shrinks embedded curves to round points. *J. Diff. Geom.*, 26:285–314, 1987.

[58] M. A. Grayson. A short note on the evolution of surfaces via mean curvatures. *J. Diff. Geom.*, 58:285–314, 1989.

[59] M. A. Grayson. Shortening embedded curves. *Annals. of Math.*, 129:71–111, 1989.

[60] J. W. Grove. Anomalous waves in shock wave – fluid interface collisions. In B. Lindquist, editor, *Current Progress in Hyperbolic Systems: Riemann Problems and Computations*, volume 100 of *Contemporary Mathematics*, pages 77–90. American Mathematics Society, Providence, RI, 1989.

[61] J. W. Grove. Applications of front tracking to the simulation of shock refractions and unstable mixing. *J. Appl. Num. Math.*, 14:213–237, 1994.

[62] J. W. Grove, R. Holmes, D. H. Sharp, Y. Yang, and Q. Zhang. Quantitative theory of Richtmyer-Meshkov instability. *Phys. Rev. Lett.*, 71(21):3473–3476, 1993.

[63] J. W. Grove and R. Menikoff. The anomalous reflection of a shock wave at a material interface. *J. Fluid Mech.*, 219:313–336, 1990.

[64] Martin Held. A collection of efficient and reliable intersection tools. *J. Graphics Tools*, 2:25–47, 1997.

[65] W. Henshaw. A scheme for the numerical solution of hyperbolic conservation laws. *J. Comp. Phys.*, 68:25–47, 1987.

[66] S. E. Hieber and P. Koumoutsakos. A lagrangian particle level set method. *J. Comp. Phys.*, 210:342–367, 2005.

[67] C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *J. Comp. Phys.*, 14:227–253, 1974. Reprinted in **135** (1997), pp. 203–216.

[68] R. L. Holmes, B. Fryxell, M. Gittings, J. W. Grove, G. Dimonte, M. Schneider, D. H. Sharp, A. Velikovich, R. P. Weaver, and Q. Zhang. Richtmyer-Meshkov instability growth: Experiment, simulation, and theory. *J. Fluid Mech.*, 389:55–79, 1999. LA-UR-97-2606.

[69] R. L. Holmes, J. W. Grove, and D. H. Sharp. Numerical investigation of Richtmyer-Meshkov instability using front tracking. *J. Fluid Mech.*, 301:51–64, 1995.

[70] T. Hou, J. Lowengrub, and M. Shelley. Boundary integral methods for multiphase problems in fluid dynamics and materials science. *Submitted to J. Comp. Phys.*, 2000.

[71] G. Huisken. Flow by mean curvature of convex surfaces into spheres. *J. Differ. Geom.*, 20:237–266, 1984.

[72] G. Huisken. Asymptotic behaviour for singularities of the mean curvature flow. *J. Differ. Geom.*, 31:285–299, 1990.

[73] G. Huisken and C. Sinestrari. Mean curvature flow singularities for mean convex surfaces. *Calc. Var. PDE*, 8:1–14, 1999.

[74] J. M. Hyman. Numerical methods for tracking interfaces. *Physica D*, 12:396–407, 1984.

[75] T. Ilmanen. Elliptic regularization and partial regularity for motion by mean curvature. *Mem. Amer. Math. Soc.*, 108, 1994. No. 520.

[76] X. Jiao and H. Zha. Normal and curvature estimation using face-based quadratic fittings for general surface meshes. *Computer-Aided Geometric Design*, submitted, 2007.

[77] S. Jin, R. R. Lewis, and D. West. A comparison of algorithms for vertex normal computation. *The Visual Computer*, 21:71–82, 2005.

[78] Q. Li. *Wave Interactions and Bifurcation for Front Tracking in Three Dimensions*. PhD thesis, State University of New York at Stony Brook, 1995.

[79] X.-L. Li. Study of three dimensional Rayleigh-Taylor instability in compressible fluids through level set method and parallel computation. *Phys. Fluids A*, 5:1904–1913, 1993.

[80] P. F. Linden, D. L. Youngs, and S. B. Dalziel, editors. *Proceedings of the 4th International Workshop on the Physics of compressible turbulent mixing*. Cambridge University Press, Cambridge, England, 1993.

[81] Peter Liovic, Murray Rudman, Jong-Leng Liow, Djamel Lakehal, and Doug Kothe. A 3d unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction. *Computers and Fluids*, 2005. Submitted.

[82] J.-J. Liu, J. Glimm, and X.-L. Li. A conservative front tracking method. In *Proceedings of the Tenth International Conference on Hyperbolic Problems: Theory, Numerics, and Applications*. Yokohama Publishers, Osaka, Japan, 2005. In Press.

[83] Xinfeng Liu, Erwin George, Wurigen Bo, and James Glimm. Turbulent mixing with physical mass diffusion. *Physical Review E*, 73 (issue 5), 2006.

[84] Xinfeng Liu, Yuanhua Li, James Glimm, and Xiaolin Li. A front tracking method for limited mass diffusion. *J. of Comp. Physics*, 222:644–653, 2007.

[85] M.S. Longuet-Higgins and E.D.Cokelet. The deformation of steep surface waves on water. i.a numerical method of computation. *Proc. Roy. Soc. London Ser. A*, 350:1–26, 1976.

[86] L. G. Margolin. Introduction to "an arbitrary lagrangian-eulerian computing method for all flow speeds". *J. Comp. Phys.*, 135:198–202, 1997.

[87] N. Max. Weights for computing vertex normals from facet normals. *Journal of Graphics Tools Archive*, 4:1–6, 1999.

[88] U. F. Mayer. Numerical solutions for the surface diffusion flow in three space dimensions. *Comput. Appl. Math.*, 20:361–379, 2001.

[89] J. McHyman. Numerical methods for tracking interfaces. *Physica D*, 12:396–407, 1984.

[90] D. S. Meek and D. J. Walton. On surface normal and gaussian curvature approximations given data sampled from a smooth surface. *Comput. Aid. Geom. Des.*, 17:521–543, 2000.

[91] B. Merriman, J. Bence, and S. Osher. Diffusion generated motion by mean curvature. CAM report, Dept. of Mathematics, University of California Los Angeles 92-18, 1992.

[92] G. Moretti. Thoughts and afterthoughts about shock computations. Rep. No. PIBAL-72-37, Polytechnic Institute of Brooklyn, 1972.

[93] D. Oron, O. Sadot, Y. Srebro, A. Rikanti, Y. Yedvab, U. Alon, L. Erez, G. Erez, G. Ben-Dor, L. A. Levin, D. Ofer, and D. Shvarts. Studies in the nonlinear evolution of the Rayleigh-Taylor and Richtmyer-Meshkov instabilities and their role in inertial confinement fusion. *Lasers and Particle Beams*, 17:465–475, 1999.

[94] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi equations. *Jour. Comp. Phys*, 79:12–49, 1988.

[95] M. Paolini and C. Verdi. Asymptotic and numerical analyses of the mean curvature flow with a space-dependent relaxation parameter. *Asymptotic Analysis*, 5:553–574, 1992.

[96] S. Popinet and S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. *Int. J. Numer. Meth. Fluids*, 30:775, 1999.

[97] J. M. Rallison and A. Acrivos. A numerical study of the deformation and burst of a viscous drop in an extensional flow. *J. Fluid Mech.*, 89:191–200, 1978.

[98] K. I. Read. Experimental investigation of turbulent mixing by Rayleigh-Taylor instability. *Physica D*, 12:45–58, 1984.

[99] R. Richtmyer and K. Morton. *Difference Methods for Initial Value Problems*. Interscience, New York, second edition, 1967.

[100] W. J. Rider and D. B. Kothe. A marker particle method for interface tracking. In *Sixth International Symposium on Computational Fluid Dynamics*, 1995. AIAA-95-1717.

[101] W. J. Rider and D. B. Kothe. Stretching and tearing interface tracking methods. In *The 12th AIAA CFD Conference*, San Diego, CA, June 20, 1995. AIAA-95-1717.

[102] W. J. Rider and D. B. Kothe. Reconstructing volume tracking. *J. Comp. Phys.*, 141:112–152, 1997.

[103] S. J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *J. Comp. Phys.*, 144:603–625, 1998.

[104] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.*, 31:567–603, 1999.

[105] J. A. Sethian. Numerical algorithms for propagating interfaces: Hamilton-Jacobi equations and conservation laws. *J. Differential Geometry*, 31:131–161, 1990.

[106] D. H. Sharp. An overview of Rayleigh-Taylor instability. *Physica D*, 12:3–18, 1984.

[107] W. Shyy, H. S. Udaykumar, M. M. Rao, and R. W. Smith. *Computational Fluid Dynamics With Moving Boundaries*. Taylor & Francis, Landon, 1996.

[108] V. S. Smeeton and D. L. Youngs. Experimental investigation of turbulent mixing by Rayleigh-Taylor instability (part 3). AWE Report Number 0 35/87, 1987.

[109] P. K. Smolarkiewicz. The multi-dimensional crowley advection scheme. *Monthly Weather Review*, 110:1968–1983, 1982.

[110] D. M. Snider and M. J. Andrews. Rayleigh-Taylor and shear driven mixing with an unstable thermal stratification. *Phys. Fluids*, 6(10):3324–3334, 1994.

[111] M. Sussman and E. G. Puckett. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *J. Comput. Phys.*, 162(2):301–337, 2000.

[112] B. Swartz and B. Wendroff. Aztec: A front tracking code based on Godunov's method. *Appl. Num. Math.*, 2:385–397, 1986.

[113] H. Theisel, C. Rossl, R. Zayer, and H.-P. Seidel. Normal based estimation of the curvature tensor for triangular meshes. In *Proceeding of 12th Pacific Conference on Computer Graphics and Applications*. IEEE, 2004.

[114] Wu ting Tsai and Dick K. P. Yue. Computation of nonlinear free-surface flows. *Annu. Rev. Fluid Mech.*, 28:249–278, 1996.

[115] G. Tryggvason, B. Brunnen, A. Esmaeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y. J. Jan. A front-tracking method for the computations of multiphase flow. *J. Comput. Phys.*, 169:708–759, 2001.

[116] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, imcompressible, multi-fluid flows. *J. Comp. Phys.*, 100(1):25–37, 1992.

[117] B. White. The nature of singularities in mean curvature flow of mean-convex sets. *J. Amer. Math. Soc.*, 16:123–138, 2003.

[118] Z. L. Xu, M. Kim, W. Oh, J. Glimm, R. Samulyak, X. L. Li, and C. Tzanos. Atomization of a high speed jet. *Physics of Fluids*, 2005. Submitted. SB Preprint Number: SUNYSB-AMS-05-08.

[119] R. W. Yeung. Numerical methods in free-surface flows. *Ann. Rev. Fluid Mech.*, 14:395–442, 1982.