

Stony Brook University



OFFICIAL COPY

The official electronic file of this thesis or dissertation is maintained by the University Libraries on behalf of The Graduate School at Stony Brook University.

© All Rights Reserved by Author.

Conservative Data Collections and Comparison Study for Front-tracking

A Dissertation Presented

by

Dongyung Kim

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

May 2009

Copyright by
Dongyung Kim
2009

Stony Brook University

The Graduate School

Dongyung Kim

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

James Glimm

Advisor

Department of Applied Mathematics and Statistics

Xiaolin Li

Department of Applied Mathematics and Statistics

Xiangmin Jiao

Department of Applied Mathematics and Statistics

James Davenport

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

**Conservative Data Collections and
Comparison Study for Front-tracking**

by

Dongyung Kim

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

2009

This thesis is composed in three major parts. In the first part, we discuss the conservative data collection with the implementation of averaged surface areas and volume fraction formulas. By using these formulas, we obtain the excellent agreement in the numerical closure simulation problem. In the second part, we simulate a 3D single mode Rayleigh-Taylor instability and study the surface area. The flow morphology is far less chaotic than a comparison 2D multimode Richtmyer-Meshkov flow. There is still vortex shedding from the mushroom caps. The flow is more or less chaotic in these vortex shedding region, at late time. We study the chaotic flow near the bottom of the

mushroom caps and the top of the spike tips. Here, we performed error analysis. In the third part, we compare several compressible fluid dynamics codes in a 1D geometry suggestive of an inertial confinement fusion (ICF) application. We study several algorithms and conduct a systematic mesh refinement study. The principal difficulty in the test problem selected is a strong shock passage through two narrowly spaced density layers, leading to multiple reflected waves. On account of these difficulties, numerical convergence requires extensive mesh refinement. Such refinement is within reach in 1D, which thus provides a convenient testing platform for the convergence requirements which would not be practical for a 2D or 3D simulation. The codes compared are *FronTier*, RAGE and PROMETHEUS, and within this framework, we compare a variety of algorithms, including PPM, TVD, Godunov-split and MUSCL. We explore both tracked and untracked simulations, where the tracking (if applied) is limited to the initially defined density discontinuities. Our first main conclusion is the overall similarity of the convergence properties. Within this similarity, we note several trends. *FronTier* excels on moderate grids, while PPM excels on very fine ones.

To my parents & my wife & Grace & Esther

Table of Contents

List of Figures	xv
List of Tables	xvii
Acknowledgements	xviii
1 Introduction	1
1.1 Acceleration-Driven instabilities and Turbulent Mixing	1
1.1.1 Rayleigh-Taylor and Richtmyer-Meshkov Instabilities	2
1.2 Numerical Methods of The Front-Tracking	5
1.3 Numerical Simulations and Compressible Flow Models	9
1.3.1 Averaged Microphysical equations	10
1.3.2 Closures	13
1.3.3 Validation and Comparison of Closures	19
2 Conservative Data Collection with Volume Fractions	24
2.1 Introduction	24
2.1.1 Objective	26
2.1.2 Assumptions	27

2.2	Corner Two-Colorings	28
2.3	Interface Perimeters	36
2.3.1	Perimeter Construction	36
2.3.2	Valid Triangulations of Connected Perimeters	43
2.4	Computational Procedure	50
2.4.1	Volume Fractions	53
2.4.2	Surface Areas	56
2.4.3	Formulas	57
2.5	Implementation in <i>FronTier</i>	62
2.5.1	Standard Model	63
2.5.2	Area and Volume Functions	66
2.6	Conclusions	70
3	Single Mode Surface Area in 3D Rayleigh-Taylor Instability	71
3.1	Introduction	72
3.2	Multiscale Modeling of Single Mode 3D RT Instability	73
3.3	Conclusions	80
4	Comparison of 1D Eulerian ICF Simulations	82
4.1	Introduction	83
4.2	Numerical Methods	85
4.2.1	<i>FronTier</i>	85
4.2.2	RAGE	85
4.2.3	PROMETHEUS	86
4.3	Waves and Interactions	87

4.4	Convergence : L_1 -Norm Comparison	92
4.5	Nature and Magnitude of Solution Errors	93
4.6	Convergence Rates	95
4.7	Conclusions	98
	Bibliography	99

List of Figures

1.1	Plot of the interface between the heavy (above) and light (below) fluids at late time for a Rayleigh-Taylor fluid instability. For this single mode study, density ratio is 2:1, the Atwood number is 1/3, the peak to peak amplitude of the initial disturbance is 0.3, grid size is 25 x 25 x 250 and computational domain is 5 cm x 5 cm x 50 cm.	3
1.2	Interface reconstruction using grid-free tracking. <i>Figure is created by J. Glimm, X.L. Li and J. W. Grove [24].</i>	6
1.3	Interface reconstruction using grid-based tracking. Note the bubble/spike formation typical in acceleration-driven mixing and the frequency of poor-shaped triangles on the figure. <i>Figure is created by J. Glimm, X.L. Li and J. W. Grove [24].</i>	7
1.4	The interfacial area A_I per unit volume plotted vs. time. This plot serves to define the parameter A_I for the closure Saurel-2. Left: RT, Right: RM.	19

1.5	Comparison of the model error (1.29) for three closures. Of these, only Saurel-1 depends on the value of A_I as a fitting parameter; these plots serve to locate the best fit value of A_I ($A_I = 0$) and thus to define the Saurel-1 closure. Left: RT, Right: RM. . . .	21
2.1	14 topologically distinct, or non-isomorphic, corner 2-colorings in 3D cells. The numbering follows the enumeration of isomorphism classes. For later use we give f , the number of faces with opposite black corners, in the corner 2-colorings for which $f > 0$	31
2.2	Non-isomorphic corner 2-colorings for single cell faces. From top to bottom row, the corner 2-colorings have zero, one, or two black corners. Edge crossings are shown as small black dots. . . .	36
2.3	Interface perimeters on single cell faces. They are uniquely determined by the edge crossings for cell faces with two edge crossings (left) and they are two-fold non-unique for cell faces with four edge crossings (right). Edge crossings are shown as small black dots. . . .	37

2.4	32 distinct connected interface perimeters that can be formed on the surface of a 3D cell. We group the perimeters according to f , the number of faces with four edge crossings, and the isomorphism classes of perimeters in each non-isomorphic case. Note that we have labeled these perimeter classes and that, in most cases, labels follow the names of polyhedra given by the number of edge crossings on the perimeter. The numbering of the non-isomorphic cases follows the enumeration of isomorphism classes for corner 2-colorings.	39
2.5	12 distinct disconnected interface perimeters that can be formed on a 3D cell surface, for $f = 0, 1, 2, 3$. The numbering of the non-isomorphic cases follows the enumeration of isomorphism classes for corner 2-colorings.	40
2.6	48 distinct disconnected interface perimeters that can be formed on a 3D cell surface in the non-isomorphic case with $f = 6$ (case 14).	41
2.7	Examples of perimeters in the duodecagonoid-I (left) and duodecagonoid-II (right) isomorphism classes of connected perimeters of case 14, with $f = 6$	47

2.8	Examples of interface surfaces bounded by valid perimeters belonging to isomorphism classes of connected perimeters with $T > 0$. They are, from left to right, starting at the top, the corner (case 1), the edge (case 2), the hexagonoid (case 3), the glider (case 5), the twisteroid (case 6), the plane (case 8), the twister-I (case 9), the twister-II (case 10), the hexagon (case 11), and the octagonoid-I (case 12). Note that the perimeter class of case 12 has two distinct elements. All triangulations are picked arbitrarily.	49
2.9	Valid triangulations for elements in the isomorphism classes of connected perimeters. In the table, we give for each class f , the number of faces with four edge crossings on a common face; n , the number of edge crossings on the perimeter; C_{n-2} , the Catalan number; and T , the number of valid triangulations. Only triangulations for perimeter classes with $f = 0$ were used in our implementation. In these cases, the identity $T = C_{n-2}$ holds; otherwise, $T < C_{n-2}$	51
2.10	Decomposition of the space occupied by domain 1 into five tetrahedra in one of five possible triangulations in the glider case (case 5). The crossings are the vertices \mathbf{p}_i , where $i = 1, \dots, 5$. The black corners are labeled A , B and C	55

2.11	Standard model for the fourteen cases implemented in <i>FronTier</i> : 1) Plane, 2) Edge, 3) Corner, 4) Glider, 5) Hexagon, 6) Two corners-I, 7) Twister-I, 8) Edge and corner, 9) Two corners-II, 10) Twister-II, 11) Two edges, 12) Three corners, 13) Glider and corner, 14) Four corners.	64
2.12	Location of the cell corners used in the standard model.	65
3.1	Plot of the interface between the heavy (above) and light (below) fluids at initial time for Rayleigh-Taylor fluid instability. For this study, density ratio is 2:1, the Atwood number is 1/3, the peak to peak amplitude of initial disturbance is 0.06, grid size is 20 x 20 x 200 and computational domain 1 <i>cm</i> x 1 <i>cm</i> x 10 <i>cm</i>	74
3.2	Plot of early time for Rayleigh-Taylor fluid instability. Flow and grid parameters as in Fig. 3.1.	74
3.3	Plot of middle time for Rayleigh-Taylor fluid instability. Flow and grid parameters as in Fig. 3.1.	75
3.4	Plot of late time for Rayleigh-Taylor fluid instability. Flow and grid parameters as in Fig. 3.1.	76
3.5	Plot of the interface surface area vs. time using physical units. Results for three mesh levels are displayed for the identical 3D single mode RT instability.	78

3.6	Plot of the interface area divided by the mixing zone volume vs. time. Both the area and the volume are measured in mesh units (the quantity [physical area / physical volume] $\times \Delta x$ is plotted). Results for three mesh levels are displayed for a 3D single mode RT instability.	79
4.1	Plot of initial density vs. position. $\rho_l = 1$. Left to right, $\rho_h = 4$, $\rho_h = 8$ and $\rho_h = 16$	85
4.2	Plot of final time density vs. position for the RAGE simulation. Left to right: $\rho_h = 4, 8, 16$. The results for <i>FronTier</i> and PROMETHEUS are visually identical.	87
4.3	Enlarged view of the region near the origin in Fig. 4.2. Plots of density vs. position x , for $x \in [0, 0.1]$. Left to right: $\rho_h = 4, 8, 16$ using <i>FronTier</i> . The results for RAGE and PROMETHEUS are visually similar.	88
4.4	Shock-contact wave interaction schema. S : shock, T : transmitted, R : reflected, NC : new contact.	89
4.5	Density vs. position plot at time $t = 0.02$ (left) and $t = 0.04$ (right) with $\rho_h = 4$ in <i>FronTier</i>	90
4.6	Density vs. position plot at time $t = 0.05$ (left) and $t = 0.06$ (right) with $\rho_h = 4$ in <i>FronTier</i>	90
4.7	Density vs. position plot at time $t = 0.07$ (left) and $t = 0.08$ (right) with $\rho_h = 4$ in <i>FronTier</i>	91

4.8	Wave position vs. time plot (left) and detailed plot between $t = 0$ and $t = 0.1$ with $\rho_h = 4$ (right) in <i>FronTier</i> . RAGE and PROMETHEUS is visually similar.	91
4.9	Left to right: Grid size 400 and 1600 L_1 -error with $\rho_h = 4$. . .	93
4.10	Left to right: Grid size 400 and 1600 L_1 -error with $\rho_h = 16$. . .	94
4.11	C: contact, S-C: shock-contact interaction. Left : Local wave error in units of mass vs. time with $\rho_h = 4$ and grid size 1600, <i>FronTier</i> -MUSCL. ORS: outgoing reflected shock (ORS1= S_R , ORS2= S_{TRT}), PS: primary shock, NFS: new forward shock, SWR: shock after wall reflection. Right : Local wave error for original four contacts, $\rho_h = 4$ and grid size 1600, <i>FronTier</i> -MUSCL. C: contact, S-C: shock-contact interaction, R: wall reflection, S-S: shock-shock interaction.	95
4.12	C: contact, S-C: shock-contact interaction, R: wall reflection, S-S: shock-shock interaction. Left : Local wave error for original four contacts, $\rho_h = 4$ and grid size 1600, RAGE. Right : Local wave error for original four contacts, $\rho_h = 4$ and grid size 1600, PPM.	96
4.13	The solid line values at the top are the contact error as a percentage of the total error (with new and four original contacts) vs. time. The dash-dot lower line show the error carried by the original four contacts, expressed as a percentage of the total error vs. time. S-C: shock-contact interaction, R: wall reflection, S-S: shock-shock interaction. Here $\rho_h = 4$ and the number of grid cells is 1600.	97

List of Tables

1.1	Summary of d_k^q parameter choices for RT and RM mixing. All choices except that for d_k^v in the RT case are insensitive.	17
1.2	Model errors based on comparison to simulation data.	22
1.3	Summary properties related to the closures q^* . It is remarkable that the closures depend sensitively on their defining parameter d_k only in the case of the RT data for the v^* closure.	22
2.1	Properties of all the valid interface perimeters in 3D cells. Lower roman letters indicate the number of distinct interface perimeters with similar connected components. Note that there are two distinct interface perimeters with two corners, one corresponding to the non-isomorphic case 3, and the other corresponding to the non-isomorphic case 4. Also note that all perimeters with hexagonoid, twisteroid and octagonoid-I components (in bold) were excluded in our implementation.	58
3.1	Table relating Re and λ_K to Re_{mesh} at $t = 11.0$ for several mesh levels. Kolmogorov scale $\lambda_{Kmesh} = \lambda_K/\Delta x$. Schmidt number $Sc = 5.2$	80

4.1	Summary, L_1 -error from $t=0.0$ to $t=0.5$ with $\rho_h=4$ as t depends on grid size and numerical algorithm. The first order Godunov, MUSCL, TVD, TVDAC use front Tracking; RAGE and PPM are untracked.	97
4.2	Summary of grid size and convergence order with $\rho_h=4$. The first order Godunov, MUSCL, TVD, TVDAC are front Tracking; RAGE and PPM are untracked.	98

Acknowledgements

I would like to extend my deepest gratitude to my advisor, Professor James Glimm, for his guidance and support in these past four years. His experience, dedication and formidable knowledge were a constant source of inspiration to me. I will forever treasure his teachings and hope to emulate in my professional life the rigor and integrity he brings to the scientific endeavor.

I would like to thank Professor Xiaolin Li for his kind and insightful help throughout my years of study and research. He is a remarkable teacher and an exceptional scientist and I was fortunate to have learned so much from him in my time here.

I would like to thank Dr. Bruce Fryxell for his cooperation work in our research and his advice.

I would like to appreciation to Dr. James Davenport, who kindly accepted to make the journey from Brookhaven National Laboratory and join this committee.

I would like to thank Professor Seiyong Chung who is my advisor in South Korea and Seiyong Oh, Seongrae Kim, Joonkook Shin, Deokhoon Boo, Keonhee Lee, Seongkyu Choi, Jinkoo Chu, Sekwon Oh, Byungryun Kang in Chungnam National University (CNU) and Dr. Byungsoo Moon in Korean

Atomic Energy Research Institute (KAERI).

I would like to thank my peers and friends at Stony Brook and CNU, both inside and outside the Department, for their constant help and kindness.

Finally, I would like to extend my love and gratitude to my wife, Yoojoo Koh, two daughters, Grace Kim and Esther Kim, my elder brother, Dongeon Kim, Brother' wife, Youngjoo Cho, their son and daughter, Dongha Kim, Yunha Kim, my younger sister, Dongok Kim, her husband, Yangsoo Ahn, their son, Seongbo Ahn and to my parents, Jinseok Kim, Youngsoon Park who never stopped supporting me and my grand mother, Boksoon Shin, even half a world away.

Chapter 1

Introduction

1.1 Acceleration-Driven instabilities and Turbulent Mixing

In this work we discuss the following three problems:

1. The averaged volume fractions and surface areas in 3D grid cells cut by a two-fluid interface for the conservative data collection in compressible multi-species and multi-phase flow models (Chapter 2).
2. Single mode Rayleigh-Taylor with surface area in 3D and its error analysis (Chapter 3).
3. Comparison of several compressible fluid dynamics codes with several algorithms and study of a systematic mesh refinement in 1D Eulerian simulations (Chapter 4).

Each problem has own conclusions at the end of each chapter. A fundamental understanding of these problems involves acceleration-driven hydrody-

dynamic instabilities and turbulent mixing such as the Richtmyer-Meshkov and Rayleigh-Taylor instabilities. Problem 3 is applied to development of inertial confinement fusion (ICF) technology, where the implosion of target capsules is susceptible to acceleration-driven instabilities and turbulent mixing.

1.1.1 Rayleigh-Taylor and Richtmyer-Meshkov Instabilities

Two classical acceleration-driven hydrodynamic instabilities are the Rayleigh-Taylor and Richtmyer-Meshkov instabilities. The Rayleigh-Taylor instability occurs when a light fluid steadily accelerates into a heavier fluid; the Richtmyer-Meshkov instability happens when a shock wave passes through an interface. The Rayleigh-Taylor instability applies to a heavy and light fluid mixing in a gravitational field. Since the paper in 1950 [49], many scientists have studied the Rayleigh-Taylor instability. As the RT instability develops, the perturbations in the initial contact lead to the formation of bubbles and spikes. In particular, bubbles of light fluid tend to rise up into the heavier fluid while spikes, or “fingers”, of heavy fluid tend to move downward into the lighter fluid. Fig. 1.1 shows the development of the instability at late time computed on a 5 cm x 5 cm x 50 cm 3D domain. In the late time single mode non-linear regime of the instability, the growth is typically studied using Layzer’s model [36] which gives a constant rate of bubble growth using the bubble radius and the gravitational acceleration.

George et al. [19–21] conducted a Rayleigh-Taylor simulation using the high-resolution front-tracking code *FronTier*, which prevents interfacial mass

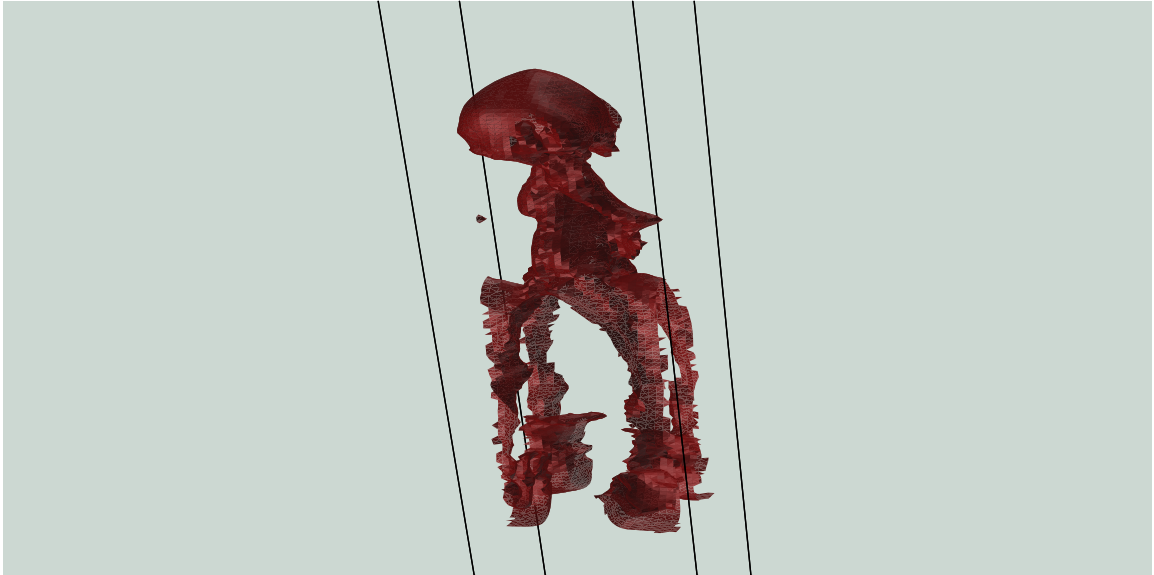


Figure 1.1: Plot of the interface between the heavy (above) and light (below) fluids at late time for a Rayleigh-Taylor fluid instability. For this single mode study, density ratio is 2:1, the Atwood number is $1/3$, the peak to peak amplitude of the initial disturbance is 0.3, grid size is $25 \times 25 \times 250$ and computational domain is $5 \text{ cm} \times 5 \text{ cm} \times 50 \text{ cm}$.

diffusion, as well as untracked simulations for comparison. Their results show that the lower acceleration rate found in the untracked simulations, such as total variation diminishing (TVD), is caused, to a large extent, by a reduced buoyancy force due to numerical interfacial mass diffusion. The bubble height penetration h , the depth to which the turbulent mixing zone penetrates the denser fluid, is given by

$$h(t) = \alpha A g t^2 \tag{1.1}$$

where $A = (\rho_1 - \rho_2) / (\rho_1 + \rho_2)$ is the Atwood number. All values of the acceleration rate α (theoretical, experimental, and from simulation) are consistent if the diffusive calculation of α is renormalized to account for mass diffusion. Refer to [20] for these developments.

In particular, the study of RT instability is very important in a wide variety of applications, for example, nuclear weapon explosions, inertial confinement fusion (ICF) and supernova explosions.

The Richtmyer-Meshkov instability, which occurs for impulsively-accelerated or shock-accelerated density interfaces, is also essential in the study of supernovae, inertial confinement fusion (ICF), and other practical processes. Moreover, it is a particularly meaningful and convenient test problem for turbulence studies. In effect, experimental measurements of RM instability provide benchmarks for validation and verification of numerical codes modeling turbulent flows. In 1960, Richtmyer provided the first rigorous treatment of this particular instability through his theoretical and numerical analysis [44]. In 1969, Meshkov's shock-tube work provided an experimental basis for Richt-

myer’s predictions [40].

As with the RT instability, the initial perturbations at the surface of the contact interface between two fluids will grow larger in amplitude once the shock passes through the interface. So, if the initial setup consists of a multi-mode perturbed sinusoidal interface between two fluids, one heavy and one light, then the perturbations will grow in amplitude and develop into crests and troughs. These will tend to grow at an experimental rate, as predicted by solutions of linearized the equations to a certain point (especially for low initial amplitudes and early times) until the instability has gone into a fully non-linear regime and the mixing layer becomes highly turbulent. At that point, the same formations observed in RT mixing of light bubbles and heavy spikes will develop with the additional features of a roll-up, or “mushrooming”, of the bubbles (due to vorticity diffusion effects) as well as the appearance of smaller-scale structures.

1.2 Numerical Methods of The Front-Tracking

In fluid dynamic equations, a new closure was proposed by Glimm et al. [4] in the next section. As a kind of high-resolution numerical simulation, our research began with a validation study of the proposed closure of fluid dynamic equations. Directly, it was simulated, collected and analyzed the data from the microphysical fluid mixing equations [3].

These fluid mixing simulations use the front-tracking software package *FronTier*. The objective of a front-tracking hydrodynamic program is to provide a numerical method that can represent interfaces explicitly as lower-

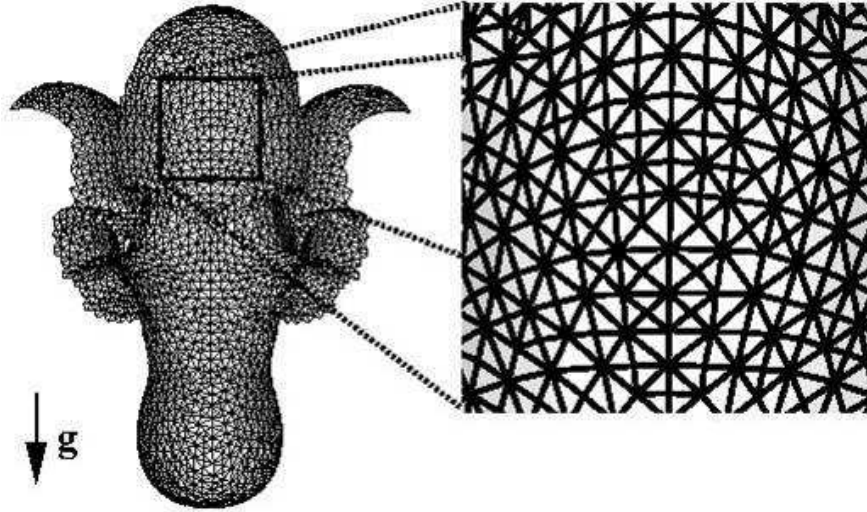


Figure 1.2: Interface reconstruction using grid-free tracking. *Figure is created by J. Glimm, X.L. Li and J. W. Grove [24].*

dimensional meshes moving through a volume-filling grid in addition to keeping track of the initial interface (the surface, or set of surfaces that separate distinct fluids) throughout the simulation. Thus, its main characteristic is to follow whatever initial interface is input by the user in a topologically correct manner. *FrontTier* is a front-tracking program that achieves this geometrical criterion in sophisticated ways, with an ability to handle interface bifurcations and restore untangled interfaces at each time step [24].

There are two approaches to front-tracking. One is *grid-free* tracking and the other is *grid-based* tracking (Fig. 1.2, 1.3). In *grid-free* tracking, the tracked front is the triangulated interface propagating freely through a rectangular volume-filling mesh. In *grid-based* tracking, the front is reconstructed at each

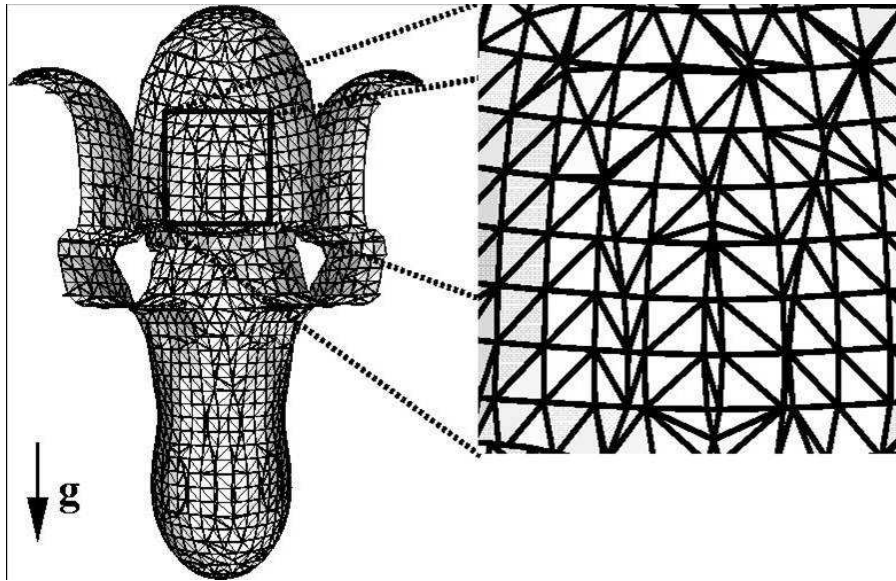


Figure 1.3: Interface reconstruction using grid-based tracking. Note the bubble/spike formation typical in acceleration-driven mixing and the frequency of poor-shaped triangles on the figure. *Figure is created by J. Glimm, X.L. Li and J. W. Grove [24].*

time step using the intersection points of the interface with all grid cell edges. Grid-based tracking is very robust but inaccurate because of numerical surface tension, a form of surface smoothing. Grid-free tracking is more accurate but less robust in resolving interface bifurcations. *FronTier* instead of using either algorithm, combines the two to preserve the advantages of each: robustness and accuracy. This hybrid solution is called *locally grid-based* tracking. The idea is simply to rely on the more accurate grid-free propagation unless there is a bifurcation. If a bifurcation occurs, the algorithm switches to a robust grid-based reconstruction inside a region around the bifurcation. The reconstructed interface inside this region is then joined to the grid-free interface outside using a construction resembling grid-based propagation [3]. Another important aspect of the *FronTier* code is its ability to support additional surface-based physics, namely surface tension and mass diffusion. These two phenomena are key in, respectively, the study of immiscible and miscible simulations. Surface tension forces a pressure jump at the interface that is proportional to the surface curvature. Thus, when solving for the mid state pressure across the front at each dynamic time step - that is, solving a Riemman problem to connect two states on each side of the front - it suffices to modify the equation of pressure equilibration according to the pressure jump to account for the presence of surface tension. To account for physical mass diffusion, numerical mass diffusion has to be eliminated while a physical mass diffusion has to be added back into the calculation on the basis of prescribed values for the physical mass diffusion constant. Effectively, the elimination step is done by *FronTier* [24].

1.3 Numerical Simulations and Compressible Flow Models

In order to study the numerical errors and the convergence properties of several averaged quantities that characterize the flow, or for the purposes of validating the proposed closures for systems of fluid dynamic equations, our research program always starts from a series of high-resolution numerical simulations. Directly, we simulate the microphysical equations of fluid mixing to extract the data that is subsequently used in our analysis [3]. The diversity of approaches and the vast spectrum of models that have been put forth by the scientific community in the theoretical study of compressible multi-species, multi-phase fluid mixing is remarkable. Some of these include phenomenological mixing models, hybrid turbulent mixing models, multi-fluid interpenetration mixing models and buoyancy drag models [6]. Our approach for modeling compressible multi-species, multi-phase flow is based on the single-phase averaging of the microphysical fluid equations (Euler equations) over an infinite ensemble of flow realizations, as explained in detail by Glimm et al. [4, 32]. The following exposition is based on the latest developments discussed in [4].

The ensemble-averaging process is applied to a microphysical description of distinct fluids separated by a sharp interface and leads to undefined averages of nonlinear quantities in the primitive variables. These need to be modeled in order to close the system of equations. Assuming homogeneity and the absence of internal length scales within the mixing zone, a general closure is proposed in the form of integral identities based on exact interfacial

terms. These closures satisfy conservation and boundary constraints for the continuity and momentum equations and also allow important scale-breaking phenomena in the equations of motion, such as mass diffusion and surface tension. Moreover, they are validated against experimental and numerical data. The model is valid for acceleration-driven mixing processes, including Rayleigh-Taylor and Richtmyer-Meshkov mixing. With the trajectories or velocities of the mixing zone edges as the phenomenological input (and without adjustable parameters for added mass, buoyancy and drag, which are typically treated phenomenologically), it is capable of describing the coherent structures dominating the mixing zone: the bubbles and spikes in the outer layer.

1.3.1 Averaged Microphysical equations

We consider two-phase flow equations. Let the function X_k be the phase indicator for material k ($k = 1, 2$). $X_k(t, \mathbf{x})$ equals 1 if the position \mathbf{x} is in fluid k at time t , otherwise it is zero. Let us apply the averaging process to the microphysical equations. We average the advection law [14] for the indicator function X_k of the region occupied by the fluid k ,

$$\frac{\partial X_k}{\partial t} + v_{\text{int}} \cdot \nabla X_k = 0 . \quad (1.2)$$

Here v_{int} is the microphysical velocity evaluated at the interface (the velocity component normal to the boundary ∂X_k is continuous so that $v_{\text{int}} \cdot \nabla X_k$ is well defined).

We also average the microscopic conservation equations (continuity, mo-

mentum and energy equations):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0 , \quad (1.3)$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \rho \mathbf{v} \mathbf{v} = -\nabla p + \rho \mathbf{g} , \quad (1.4)$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot \rho \mathbf{v} E = -\nabla \cdot p \mathbf{v} + \rho \mathbf{v} \mathbf{g} . \quad (1.5)$$

As usual, \mathbf{v} , ρ , p , and E denote, respectively, the velocity, density, pressure, and total energy with $E = e + \mathbf{v}^2/2$ where e is the internal energy.

To account for scale-breaking physics, such as mass diffusion or surface tension, new terms must be added on the right-hand side of these equations. See [4] for equations that account for surface tension and transport. For simplicity we assume no mass diffusion across an interface for mixing processes with immiscible fluids. Additionally, we can assume no surface tension for mixing with miscible fluids.

Applying the ensemble average to Eqs. (1.2)-(1.5), we typically consider averaged equations with one dependent space variable (say, in the z-direction for a planar case) so that the ensemble to be averaged is independent of the missing variables. The ensemble average then includes an averaging over the suppressed spacial variables. To allow for statistical ensembles possessing cylindrical or spherical geometry we introduce the geometry indicator s , where $s = 0, 1, 2$ corresponds to planar, cylindrical and spherical forms of the conservation equations. When the equations are averaged in cylindrical or spherical coordinates, covariant centrifugal forces due to the curvilinearity of

the coordinate systems are introduced into the averaged equations.

Multiphase equations for the phase k are obtained by multiplying Eqs. (1.2)-(1.5) by X_k and performing an ensemble average. We denote the ensemble average by $\langle \cdot \rangle$. The average $\langle X_k \rangle$ of the indicator function X_k is denoted β_k ; $\beta_k(z, t)$ is then the expected fraction of the horizontal layer at height z that is occupied by fluid k at time t . The quantities ρ_k and p_k are, respectively, phase averages of the density ρ and pressure p :

$$\rho_k = \frac{\langle X_k \rho \rangle}{\langle X_k \rangle}, \quad p_k = \frac{\langle X_k p \rangle}{\langle X_k \rangle}. \quad (1.6)$$

The quantities v_k and E_k are phase mass-weighted averages of the fluid z -velocity v_z and total energy E :

$$v_k = \frac{\langle X_k \rho v_z \rangle}{\langle X_k \rho \rangle}, \quad E_k = \frac{\langle X_k \rho E \rangle}{\langle X_k \rho \rangle}. \quad (1.7)$$

We obtain the one-dimensional, two-pressure, two-phase flow averaged equations by applying the ensemble average to Eqs. (1.2)-(1.5).

$$\frac{\partial \beta_k}{\partial t} + \langle \mathbf{v} \cdot \nabla X_k \rangle = 0, \quad (1.8)$$

$$\frac{\partial(\beta_k \rho_k)}{\partial t} + \frac{1}{z^s} \frac{\partial}{\partial z} (z^s \beta_k \rho_k v_k) = 0, \quad (1.9)$$

$$\frac{\partial(\beta_k \rho_k v_k)}{\partial t} + \frac{1}{z^s} \frac{\partial}{\partial z} (z^s \beta_k \rho_k v_k^2) + \frac{\partial(\beta_k p_k)}{\partial z} = \left\langle p \frac{\partial X_k}{\partial z} \right\rangle + \beta_k \rho_k g, \quad (1.10)$$

$$\frac{\partial(\beta_k \rho_k E_k)}{\partial t} + \frac{1}{z^s} \frac{\partial}{\partial z} [z^s \beta_k v_k (\rho_k E_k + p_k)] = \langle p \mathbf{v} \cdot \nabla X_k \rangle + \beta_k \rho_k v_k g, \quad (1.11)$$

for the volume fraction β_k , velocity v_k , density ρ_k , pressure p_k , and total energy E_k of phase k . Here $k = 1 = b$ (bubble) and $k = 2 = s$ (spike) denote the light and heavy fluids respectively, $k' = 3 - k$ and $g = g(t) > 0$ is the gravitational acceleration.

We add further terms to the right-hand side of these equations when considering surface tension and transport effects, as shown in [4].

1.3.2 Closures

Let v^* , p_k^* and $(p_kv)^*$ denote the averaged quantities at the interface. These interfacial terms are given by

$$v^* = \frac{\langle \mathbf{v} \cdot \nabla X_k \rangle}{\langle \mathbf{n}_3 \cdot \nabla X_k \rangle}, \quad p_k^* = \frac{\langle p_k \mathbf{n}_3 \cdot \nabla X_k \rangle}{\langle \mathbf{n}_3 \cdot \nabla X_k \rangle}, \quad (p_kv)^* = \frac{\langle p_k \mathbf{v} \cdot \nabla X_k \rangle}{\langle \mathbf{n}_3 \cdot \nabla X_k \rangle}, \quad (1.12)$$

where \mathbf{n}_3 is the unit normal vector in the preferred direction. Although the equations (1.12) allow multiple fluids, they only allow a single interface type (*i.e.*, at most two fluid phases). A generalization [11] of this framework to multiple phases specifies an interface type for each pair of phases that are in contact, and leads to a generalization of the system (1.12).

The definitions (1.12) are fundamental to all that follows. They are mathematically exact consequences of the averages of Eqs. (1.3)-(1.5) and specify the quantities (the right hand side of (1.12)) that are to be approximated in a definition of closure to complete the averaged equations (1.8)-(1.11).

Note that ∇X_k equals the unit normal to the boundary ∂X_k times a

delta function concentrated on ∂X_k . The definitions assume that interface fluxes weighted by this vector measure are proportional to fluxes through the z direction only. Also for an interface quantity such as p^* , which may be discontinuous across the interface (due to surface tension), the notation p_k^* indicates evaluation from the interior X_k side of ∂X_k .

Here we explain the basic ideas of our proposed closures, and their relation to the closures of Saurel and coworkers [1, 45, 46]. There are three terms to define in (1.12), $v^*, p^*, (pv)^*$, the interface averages of p, v, pv , which we denote generically as q^* , $q = v, p, pv$. For the Saurel et al. closed equations, these same quantities are also required (and have different definitions). Their equations contain additional terms called relaxation terms.

We assume a closure which represents q^* as a convex sum of the primitive variables q_k for $q = v, p$,

$$q^* = \mu_1^q q_2 + \mu_2^q q_1, \quad q = v, p \quad (1.13)$$

and a related bilinear expression

$$(pv)^* = p^*(\mu_1^{pv} v_2 + \mu_2^{pv} v_1) + v^*(\mu_1^{pv} p_2 + \mu_2^{pv} p_1) - (\mu_1^{pv} p_2 v_2 + \mu_2^{pv} p_1 v_1) \quad (1.14)$$

for $(pv)^*$.

Saurel et al. employ a related but different functional form for their closure. They also consider a convex sum

$$q'^{S*} = \mu_1^{Sq} q_2 + \mu_2^{Sq} q_1 \quad q = v, p, \quad (1.15)$$

and then define

$$v^{S*} = v'^{S*} + \operatorname{sgn} \left(\frac{\partial \beta_1}{\partial x} \right) \frac{p_2 - p_1}{\mathcal{Z}_1 + \mathcal{Z}_2} , \quad (1.16)$$

$$p^{S*} = p'^{S*} + \operatorname{sgn} \left(\frac{\partial \beta_1}{\partial z} \right) \left(\frac{\mathcal{Z}_1 \mathcal{Z}_2}{\mathcal{Z}_1 + \mathcal{Z}_2} \right) (v_2 - v_1) . \quad (1.17)$$

Here \mathcal{Z}_k is the acoustic impedance of fluid k . Also

$$(pv)^{S*} = p^{S*} v^{S*} . \quad (1.18)$$

The Saurel closures also include relaxation, as an additional term in the RHS of the associated volume fraction, momentum and energy equations. These terms (for $k = 1$) have the form:

$$\mu^S (p_1 - p_2) \quad \text{volume fraction source term} \quad (1.19)$$

$$\lambda^S (v_2 - v_1) \quad \text{momentum source term} \quad (1.20)$$

$$\lambda^S v'^{S*} (v_2 - v_1) - \mu^S p'^{S*} (p_1 - p_2) \quad \text{energy source term} \quad (1.21)$$

Here

$$\mu^S = \frac{A_I}{2(\mathcal{Z}_1 + \mathcal{Z}_2)} , \quad \lambda^S = \mu \mathcal{Z}_1 \mathcal{Z}_2 , \quad (1.22)$$

where A_I is the interfacial area per unit volume.

The convex coefficients μ_k^q and μ_k^{Sq} have a fractional linear form, namely

$$\mu_k^q = \frac{\beta_k}{\beta_k + d_k^q \beta_{k'}} , \quad q = v, p, pv \quad (1.23)$$

$$\mu_k^{Sq} = \frac{1}{1 + d_k^{Sq}} , \quad q = v, p \quad (1.24)$$

The μ_k^q and μ_k^{Sq} thus depend on a single parameter d_k^q or d_k^{Sq} . The form of the μ_k^q is suggested by rewriting the exact equations to derive an exact (unclosed) expression for q^* and μ_k^q . See theorems in [4] for these results. The form (1.23) of μ_k^q is motivated by these expressions as well and is, moreover, required by theoretically required boundary conditions at the edges of the mixing zone,

$$q^* = q_k \quad \text{at} \quad z = Z_k . \quad (1.25)$$

Our closures satisfy (1.25). A preliminary choice of d_k^q , motivated by a homogenization of the exact expressions mentioned above, is given in [4]. In all but one case, the d_k^q are insensitive and are arbitrarily set to 1. See Table 1.1.

For the RT case, we assume $(-1)^k V_k = (-1)^k \dot{Z}_k \geq 0$ so that the mixing zone is expanding. The growing mixing zone entrains pure phase fluid into the mixture, and thus creates mixed fluid volume for both phases. In the incompressible, non-diffusive RT case, this is seen clearly from the closed form solution

$$d_k^v(t) = \left(\frac{Z_{k'}}{Z_k} \right)^s \left| \frac{V_{k'}}{V_k} \right| . \quad (1.26)$$

The solution (1.26) is a consequence of the closed form expression obtained for the solution of the model equations and a simple calculation. See references of Glimm, Sharp and Saltz [27–29] for details of the derivation. In the planar case ($s = 0$), the ratio in (1.26) is a function of the Atwood number $A = (\rho_2 - \rho_1)/(\rho_2 + \rho_1)$, and can be determined exactly on the basis of a theory [7, 8] which has been validated against experiments. For the data (Atwood number) considered here, (1.26) has the value 1.5. The final choice of the d_k^q

	RT	RM
d_1^v	(1.26)	1.0
d_1^p	1.0	1.0
d_1^{pv}	1.0	1.0

Table 1.1: Summary of d_k^q parameter choices for RT and RM mixing. All choices except that for d_k^v in the RT case are insensitive.

to complete the model definition is specified in Table 1.1.

The hyperbolic character of the resulting model equations is easily derived, but this analysis reveals a missing internal boundary condition at each edge of the mixing zone. At the edges of the mixing zone, the hyperbolic analysis detects a missing incoming characteristic from the single fluid side. We supply this missing condition by coupling the model at the mixing zone edges to the buoyancy drag equation [10],

$$\ddot{Z}_k = (-1)^k Ag - (-1)^k C_k \frac{\rho_{k'}}{\rho_1 + \rho_2} \frac{V_k^2}{Z_k}. \quad (1.27)$$

Here C_k is a drag coefficient. This analysis is not needed for the present paper, in which we are analyzing closure terms relative to simulation data, but it is needed as part of a direct solution of the model equations, as has been discussed by Glimm et al. [25].

We see that the simple choice (1.26) for d_k^v depends on the buoyancy-drag equations for the mixing zone edge position Z_k . This equation has a free drag parameter which can be set by appeal to a bubble merger model [9], and to a model to set the ratio of bubble to spike growth rates [8]. This range of issues

have also been discussed for compressible mixing [33]. In fact, our study of highly compressible RT mixing [19] shows that the self similar scaling law for RT mixing (and thus the buoyancy drag equation, which allows a self similar solution) remain valid in the deeply compressible region. To achieve this result it is necessary to use time dependent Atwood numbers, reflecting the differing densities in a stratified highly compressible atmosphere. We consistently predict a strong increase in the mixing rate α with compressibility in this sense.

The Saurel et al. expressions,

$$d_k^{Sv} = Z_k/Z_{k'} \quad d_k^{Sp} = Z_{k'}/Z_k, \quad (1.28)$$

are derived from solutions of approximate (linearized) Riemann problems modeling multiphase flows at the sub-grid level; their closure does not satisfy (1.25). The Saurel et al. model supplies the missing internal boundary conditions at the edges of the mixing zone by imposition of equal pressures [45] $p_1(z = Z_k) = p_2(z = Z_k)$.

We have two interpretations of the Saurel et al. model. In the first, which we denote as Saurel-1, A_I is regarded as a fitting parameter. The second interpretation of their model is to take advantage of the fact that A_I is a computed quantity in our data, and to use this time dependent value in the definition of the model. We denote this model as Saurel-2. A_I is plotted as a function of time in Fig. 1.4, to complete the definition of the Saurel-2 closure. We note that A_I has the dimensions of an inverse length and takes on large

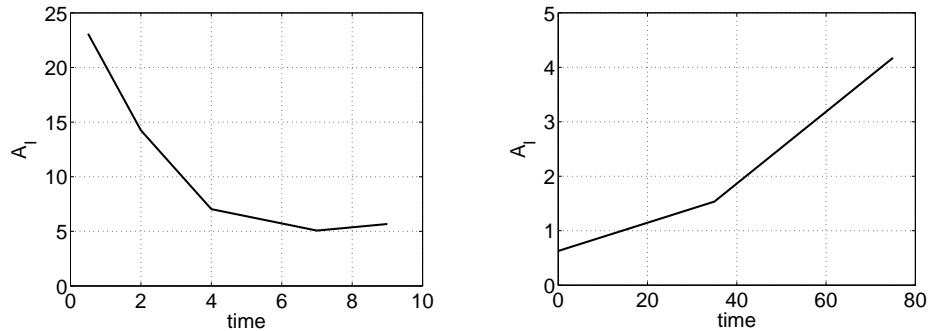


Figure 1.4: The interfacial area A_I per unit volume plotted vs. time. This plot serves to define the parameter A_I for the closure Saurel-2. Left: RT, Right: RM.

values at early time as the mixing layer tends to its (small) initial amplitude.

1.3.3 Validation and Comparison of Closures

The equations of this compressible, multi-species, multi-phase flow model, which account for surface tension and transport, were validated against multiple numerical simulation data sets. For 3D RT mixing, two classes of simulations were used: an ideal simulation of multi-mode 3D RT mixing with no surface tension, no mass diffusion, no viscosity (except for numerical viscosity) and no thermal conductivity, and a simulation similar to the ideal one except for added surface tension, in the form of dimensionless values that spanned the experimental range of dimensionless surface tensions found for the immiscible fluids (with no surfactants) in the experiments of Reed [43] and Smeeton-Youngs [48]. For 2D RM mixing, the data sets obtained in our previous work on uncertainty quantification [53] were used. Another simulation modeled

the Banershee-Andrews air-helium experiments with physical values of mass diffusion dimensionalized using an initial wave length scale inferred from observation [4].

In [4] we compare our closure, the two Saurel et al. closures and the RT or RM simulation data. In comparison to simulation data, we use the definition

$$\text{relative error} = \frac{1}{3} \sum_{v^*, p^*, (pv)^*} \frac{\int \int |\text{NS} - \text{model}| dz dt}{\int \int |\text{NS}| dz dt} \quad (1.29)$$

for the sum of the relative model errors for v^* , p^* and $(pv)^*$. The integration extends over the mixing zone. In the RT case, we exclude early time, before the bubbles have had much of a chance to interact, and to which the model is not supposed to apply, and integrate (*i.e.*, sum) over times from 4 to 10.

Our first main conclusion is the excellent (with about 10% error) agreement of our closures with the simulation data.

Our second main conclusion is the comparison of our closure to Saurel et al. We have previously introduced two distinct interpretations of the Saurel et al. model. In Fig. 1.5, we show the dependence of the total relative error on the value assumed for the area A_I . The error is minimized for $A_I = 0$ for both the RT data and the RM data, defining two different Saurel-1 closures for these two data sets. With that choice, we compare the total relative errors in our model and the two Saurel models, see Table 1.2. In summary, our model has errors about one quarter to one half the size of those for the Saurel et al model. This conclusion addresses the validation of the closure models by computation of the residuals for the closure terms within the validated

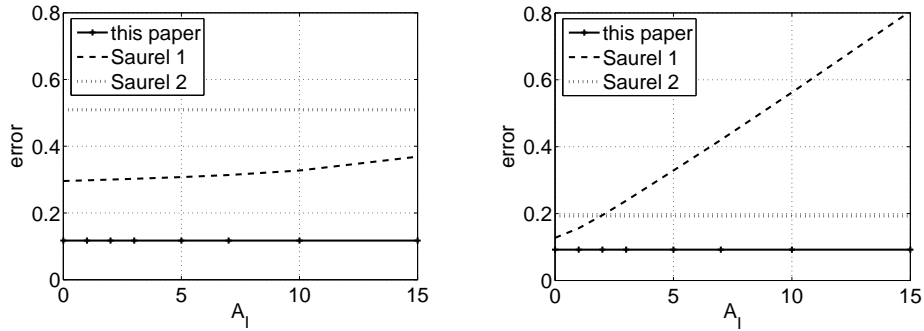


Figure 1.5: Comparison of the model error (1.29) for three closures. Of these, only Saurel-1 depends on the value of A_I as a fitting parameter; these plots serve to locate the best fit value of A_I ($A_I = 0$) and thus to define the Saurel-1 closure. Left: RT, Right: RM.

numerical simulations.

Our third main conclusion is the high degree of insensitivity of the q^* to the defining quantities μ_k^q , for many cases. We consider the condition

$$|\Delta q| \equiv |q_1 - q_2| \ll |\bar{q}| \equiv |\beta_1 q_1 + \beta_2 q_2| \quad q = v, p. \quad (1.30)$$

When (1.30) holds, q^* is effectively independent of the convex coefficients μ_k^q and thus of the value of q_k , as can be shown. This is the case for all RM data, $q^* = v^*, p^*$, and for p^* in the RT data. Moreover, $(pv)^*$ is insensitive to d_k^{pv} whenever (1.30) holds for $q = p$, which occurs for both RT and RM data, and in this case

$$(pv)^* \approx p^* v^*. \quad (1.31)$$

For these cases, the agreement of the two models and their agreement with the simulation data is understandable, but not a stringent test of the models.

Closure	v^*	p^*	$(pv)^*$	Average
Comparison to RT data				
This paper	18%	00%	18%	12%
Saurel-1	43%	02%	42%	30%
Saurel-2	56%	46%	51%	51%
Comparison to RM data				
This paper	07%	00%	20%	09%
Saurel-1	13%	04%	22%	13%
Saurel-2	12%	15%	31%	19%

Table 1.2: Model errors based on comparison to simulation data.

	v^*	p^*	$(pv)^*$
RT: Closure sensitive to d_k^q ((1.30) or (1.31) invalid)	yes	no	no
RT: Closure sensitive to v^*	–	–	yes
RT: Relaxation important	no	late time	late time
RM: Closure sensitive ((1.30) or (1.31) invalid)	no	no	no
RM: Relaxation important	no	early time	early time

Table 1.3: Summary properties related to the closures q^* . It is remarkable that the closures depend sensitively on their defining parameter d_k only in the case of the RT data for the v^* closure.

The insensitivity allows a simple choice of d_k^q , in cases where (1.30) holds. The only case where (1.30) is invalid is the v^* case for the RT instability. For the RT instability, v_1 and v_2 have generally opposite signs (in the frame stationary with the fluid container), so that $|\bar{v}|$ is small relative to $|\Delta v|$.

Our fourth main conclusion concerns the cases which violate (1.30), and thus for which the data is a stringent test of the models. This occurs for the v^* closure for the RT data. We find that our μ_k^v and those of Saurel et al. are significantly different.

The summary results of Table 1.3 can be understood as follows. The

sensitive case for (1.30) occurs for the RT v^* closure only. In this case, the computational frame is that of the average interface position. The light fluid moves away from the direction of g and the heavy fluid moves in the direction of g . Thus in most parts of the mixing zone, v_1 and v_2 has opposite signs, so that \bar{v} is nearly zero relative to Δv , or in other words, Δv is large. All other cases are insensitive, so that the closure in these cases is basically independent of the d_k^q . In (1.19), the Δv contribution to the relaxation terms is larger than the Δp contribution, and as Δv occurs in the p^* relaxation, the p^* relaxation terms are generally significant while the v^* relaxation terms (proportional to Δp) are not.

When Δp is small, which is the case for all simulations considered here, including those with surface tension, $(pv)^* \approx p^*v^*$ is approximately independent of d_k^{pv} , which is thus insensitive to d^{pv} . It is sensitive to v^* .

Our fifth main conclusion is the apparent insensitivity of the closures and averaged flow quantities to variation of (secondary) physics effects (ideal vs. surface tension vs. mass diffusion). These secondary mechanisms have a substantial influence on the growth rates of the mixing zone, i.e. α and the $Z_k(t)$. But once the influence of the edge motions has been scaled out of the data, the secondary physics appears to play only a small role in the simulation data, at least at length scales accessible at present levels of mesh refinement.

Chapter 2

Conservative Data Collection with Volume Fractions

The material in this chapter is based on the paper [4]. We develop the averaged surface area and volume fraction formulas to solve the numerical closure problem which is introduced in chapter 1. These formulas support the average $\langle X_k \rangle$ of the indicator function X_k which is denoted as β_k (chapter 1). The detailed formulas and proof are written in the technical report [34]. I led in developing these formulas in conservative data collection. It is a pleasure to thank my coauthors for their contributions to this joint work.

2.1 Introduction

The validation of the closures in the compressible flow models discussed in [4] was performed through the analysis of averaged quantities obtained from the numerical simulation (NS) data of 3D large-scale Rayleigh-Taylor simulations. This data was in turn collected and processed in a conservative manner.

What this means is that at each data collection time step a tracked interface was reconstructed as a grid-based interface for the sole purposes of data collection. For each grid cell traversed by the interface (*i.e.*, mixed cell), we then computed the volumes for each phase and the surface area of the interface separating these phases. Normalizing the volumes, we devised an algorithm that uses these fractional volumes in mixed cells to compute the phase cell averages for state variables associated with each phase. In particular, these phase cell averages are determined from the in-phase densities, or micro-densities, for mass, momentum, and energy, multiplied by the fractional volume fractions to obtain total phase mass, momentum, and energy. Other variables were constructed similarly as spatial volume-weighted averages. Here we present in detail the ideas behind the geometrical formulation for the simplified reconstructed interface and the computation of volume fractions and surface areas in mixed cells [34]. Our geometrical computations are based on the grid crossings of the interface with the cell edges and the 2-coloring of the cell corners, where each color is associated with a distinct phase.

Consider a 3D grid cell divided into two sub-domains. We compute the volume fraction of each of these domains and the surface area of the interface which separates them. This problem, as so formulated, is arbitrarily complex, but we assume a simple form of the two volumes and the surface area. This simple form is based on an interface reconstruction, which is determined by the crossings of the interface with the grid cell edges and the 2-coloring of the corners, where each color represents one of the two domain components. We derive from geometrical principles the fourteen non-trivial cases that, based on

our assumptions, give topologically distinct configurations for edge crossings. For each distinct case, we reconstruct the contact interface and discuss our procedure for computing volumes and surface areas. The construction is not unique; all possible constructions are obtained.

Finally, we document for the volume fraction and surface area formulas implemented in the hydrodynamic front tracking code *FronTier* .

Given a 3D grid cell divided by an interface, our aim is to determine:

1. The volume fractions of the two domains separated by the interface, respectively denoted by β_1 and β_2 . Since $\beta_1 + \beta_2 = 1$, we only compute the volume fraction β_1 .
2. The surface area A of the contact interface.

We now state our objective and our assumptions for framing this problem geometrically.

2.1.1 Objective

Our objective is to implement an accurate statistical routine for the collection of flow quantities in the *FronTier* package based on the volume and surface area formulas developed here. Conservative data collection using these formulas will aid in the data analysis for simulations of turbulent fluid mixing [4]. For the purpose of computing β_1 and A we reconstruct a simplified interface [24] within each cell, following the major steps given below:

1. Compute the crossings of the interface with the cell edges.

2. Determine component values at the cell corners and eliminate inconsistent crossings.
3. Reconstruct a new interface using consistent crossings.

To determine component values at the cell corners and identify inconsistent crossings for step 2, the algorithm relies on the crossings computed in step 1. At the end of step 2, topological inconsistencies due to interface self intersections have been removed and multiple interface crossings (more than one per edge) have been eliminated. We begin with an analysis of step 3, thus assuming that the interface has already been untangled.

The volumes and surface areas are based on a reconstructed interface that is determined by the set of crossing points and the domain components of the cell corners. It does not coincide with the original interface.

2.1.2 Assumptions

A 2-color scheme distinguishes the components of the cell corners. The corner 2-coloring is then used to locate edges which have crossings with the interface, namely those linking corners with different colors. Two objects, the *interface perimeter* and the *interface surface*, play a major part in the reconstruction step.

Consider a cell with n edge crossing points, each one belonging to a distinct edge.

Definition 3.1 *An **interface perimeter** is a non self-intersecting curve*

on the cell surface that is linear on each face, and that crosses the cell edges exactly at the n given edge crossing points.

Definition 3.2 An *interface surface* is a non self-intersecting mesh of triangular surface elements that has as lattice points the n edge crossing points within the cell, with none of the triangular elements (including their interior boundaries) lying on the cell surface.

The boundary of an interface surface is an interface perimeter.

In Section 2 we derive the fourteen non-trivial topologically distinct corner 2-colorings described in [24] for 3D cells. These cases, the starting point of our study, are the complete isomorphism invariants relative to the group of all isometries (distance-preserving transformations) and color inversions of the 2-colorings of the cell corners. In Section 3 we construct all possible interface perimeters consistent with a given set of edge crossings, and derive the complete set of triangulations for connected perimeters that may be realized as boundaries of interface surfaces in the cell. In Section 4 we discuss the procedure for averaging over the non-unique choices of perimeters and surfaces. In Section 5 we discuss the implementation of the volume fraction and surface area formulas in the front tracking code *FronTier*.

2.2 Corner Two-Colorings

Here we derive the fourteen non-trivial isomorphism classes for corner 2-colorings in 3D cells.

Proposition 3.1 *14 non-trivial, non-isomorphic cases give the complete list of topologically distinct configurations for edge crossings in 3D cells.*

Proof The isomorphism group G in question is generated by isometries of the cell and by color inversion. The proof is in two steps. First we enumerate 14 isomorphism classes of non-trivial corner 2-colorings. Then we determine the total number of non-equivalent and non-trivial corner 2-colorings according to Polyá's enumeration formula, following [24, 42, 50], to conclude that the enumerated list of 14 is complete.

Let C be the set of 2^d possible corner 2-colorings within a cell, where d is the number of cell corners. For a 3D cell, $d = 8$, so there is a total of $2^8 = 256$ possible corner 2-colorings. Two corner 2-colorings are *isomorphic* if there exists a $g \in G$ that maps one corner 2-coloring onto the other. Then G induces a partition of C into isomorphism classes, the disjoint subsets of C which group together all 2-colorings that are isomorphic to one another.

We enumerate the following 14 isomorphism classes for corner 2-colorings according to the number of black corners in the cell (where the case with zero black corners is discarded as trivial):

1. One black corner.

Two black corners such that:

2. they are connected by an edge.
3. they share a common face but are not connected by an edge.
4. they do not share a common face.

Three black corners such that:

5. they share a common face (*i.e.*, they are pairwise connected by two edges).
6. two are connected by an edge and the other does not share a common face with that edge.
7. none of the three are connected by an edge.

Four black corners such that:

8. they share a common face.
9. they are pairwise connected by three edges and form a broken line whose successive increments define a right-handed coordinate system.
10. they are pairwise connected by three edges and form a broken line whose successive increments define a left-handed coordinate system.
11. they are pairwise connected by three edges but do not share a common face of the cube nor form a broken line.
12. three are pairwise connected by two edges and the other is not connected to either of those edges.
13. two pairs are each connected by an edge, but they do not share a common face.
14. none of the four are connected by an edge.

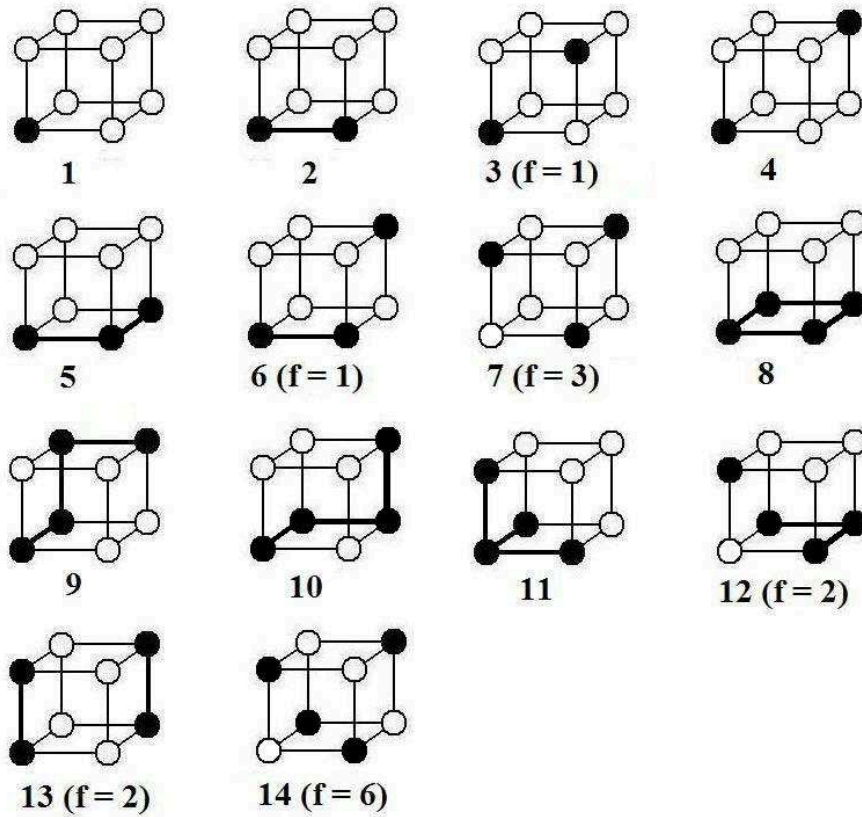


Figure 2.1: 14 topologically distinct, or non-isomorphic, corner 2-colorings in 3D cells. The numbering follows the enumeration of isomorphism classes. For later use we give f , the number of faces with opposite black corners, in the corner 2-colorings for which $f > 0$.

From their description, each of these classes is invariant under G .

In Fig. 2.1 we show the 14 non-isomorphic and non-trivial corner 2-colorings which follow the list of isomorphism classes given above. We also give f , the number of faces with opposite black corners and opposite white corners (*i.e.*, with four edge crossing points), in the 2-colorings for which $f > 0$. This number plays a crucial role in the construction of interface perimeters, the topic of the next section.

To complete the proof, we show by abstract methods that the number of isomorphism classes is 14, so that each case identified above must be a single isomorphism class, and the list of 14 is a complete list of isomorphism classes. We follow the derivation in [50] (Example 4, Section 9.4) that applies Polya's enumeration formula [42] to the non-equivalent corner 2-colorings of a cube.

First we show that a floating cube has 24 distinct *symmetries* involving revolutions about axes passing through the centers of opposite faces, the midpoints of diagonally opposite edges on opposite faces, and diagonally opposite corners on opposite faces. These symmetries are the complete set of isometries I that map a fixed cube onto itself. We have, for each of the three distinct pairs of opposite faces in the cube, either 90° , 180° , or 270° revolutions that yield symmetries. With the trivial identity symmetry (the 0° revolution), that makes a total number of 10 symmetries. For the six distinct pairs of diagonally opposite edges on opposite faces, only the 180° revolution yields a symmetry. Thus, the total number of symmetries is increased to 16. Finally, for the four distinct pairs of diagonally opposite corners on opposite faces, the 120° and 240° revolutions yield symmetries. This brings the total number of symmetries

to $16 + 8 = 24$.

Next we determine the *pattern inventory* for the corner 2-colorings of a floating cube. The pattern inventory, denoted by $P_I(c_1, c_2, \dots, c_n)$, is a generating function that gives the total number of colorings of an unoriented figure using different possible collection of colors c_1, c_2, \dots, c_n . As an instance, the pattern inventory for the 2-coloring of a cell face using a black and white color scheme is given by $P_I(b, w) = 1b^4 + 1b^3w + 2b^2w^2 + 1bw^3 + 1w^4$, where the term $1b^3w$, for example, gives the number of non-equivalent colorings of the face with three black corners and one white corner. The pattern inventory is determined by summing the *cycle structure representations* of all the symmetries written in terms of the colors c_1, c_2, \dots, c_n and dividing this sum by the number of total symmetries. Thus, we first seek to find the cycle structure representation of the 24 symmetries we have identified. A symmetry is naturally characterized by a given permutation π_i of the eight corners of the cube. This permutation can be, in turn, represented as a product of disjoint cyclic permutations, or *cycles*. For example, the symmetry that involves a 180° revolution in the case of a cell face with cycling corners a, b, c, d is characterized by the permutation $\pi = (ac)(bd)$ which permutes corners a and c to corners c and a . The cycle structure representation of this symmetry is then given by $x_2x_2 = x_2^2$, where x_2 is the length of each 2-cycle. In general the cycle structure representation of a symmetry characterized by the permutation π_i is the product of the lengths x_i of the disjoint i -cycles that represent the permutation. Looking at the permutations of the corners resulting from the revolutions that yield symmetries in our case of a floating cube, we ob-

tain the cycle structure representation for each of the 24 symmetries. The identity symmetry leaves all corners unchanged and, thus, has cycle structure representation $x_1x_1x_1x_1x_1x_1x_1x_1 = x_1^8$. For opposite faces, a 90° or a 270° revolution permutes all corners belonging to each opposite face so that the cycle structure representation of these symmetries is $x_4x_4 = x_4^2$. The 180° revolution permutes only opposite corners on each opposite face and so the cycle structure representation is $x_2x_2x_2x_2 = x_2^4$. Since there are three distinct pairs of opposite faces in a cube, the total contribution of opposite-face revolutions is $6x_4^2 + 3x_2^4$. Now for diagonally opposite edges on opposite faces, the 180° revolution permutes pairs of distinct corners so that the cycle structure representation is $x_2x_2x_2x_2 = x_2^4$. Since there are six pairs of distinct opposite edges in a cube, the total contribution of opposite-edge revolutions is $6x_2^4$. Finally, for opposite corners the 120° and 240° revolutions leave both opposite corners fixed while cyclically permuting the three corners adjacent to them. The cycle structure representation for these symmetries is then $x_1x_3x_3x_1 = x_1^2x_3^2$, and the total contribution of opposite-corner revolutions is $8x_2^4$ since there are four pairs of distinct opposite corners in a cube.

Collecting terms we obtain the generating function that gives the pattern inventory for the corner 2-colorings of a floating cube. Namely,

$$P_I(x_1, x_2, x_3, x_4) = \frac{1}{24} (x_1^8 + 6x_4^2 + 9x_2^4 + 8x_1^2x_3^2) . \quad (2.1)$$

For corner 2-colorings of the cube using a black and white scheme we use the algebraic identity $x_i = (b^i + w^i)$ given in [50]. This then gives the pattern inventory

$$P_I(b, w) = \frac{1}{24} \left((b+w)^8 + 6(b^4+w^4)^2 + 9(b^2+w^2)^4 + 8(b+w)^2(b^3+w^3)^2 \right) . \quad (2.2)$$

Expanding Eq. 2.2 we get

$$P_I(b, w) = (b^8 + b^7w + 3b^6w^2 + 3b^5w^3 + 7b^4w^4 + 3b^3w^5 + 3b^2w^6 + bw^7 + w^8) . \quad (2.3)$$

In Eq. 2.3 we drop terms that are redundant. Thus, corner 2-colorings with more than five black corners (b^i terms with $i > 5$) are dropped since these can be mapped to colorings with less than five corners after a color interchange (white to black, black to white). In addition, the terms w^8 and b^8 are dropped since 3D cells cut by an interface have, by definition, at least two corners with different domain components (*i.e.*, different colors). This ultimately yields the number of non-trivial 2-colorings of a floating cube, namely

$$P_I(1, 1) = 1 + 3 + 3 + 7 = 14 . \quad (2.4)$$

From this result we conclude that the 14 corner 2-colorings shown in Fig. 2.1 constitute a complete list of non-trivial isomorphism classes for corner 2-colorings in 3D cells. \square

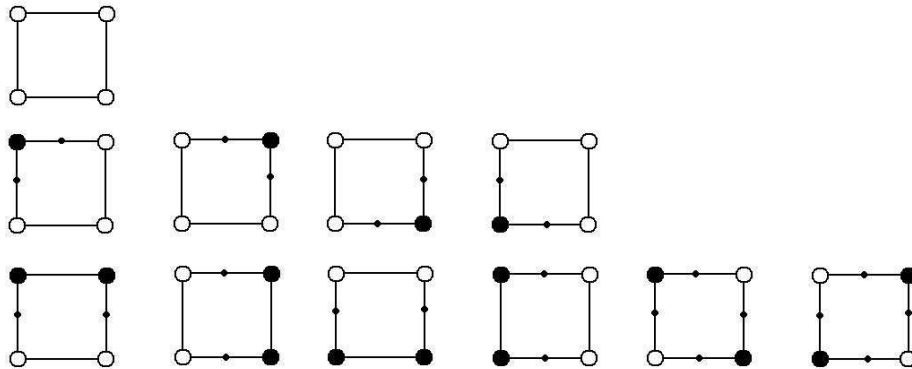


Figure 2.2: Non-isomorphic corner 2-colorings for single cell faces. From top to bottom row, the corner 2-colorings have zero, one, or two black corners. Edge crossings are shown as small black dots.

2.3 Interface Perimeters

2.3.1 Perimeter Construction

Here we construct all the possible perimeter curves that can be formed on the surface of the 3D cell, following Definition 3.1, for each isomorphism class of corner 2-colorings. Our approach is to start from a single cell face and then proceed to the union of the cell faces.

Proposition 3.2 *There are either zero, two, or four edge crossings on any cell face.*

Proof A cell face is a square with four edges and four corners. A complete circuit around the square observes an even number of crossings, since the starting and ending colors are identical. In the case that all corners are of a



Figure 2.3: Interface perimeters on single cell faces. They are uniquely determined by the edge crossings for cell faces with two edge crossings (left) and they are two-fold non-unique for cell faces with four edge crossings (right). Edge crossings are shown as small black dots.

single color, there are no crossings. Moreover, there is at most one crossing per edge, so the maximum number of crossings is 4. We show all possible cases in Fig. 2.2. \square

Proposition 3.3 *The interface perimeter on a given cell face is uniquely determined by the edge crossings for the cases of zero or two edge crossings. It is two-fold non-unique in the case of four edge crossings.*

Proof Faces with two edge crossings have a unique interface perimeter, namely the linear segment connecting the crossings (see Fig. 2.3, left). For cell faces with four edge crossings, the interface perimeter is, by definition, the union of two linear segments connecting distinct pairs of edge crossings. Since these segments are not allowed to intersect, they must join crossings on adjacent edges. There are exactly two ways to do this (see Fig. 2.3, right). \square

To construct interface perimeters on the union of the cell faces, we proceed by locating all edge crossings in each corner 2-coloring of Fig. 2.1 and then forming the perimeters on each face. Since the interface perimeter crosses the

cell edges exactly at the edge crossing points, it is formed by selecting one of the possible perimeters for each single cell face. In general, the interface perimeter is not connected but is the union of multiple disjoint connected components.

Proposition 3.4 *There are 92 distinct interface perimeters that can be formed from the fourteen non-isomorphic, non-trivial, configurations of edge crossings in 3D cells.*

Proof This is the extension in 3D of Proposition 3.3. Non-uniqueness on a single cell face results in non-uniqueness for the interface perimeter. According to Proposition 3.3, this only occurs in cases where a cell face has four edge crossings (*i.e.*, opposite black corners and opposite white corners). For a given corner 2-coloring, the number P of possible interface perimeters is given by $P = 2^f$, where f is the number of faces with four edge crossings. For $f = 0$, $P = 1$ in cases 1, 2, 4, 5, 8, 9, 10, and 11. The remaining corner 2-colorings have $f = 1$ (cases 3 and 6), $f = 2$ (cases 12 and 13), $f = 3$ (case 7), and $f = 6$ (case 14) faces with four edge crossings resulting in, respectively, $P = 2^1 = 2$, $P = 2^2 = 4$, $P = 2^3 = 8$, and $P = 2^6 = 64$ possible distinct perimeter constructions. Summing these possibilities, we have $8(2^0) + 2(2^1) + 2(2^2) + 2^3 + 2^6 = 92$ possible interface perimeters in 3D cells. \square

In general, interface perimeters are disconnected and consist of several connected components. We present in Figs. 2.4, 2.5, and 2.6 the 92 interface perimeters, organized according to their number of connected components. Fig. 2.4 shows all the connected perimeters, while Figs. 2.5 and 2.6 show the








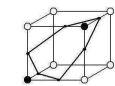
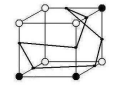



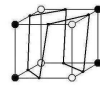



















f = 0	 1 - Corner	 2 - Edge	 5 - Glider	 8 - Plane		
	 9 - Twister-I	 10 - Twister-II	 11 - Hexagon			
f = 1	 3 - Hexagonoid	 6 - Twisteroid				
f = 2	 12 - Octagonoid-I	 13 - Octagonoid-II				
f = 3	 7 - Enneagonoid					
f = 6	 14 - Duodecagonoid-I					
						
						
	14 - Duodecagonoid-II					

Figure 2.4: 32 distinct connected interface perimeters that can be formed on the surface of a 3D cell. We group the perimeters according to f , the number of faces with four edge crossings, and the isomorphism classes of perimeters in each non-isomorphic case. Note that we have labeled these perimeter classes and that, in most cases, labels follow the names of polyhedra given by the number of edge crossings on the perimeter. The numbering of the non-isomorphic cases follows the enumeration of isomorphism classes for corner 2-colorings.

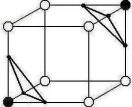
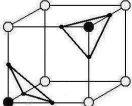
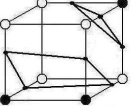
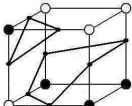
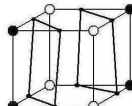
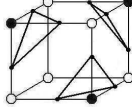
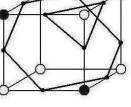
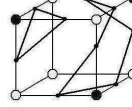
f = 0	 4 - Two Corners			
f = 1	 3 - Two Corners		 6 - Edge + Corner	
f = 2	 12 - Glider + Corner		 13 - Two Edges	
f = 3	 7 - Three Corners		 7 - Hexagon + Corner	
	 7 - Hexagonoid + Corner			

Figure 2.5: 12 distinct disconnected interface perimeters that can be formed on a 3D cell surface, for $f = 0, 1, 2, 3$. The numbering of the non-isomorphic cases follows the enumeration of isomorphism classes for corner 2-colorings.

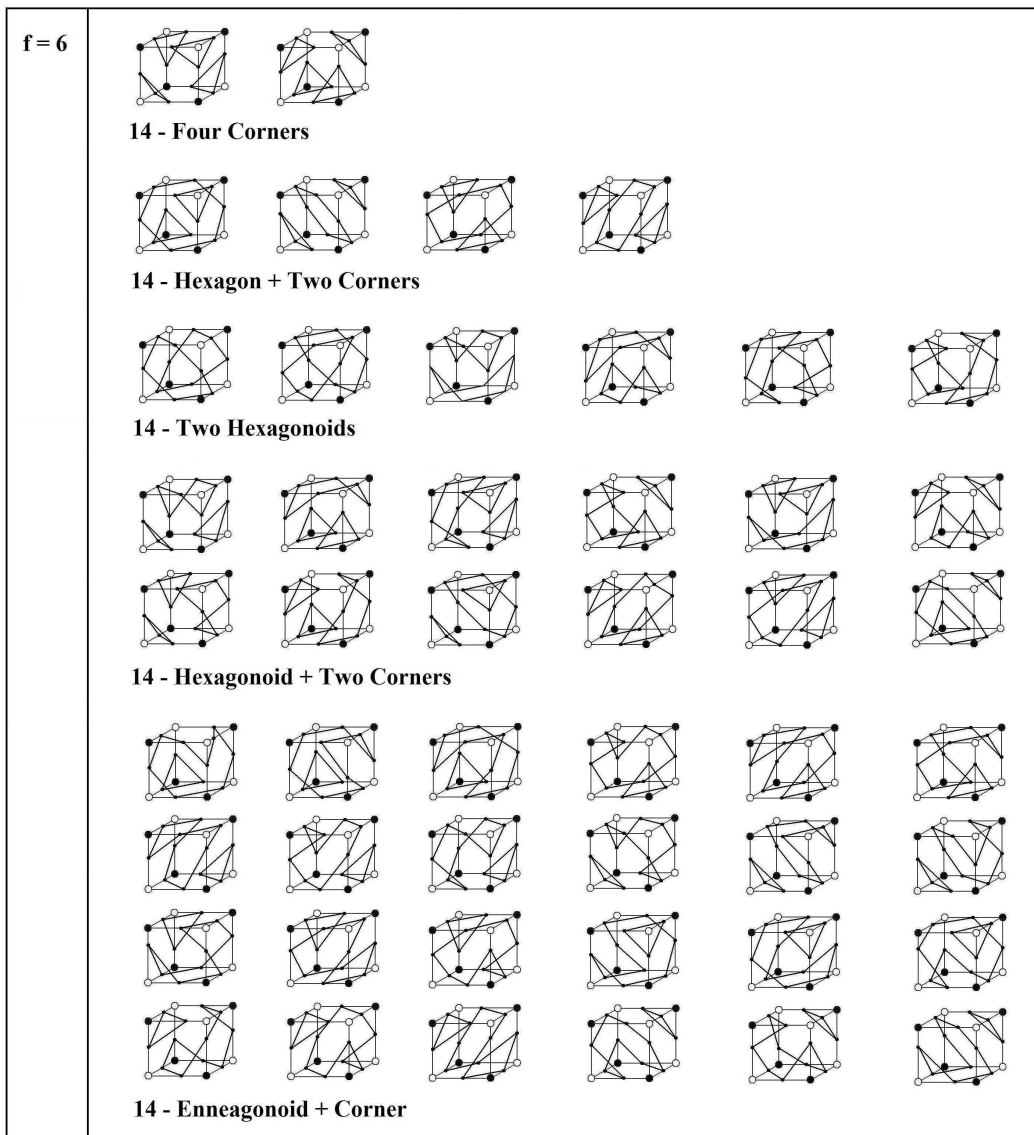


Figure 2.6: 48 distinct disconnected interface perimeters that can be formed on a 3D cell surface in the non-isomorphic case with $f = 6$ (case 14).

disconnected ones. For the non-isomorphic case 14, with $f = 6$, there is a total of $P = 2^6 = 64$ distinct interface perimeters, including the 48 disconnected perimeters shown in Fig. 2.6. By counting the distinct interface perimeters seen in all three figures, we have a total number of $32 + 12 + 48 = 92$ possible perimeter constructions. From Proposition 3.4, we conclude that Figs. 2.4, 2.5, and 2.6 show all the distinct interface perimeters that can be formed on the surface of a 3D cell. A disconnected perimeter has at most four connected components; the “maximal case” occurs when there is an edge crossing on each cell edge. As before, we consider isomorphism classes for connected perimeters relative to the group of isometries and color inversion.

Proposition 3.5 *The 32 connected perimeters displayed in Fig. 2.4 are grouped (by their common labels) into 14 isomorphism classes.*

Proof We show that each of the fourteen classes of connected perimeters depicted in Fig. 2.4 form an isomorphism class. For cases with $f = 0, 1$, the isomorphism classes of perimeters consist of exactly one element distinguished by an isomorphism class of the corner 2-colorings. In the two cases with $f = 2$ (12 and 13), the isomorphism is a 180° rotation about the line passing through the midpoints of the front-left and back-right vertical edges (diagonally opposed in the cell), followed by a color inversion, for the octagonoid-I perimeter class, and the same “up-down” rotation about the opposite edges with black corners for the octagonoid-II perimeter class. For the enneagonoid perimeter class of case 7, with $f = 3$, the three symmetries are the 0° , 120° , and 240° rotations about the line passing through the top-right front and the bottom-left

back corners (diagonally opposed corners in the cell). For the duodecagonoid-I perimeter class of case 14 with $f = 6$, the four symmetries consist of 0° and 180° rotations about the line passing through the center points of the top and bottom cell faces, and of 90° and 270° rotations about the same axis followed by a color inversion. In the same case, the duodecagonoid-II perimeter class has twelve symmetries given by the same rotations as in the duodecagonoid-II and the enneagonoid perimeter classes, namely 0° and 180° rotations about the line passing through the center points of the top and bottom cell faces, 90° and 270° rotations about the same axis followed by a color inversion, and 0° , 120° , and 240° rotations about the line passing through the top-right front and the bottom-left back corners (diagonally opposed corners in the cell). For each rotation about the axis passing through the diagonally opposed corners, there are four symmetries generated by the rotations about the vertical axis passing through the top and bottom faces. This brings the total number of symmetries to $4 \times 3 = 12$. The two isomorphism classes of perimeters in case 14 have topologically distinct perimeters distinguished by an invariant cyclic ordering around the perimeter of the edges lying on opposite faces of the cell. We return to this topological invariant in the next section, where we discuss the triangulation of connected perimeters, and complete the proof. \square

2.3.2 Valid Triangulations of Connected Perimeters

We now turn to valid triangulations of connected perimeters, as these determine the interface surface. We first recall some results on the triangulation

of polygons and then define validity for triangulations of interface perimeters.

Preliminaries on the Triangulation of Polygons

We begin by recalling some elementary definitions [41].

Definition 3.3 A **graph isomorphism** f is a bijection, or one-to-one mapping, between the vertices of two graphs G and H ($f : V(G) \rightarrow V(H)$) with the property that any two vertices u and v from G are adjacent if and only if $f(u)$ and $f(v)$ are adjacent in H . If there exists such a mapping between two graphs, then they are said to be isomorphic.

Connected perimeters are then clearly graph isomorphic to polygons.

Definition 3.4 Any polygon P can be **triangulated** by connecting pairs of non-adjacent vertices with straight lines called diagonals, so that every vertex is the endpoint of at least one diagonal.

This leads to the following result.

Theorem 3.1 If P is a polygon with n sides, then there is a total of $n - 3$ diagonals which will decompose P into $n - 2$ triangles.

Proof We use a straightforward induction proof taken from O'Rourke [41]. We start with $n = 3$ (the trivial case of a triangle). For $n \geq 4$, we partition the polygon P into two sub-polygons P_1 and P_2 so that $P_1 \cup P_2 = P$ and the diagonal d is adjacent to both P_1 and P_2 . These polygons have, respectively, n_1 and n_2 vertices. Clearly, we have a vertex count of $n_1 + n_2 = n + 2$ since the endpoints of d are counted twice: first in the vertices of P_1 , then

in the vertices of P_2 . Applying the induction hypothesis to both sub-polygons and using the identity $n_1 + n_2 - 2 = n$, we get a diagonal count for P of $(n_1 - 3) + (n_2 - 3) + 1$, where the $+1$ represents the diagonal d and $n_1 - 3$ and $n_2 - 3$ represent, respectively, the diagonals of P_1 and P_2 . We thus have a total of $(n_1 - 3) + (n_2 - 3) + 1 = (n_1 + n_2) - 5 = n + 2 - 5 = n - 3$ diagonals in P and $(n_1 - 2) + (n_2 - 2) = n + 2 - 4 = n - 2$ triangles. This completes the proof. \square

A natural question that arises from Theorem 3.1 is how to identify the number of ways that an n -gon can be divided into $n - 2$ triangles if different placements of the $n - 3$ diagonals are counted separately. This is Euler's famous *polygon division problem*. It is well-known in combinatorics that the answer to this question is given by a sequence of natural numbers, denoted by $\{C_n\}$, called the *Catalan numbers*.

Theorem 3.2 *The answer to Euler's polygon division problem is the Catalan number C_{n-2} , where the n th term of the sequence $\{C_n\}$ is given by*

$$C_n = \frac{1}{n+1} \binom{2n}{n} = \frac{(2n)!}{(n+1)!n!}. \quad (2.5)$$

Proof We do not prove this result, as it is beyond the scope of our discussion, but we refer the reader to an elegant and succinct geometrical proof by Lamé [35]. \square

The first few Catalan numbers are

$$\{C_n\} = \{1, 1, 2, 5, 14, 42, 132, \dots\} , \quad (2.6)$$

for $n = 0, 1, 2, 3, \dots$

The formula in Eq. 2.5 is derived according to [30] using the recurrence relation

$$C_0 = 1, C_{n+1} = \sum_{i=0}^n C_i C_{n-i} , \quad (2.7)$$

for $n \geq 0$, that is satisfied by the sequence $\{C_n\}$. Then, defining a generating function of the form

$$c(x) = \sum_{n=0}^{\infty} C_n x^n , \quad (2.8)$$

and using Eq. 2.7 we have $c(x) = 1 + xc(x)^2$ and, hence, $c(x) = \frac{1-\sqrt{1-4x}}{2x}$.

Expanding the square root term as a power series using the identity

$$\sqrt{1+y} = 1 - 2 \sum_{n=1}^{\infty} \binom{2n-2}{n-1} \left(\frac{-1}{4}\right)^n \frac{y^n}{n} , \quad (2.9)$$

and substituting the result for $c(x)$ gives

$$c(x) = \sum_{n=0}^{\infty} \binom{2n}{n} \frac{x^n}{n+1} . \quad (2.10)$$

Finally, equating coefficients yields Eq. 2.5.

From Eq. 2.7 we can immediately infer the number of distinct triangulations for any polygon, including those that are graph isomorphic to connected

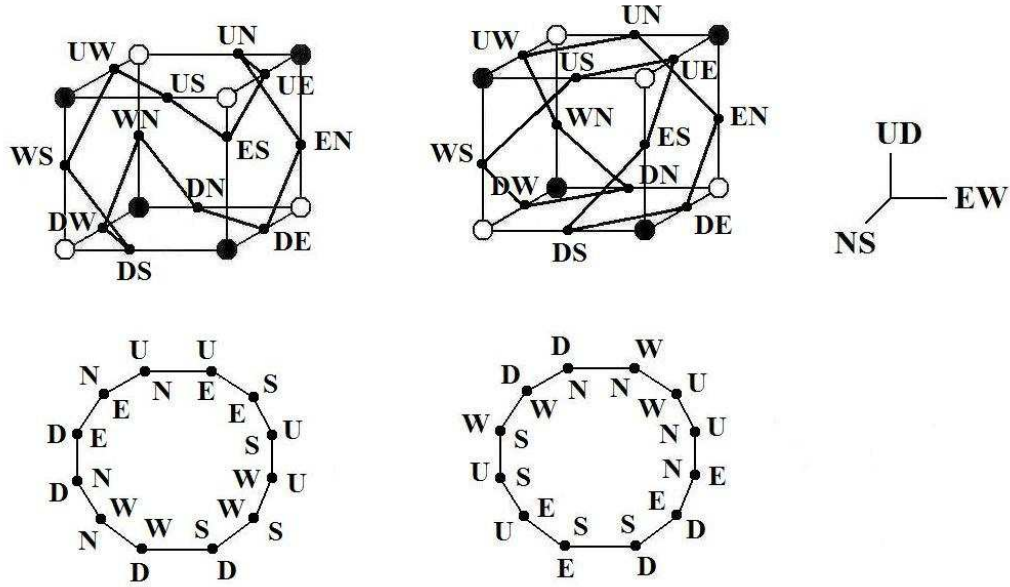


Figure 2.7: Examples of perimeters in the duodecagonoid-I (left) and duodecagonoid-II (right) isomorphism classes of connected perimeters of case 14, with $f = 6$.

perimeters (the boundaries of interface surfaces in 3D cells).

We are now in a position to complete the proof of proposition 3.4. We seek to define a topological invariant that distinguishes perimeters in the duodecagonoid-I isomorphism class from elements in the duodecagonoid-II isomorphism class. In Fig. 2.7 we have labeled edge crossings belonging to a common face according to the orientation of that face in the three spacial directions (NS, EW, or UD). We see from the two duodecagons in Fig. 2.7 that the relative location of the edges belonging to the three sets of opposite faces in the cell (*e.g.*, N-S) around the perimeter constitutes a topological invariant. Furthermore, this invariant is distinct for the two perimeter classes. For ele-

ments in the duodecagonoid-I case, edges on opposite faces are also opposite in the polygon. So for edge crossings on opposite faces, denoted by f_1 and f_2 , where $f_1 = U, E, N$ and $f_2 = D, W, S$ (according to which face they belong to in the cell), the cyclic representation of their position around the duodecagon is given by $\pi_I = (f_1, f_1, a, f_1, f_1, b, f_2, f_2, c, f_2, f_2, d)$, where a, b, c, d denote other edge crossings. For elements in the duodecagonoid-II case, the corresponding cyclic representation is given by $\pi_{II} = (f_1, f_1, a, f_2, f_2, b, f_1, f_1, c, f_2, f_2, d)$ for one of the three sets of edges on opposite faces, and by π_I for the other two sets of edges on opposite faces. The invariant for the duodecagonoid-I isomorphism class, which is distinguished by the cyclic representation π_I , has then four symmetries, while the invariant for the duodecagonoid-II isomorphism class, which is distinguished by the cyclic representation π_I for one set of edges on opposite faces and by the cyclic representation π_{II} for the other two sets of edges on opposite faces, then has $3 \times 4 = 12$ symmetries. Elements in both isomorphic classes are shown in Fig. 2.4.

Validity

Many of the connected perimeters that we have constructed previously cannot be realized as the boundary of a (valid) interface surface according to our assumptions. The restriction from Definition 3.2, that no diagonal from a triangulation lie on the cell surface, is in fact a constraint for any case in which at least one cell face has four edge crossings (*i.e.*, $f > 0$). For all cases with $f = 0$, a connected perimeter P with n edge crossings is, according to Theorem 3.2, the boundary of C_{n-2} distinct interface surfaces, each one

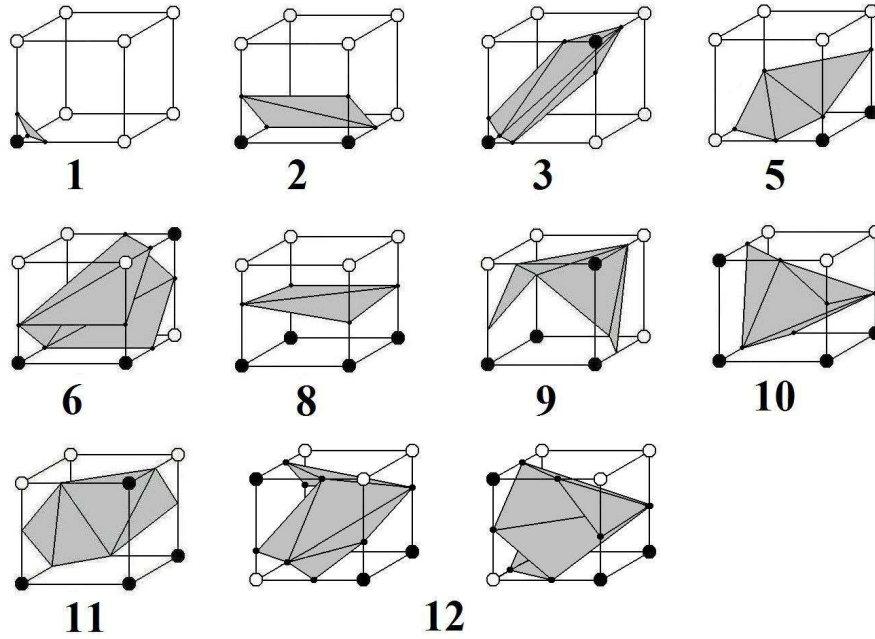


Figure 2.8: Examples of interface surfaces bounded by valid perimeters belonging to isomorphism classes of connected perimeters with $T > 0$. They are, from left to right, starting at the top, the corner (case 1), the edge (case 2), the hexagonoid (case 3), the glider (case 5), the twisteroid (case 6), the plane (case 8), the twister-I (case 9), the twister-II (case 10), the hexagon (case 11), and the octagonoid-I (case 12). Note that the perimeter class of case 12 has two distinct elements. All triangulations are picked arbitrarily.

determined by a distinct triangulation of the n -gon. As f increases, the number of prohibited diagonals increases, and in some cases no such triangulation is possible. Consider, for example, the connected perimeter shown in case 13 of Fig. 2.4, with $f = 2$. The octagonoid surface bounded by this perimeter cannot lie totally in the cell interior. This is an example of an *invalid* interface surface, according to Definition 3.2. The connected perimeter P with n edge crossings is called *valid* if there is at least one interface surface S having P as its boundary. The definition will be satisfied if there is a triangulation of the n -gon for which no diagonals lie on a face (such a triangulation is also called *valid*). General (disconnected) perimeters are valid if each of their connected components are valid.

Let T be the number of valid triangulations for the elements in a given isomorphism class of connected perimeters. Fig. 2.8 shows examples of interface surfaces bounded by valid connected perimeters belonging to perimeter classes with $T > 0$.

Proposition 3.6 *The values of T in Fig. 2.9 are correct.*

Proof The proof of this proposition is exposed in [34]. \square

2.4 Computational Procedure

Our computation of volumes and surface areas is based on the configuration of all interface surfaces that may be realized inside the cell. Thus, for connected perimeters classes with $f > 0$ we only consider valid triangulations.




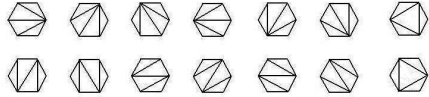
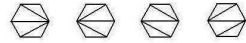
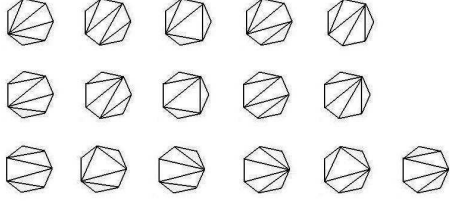
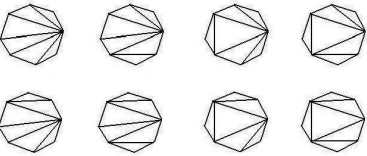
Perimeter Class	f	n	C_{n-2}	T	Valid Triangulations
Corner	0	3	1	1	
Edge / Plane	0	4	2	2	
Glider	0	5	5	5	
Hexagon / Twister-I / Twister-II	0	6	14	14	
Hexagonoid	1	6	14	4	
Twisteroid	1	7	42	16	
Octagonoid-I	2	8	132	8	
Octagonoid-II	2	8	132	0	
Enneagonoid	3	9	429	0	
Duodecagonoid-I	6	12	16796	0	
Duodecagonoid-II	6	12	16796	294	Not shown

Figure 2.9: Valid triangulations for elements in the isomorphism classes of connected perimeters. In the table, we give for each class f , the number of faces with four edge crossings on a common face; n , the number of edge crossings on the perimeter; C_{n-2} , the Catalan number; and T , the number of valid triangulations. Only triangulations for perimeter classes with $f = 0$ were used in our implementation. In these cases, the identity $T = C_{n-2}$ holds; otherwise, $T < C_{n-2}$.

A reconstructed interface is determined first by the interface perimeter P , and then by the interface surface S it bounds according to a given triangulation of $P = \partial S$. For each non-isomorphic case, we wish to compute a volume fraction and a surface area that takes into account all the valid triangulations for all the perimeters that can be constructed in that case.

Our procedure is then an averaging over a set of geometrical possibilities. For cases with a unique interface perimeter (with $f = 0$) this procedure only involves an averaging over the distinct interface surfaces and volumes given by the triangulations of the perimeter. For the remaining cases, an averaging over distinct interface perimeters is also required.

The surfaces bounded by elements in the hexagonoid, twisteroid, octagonoid-I, and duodecagonoid-I perimeter classes are not used in the construction of the areas and volumes. Geometrical constructions (not shown here) seem to show that the duodecagonoid surfaces are typically or always self-intersecting inside the 3D cell, effectively violating Definition 3.2. For this reason, and due to the complexity of these surfaces (which intuitively do not appear to be physically realistic), such terms have been omitted from the formulas for areas and volumes presented below. In the hexagonoid, twisteroid, and octagonoid-I cases, constructions seem to show that the self-intersection of surfaces does not occur. However, this topological property is not formally proved here. We also omit these terms due to their complexity.

If the computational grid spacings in the x , y and z directions are, respectively, Δx , Δy , and Δz , then we divide the fractional cell volume by the unit cell volume $V_0 = \Delta x \Delta y \Delta z$ to map V_1 , the volume of domain 1 inside

the grid cell, into a volume fraction $\beta_1 = \frac{V_1}{V_0}$, so that $0 \leq \beta_1 \leq 1$. We do not normalize A , as there is no convenient method to do so.

2.4.1 Volume Fractions

For any of the fourteen non-isomorphic cases there is either a unique interface perimeter or many distinct interface perimeters. Let P denote the number of interface perimeters that can be formed for a given case and that are implemented in *FronTier* (*i.e.*, that exclude hexagonoid, twisteroid and octagonoid-I components). Moreover, any given interface perimeter consists of at most four connected components. Let $S \in \{1, 2, 3, 4\}$ denote the number of connected components of a given interface perimeter. As before, let T denote the number of valid triangulations for a given connected perimeter. Then $T = C_{n_j-2}$ for all connected components considered in our computational procedure. Finally, let $\beta_{1,i,j}^k$, where $i = 1, \dots, C_{n_j-2}$, $j = 1, \dots, S$, and $k = 1, \dots, P$, denote the volume fraction of domain 1 given a fixed triangulation i of a connected component j , with n_j edge crossings, of an interface perimeter k . We arbitrarily pick domain 1 to be the domain with black corner components.

For a fixed triangulation i of the connected perimeter component j belonging to the interface perimeter k , we must compute $\beta_{1,i,j}^k$, the volume fraction occupied by domain 1. To do so we recall two basic results from analytic geometry.

1. The volume of a *parallelepiped* spanned by the three vectors $\mathbf{a} = (a_1, a_2, a_3)$, $\mathbf{b} = (b_1, b_2, b_3)$, and $\mathbf{c} = (c_1, c_2, c_3)$ is given by:

$$V = |\det(\mathbf{a}, \mathbf{b}, \mathbf{c})| = \left| \det \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \right| \quad (2.11)$$

2. The volume of a *tetrahedron* (i.e. a triangular pyramid) with vertices \mathbf{a} , \mathbf{b} , \mathbf{c} , and \mathbf{d} is given by:

$$V = \frac{1}{6} |\det(\mathbf{a} - \mathbf{b}, \mathbf{b} - \mathbf{c}, \mathbf{c} - \mathbf{d})| \quad (2.12)$$

For each valid triangulation of the interface perimeter there is a set of tetrahedra that decompose the space occupied by domain 1 inside the cell. Thus, given the set of crossings and the black corners in the corner 2-coloring, we can compute $\beta_{1,i,j}^k$ using Eq. 2.12. For the hexagon case, we use Eq. 2.11 in addition to Eq. 2.12. To do so, the crossings and the corners are treated as vectors in 3D Cartesian space.

Fig. 2.10 illustrates, for a given triangulation of the glider perimeter in case 5, how the space occupied by domain 1 is defined by the interface surface and can be decomposed into five tetrahedra. We arbitrarily pick this particular triangulation to demonstrate our procedure.

To compute V_1 , the total volume of the space occupied by domain 1 in Fig. 2.10, we compute the sum of five tetrahedron volumes. These five tetrahedra have vertices selected from the five edge crossings on the interface perimeter, labeled \mathbf{p}_i , where $i = 1, \dots, 5$, and the three black corners in the corner 2-coloring of the cell labeled A , B and C . They all share the corner

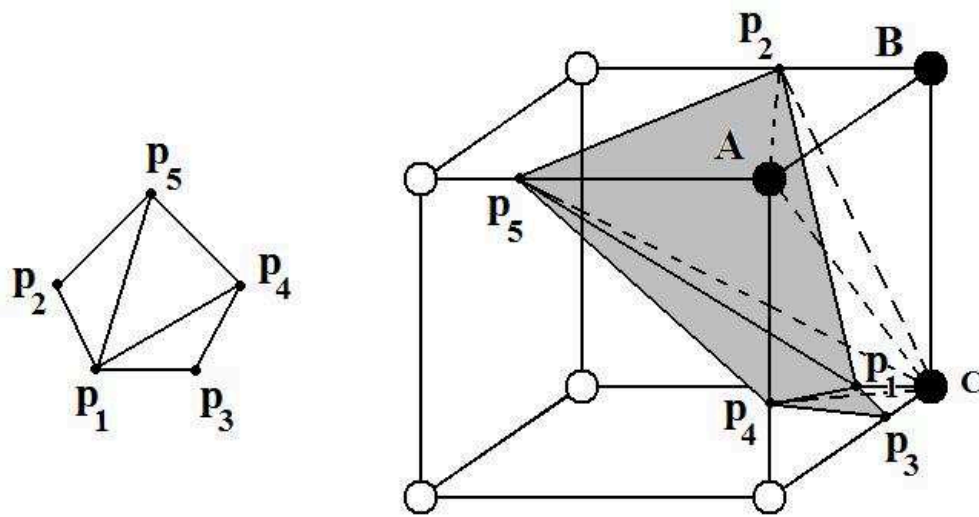


Figure 2.10: Decomposition of the space occupied by domain 1 into five tetrahedra in one of five possible triangulations in the glider case (case 5). The crossings are the vertices \mathbf{p}_i , where $i = 1, \dots, 5$. The black corners are labeled A , B and C .

vertex C .

The vertices of the five tetrahedra T_5 , $i = 1, \dots, 5$, are given below.

1. $\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4$, and C for T_1 .
2. $\mathbf{p}_1, \mathbf{p}_4, \mathbf{p}_5$, and C for T_2 .
3. $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_5$, and C for T_3 .
4. \mathbf{p}_2, A, B , and C for T_4 .
5. $\mathbf{p}_2, \mathbf{p}_5, A$, and C for T_5 .

We use Eq. 2.12 to calculate the five tetrahedron volumes $V_{1,i}$, $i = 1, \dots, 5$, so that $V_1 = \sum_{i=1}^5 V_{1,i}$.

In the hexagon case, the space occupied by domain 1 in the cell is composed of tetrahedra and half a parallelepiped. In this instance we also use Eq. 2.11 to compute the volume of the half parallelepiped.

2.4.2 Surface Areas

As with the volume fraction β_1 , we now define the interface surface area A using similar notation.

Let $A_{i,j}^k$, where again $i = 1, \dots, C_{n_j-2}$, $j = 1, \dots, S$, and $k = 1, \dots, P$, denote the surface area of the interface surface bounded by a connected component j , with n_j edge crossings, of an interface perimeter k , given a fixed triangulation i . To compute $A_{i,j}^k$ we use the 3D coordinates of the edge crossings in a Cartesian space and apply the following result.

The area of a *triangle* with vertices \mathbf{a} , \mathbf{b} , and \mathbf{c} embedded in 3D space is given by:

$$A = \frac{1}{2} \|(\mathbf{b} - \mathbf{a}) \times (\mathbf{c} - \mathbf{a})\| \quad (2.13)$$

The glider interface surface consists of three triangular elements, as a result of Theorem 3.1. In Fig. 2.10, we follow the triangulation that was arbitrarily chosen (where vertex \mathbf{p}_1 has degree equal to four) and use the coordinates of the \mathbf{p}_i crossings, $i = 1, \dots, 5$, to calculate the surface area $A_{i,j}^k$ of the interface surface. This is given as the sum of three triangular areas:

$$A = \frac{1}{2} \|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_5 - \mathbf{p}_1)\| + \frac{1}{2} \|(\mathbf{p}_5 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)\| + \frac{1}{2} \|(\mathbf{p}_4 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\| \quad (2.14)$$

2.4.3 Formulas

Given a corner 2-coloring and an interface perimeter, let n denote the total number of edge crossings. Then $n = \sum_{j=1}^S n_j$. We use $A_{[\text{comp}]}$ to denote the area $A_{i,j}^k$ of the surface bounded by the given connected perimeter component with fixed triangulation i (*e.g.*, A_{plane} , A_{hexagon}). Similarly we use $\beta_{[\text{comp}]}$ to denote the volume fraction of domain 1 $\beta_{1,i,j}^k$ for the given triangulation of the perimeter component. These quantities are computed using Eq. 2.12, Eq. 2.11 and Eq. 2.13. We assume that the volumes of domain 1 have been normalized.

From Figs. 2.4, 2.5, and 2.6 we arrive at Table 2.1, which gives the number

Case	n	S	n_1	T_1	n_2	T_2	n_3	T_3	n_4	T_4
Corner	3	1	3	1						
Edge/Plane	4	1	4	2						
Glider	5	1	5	5						
Hexagon/Twister-I Twister-II/	6	1	6	14						
Hexagonoid	6	1	6	4						
Twisteroid	7	1	7	22						
Two Corners-I/Two Corners-II	6	2	3	1	3	1				
Three Corners	9	3	3	1	3	1	3	1		
Four Corners (i)-(ii)	12	4	3	1	3	1	3	1	3	1
Hexagon + Corner	9	2	6	14	3	1				
Hexagon + Two Corners (i)-(iv)	12	3	6	14	3	1	3	1		
Hexagonoid + Corner (i)-(iii)	9	2	6	4	3	1				
Hexagonoid + Two Corners (i)-(xii)	12	3	6	4	3	1	3	1		
Glider + Corner (i)-(ii)	8	2	5	5	3	1				
Two Edges (i)-(ii)	8	2	4	2	4	2				
Edge + Corner	7	2	4	2	3	1				
Two Hexagonoids (i)-(vi)	12	2	6	4	6	4				
Octagonoid-I (i)-(ii)	8	1	8	6						

Table 2.1: Properties of all the valid interface perimeters in 3D cells. Lower roman letters indicate the number of distinct interface perimeters with similar connected components. Note that there are two distinct interface perimeters with two corners, one corresponding to the non-isomorphic case 3, and the other corresponding to the non-isomorphic case 4. Also note that all perimeters with hexagonoid, twisteroid and octagonoid-I components (in bold) were excluded in our implementation.

of triangulations $(T_i, i = 1, \dots, S)$ for the connected perimeter components j , where $j = 1, \dots, S$, with $n_i, i = 1, \dots, S$, edge crossings in all the valid interface perimeters in 3D cells.

We use Table 2.1, to give formulas for the surface area A and volume fraction β_1 of domain 1 in each of the fourteen non-isomorphic cases. Note that Table 2.1 lists all theoretically valid interface perimeters, and that in our computational procedure we exclude all terms with hexagonoid, twisteroid, octagonoid-I or duodecagonoid-I surfaces.

For the seven non-isomorphic cases 1, 2, 5, 8, 9, 10, and 11, where $P = S = 1$, the formulas are simply given by:

$$A = \frac{1}{C_{n_{[\text{comp}]}-2}} \sum_{i=1}^{C_{n_{[\text{comp}]}-2}} A_{[\text{comp}],i} , \quad (2.15)$$

and

$$\beta = \frac{1}{C_{n_{[\text{comp}]}-2}} \sum_{i=1}^{C_{n_{[\text{comp}]}-2}} \beta_{[\text{comp}],i} , \quad (2.16)$$

where the connected perimeter component corresponds to either a corner, an edge, a glider, a plane, a hexagon, a twister-I, or a twister-II, and $C_{n_{[\text{comp}]}-2}$ is the number of triangulations of that connected component. Note that $T = C_{n_{[\text{comp}]}-2}$ in all these cases.

For the seven remaining cases 3, 4, 6, 7, 12, 13, and 14, the formulas are presented below.

1. *Case 4* Two Corners-I

$$A = \sum_{j=1}^2 A_{\text{corner},j} , \quad (2.17)$$

$$\beta = \sum_{j=1}^2 \beta_{\text{corner},j} . \quad (2.18)$$

2. *Case 13* Two Edges (i)-(ii)

$$A = \frac{1}{2} \sum_{k=1}^2 \sum_{j=1}^2 \left(\frac{1}{2} \sum_{i=1}^2 A_{\text{edge},i,j}^k \right) , \quad (2.19)$$

$$\beta = \frac{1}{2} \left(1 + \sum_{j=1}^2 \frac{1}{2} \sum_{i=1}^2 (\beta_{\text{edge},i,j}^1 - \beta_{\text{edge},i,j}^2) \right) . \quad (2.20)$$

3. *Case 6* Edge and Corner

$$A = \frac{1}{2} \sum_{i=1}^2 A_{\text{edge},i} + A_{\text{corner}} , \quad (2.21)$$

$$\beta = \frac{1}{2} \sum_{i=1}^2 \beta_{\text{edge},i} + \beta_{\text{corner}} . \quad (2.22)$$

4. *Case 12* Glider and Corner (i)-(ii)

$$A = \frac{1}{2} \sum_{k=1}^2 \left(\frac{1}{5} \sum_{i=1}^5 A_{\text{glider},i}^k + A_{\text{corner}}^k \right) , \quad (2.23)$$

$$\beta = \frac{1}{2} \left(1 + \frac{1}{5} \sum_{i=1}^5 (\beta_{\text{glider},i}^1 - \beta_{\text{glider},i}^2) + \beta_{\text{corner}}^1 - \beta_{\text{corner}}^2 \right) . \quad (2.24)$$

5. *Case 3* Two Corners-II

$$A = \sum_{j=1}^2 A_{\text{corner},j} , \quad (2.25)$$

$$\beta = \sum_{j=1}^2 \beta_{\text{corner},j} . \quad (2.26)$$

6. *Case 7* Three Corners + Hexagon and Corner

$$A = \frac{1}{2} \left(\sum_{j=1}^3 A_{\text{corner},j}^1 + \frac{1}{14} \sum_{i=1}^{14} A_{\text{hexagon},i}^2 + A_{\text{corner}}^2 \right) \quad (2.27)$$

$$\beta = \frac{1}{2} \left(\sum_{j=1}^3 \beta_{\text{corner},j}^1 + \frac{1}{14} \sum_{i=1}^{14} \beta_{\text{hexagon},i}^2 - \beta_{\text{corner}}^2 \right) \quad (2.28)$$

7. *Case 14* Four Corners (i)-(ii) + Hexagon and Two Corners (i)-(iv)

$$A = \frac{1}{6} \left(\sum_{k=1}^2 \sum_{j=1}^4 A_{\text{corner},j}^k + \sum_{k=3}^6 \left(\frac{1}{14} \sum_{i=1}^{14} A_{\text{hexagon},i}^k + \sum_{j=1}^2 A_{\text{corner},j}^k \right) \right) , \quad (2.29)$$

$$\begin{aligned}
\beta &= \frac{1}{6} \left(1 + \sum_{j=1}^4 (\beta_{\text{corner},j}^1 - \beta_{\text{corner},j}^2) \right) \\
&+ \frac{1}{6} \sum_{k=3}^5 \left(1 - \frac{1}{14} \sum_{i=1}^{14} (\beta_{\text{hexagon},i}^k + \beta_{\text{corner},1}^k - \beta_{\text{corner},2}^k) \right) \\
&+ \frac{1}{6} \left(\frac{1}{14} \sum_{i=1}^{14} \beta_{\text{hexagon},i}^6 - \beta_{\text{corner},1}^6 + \beta_{\text{corner},2}^6 \right). \quad (2.30)
\end{aligned}$$

Note that for those cases where we average over more than one interface perimeter, our formulas are consistent with the domain components so that no such case where a black domain component is added to a white domain component can occur.

2.5 Implementation in *FronTier*

In grid-based front-tracking, the contact interface is reconstructed for each cell cut by a fluid interface. In the front tracking package *FronTier*, there is a two-component block file that identifies from the components of the cell corners (the corner 2-coloring) which of the fourteen non-isomorphic cases occurs in each cell. This file can be found in the location: *FronTier/src/intfc/iblk2.c*. For each case, there is a corresponding local void function called *blk_case_k_comp2_bis()*, where $k = 1, \dots, 14$, which computes the surface area of the reconstructed interface and the volume fraction of domain 1, arbitrarily picked as the one with black corner components.

Inside each case we use volume and area functions designed to

compute these quantities for each of the connected interface perimeters (*e.g.*, *area_glider()*, *volume_hexagon()*). The arguments to these functions are the interface crossings with the cell edges and the black corners in the corner 2-coloring of the cell. Below we present the standard model used in writing these functions and explain how to use them.

2.5.1 Standard Model

The standard model of Fig. 2.11 provides, for each non-isomorphic case, a fixed and standard configuration of the interface with respect to both the 2-coloring of the cell corners and the ordering of the crossings relative to the cell corners. This model, which is set up in the code *FronTier* by specific rotation functions, eliminates ambiguities associated with looking at the same case from different frames of reference (through rotations and distinct crossing positions). We follow it consistently to design volume and area functions in each case.

In order to fix the relative position of the crossings with respect to the cell corners, we adopt the lexicographic corner ordering shown in Fig. 2.12. At the start of each of the main functions *blk_case_k_comp2_bis()*, where $k = 1, \dots, 14$, we define the cell corners according to this prescribed ordering. This specification completes the reference frame needed to write the volume and area functions. In effect, the three degrees of freedom provided by the 2-coloring of cell corners, the orientation of the i , j , and k axes and the ordering of the crossings relative to the black corners are now all fixed.

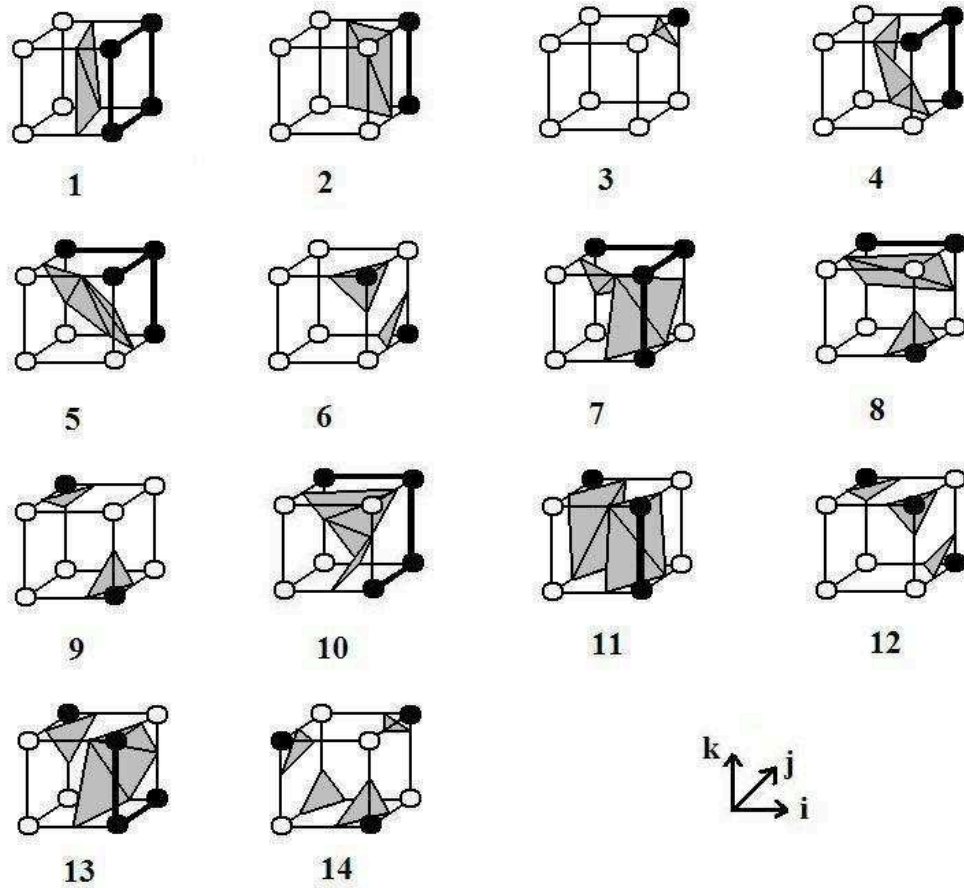


Figure 2.11: Standard model for the fourteen cases implemented in *FronTier*: 1) Plane, 2) Edge, 3) Corner, 4) Glider, 5) Hexagon, 6) Two corners-I, 7) Twister-I, 8) Edge and corner, 9) Two corners-II, 10) Twister-II, 11) Two edges, 12) Three corners, 13) Glider and corner, 14) Four corners.

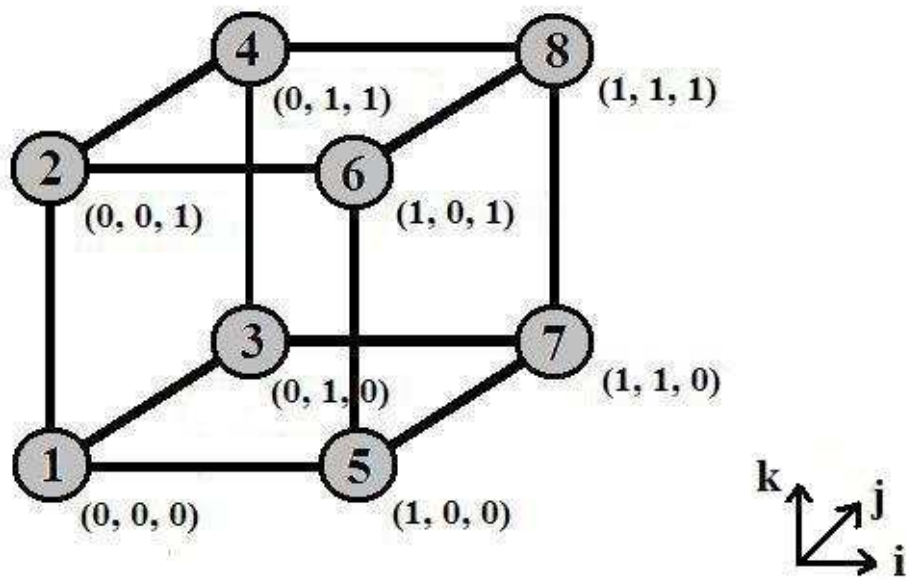


Figure 2.12: Location of the cell corners used in the standard model.

2.5.2 Area and Volume Functions

The area and volume functions are all local *float* functions that return, respectively, interface surface areas and light volume fractions for each case. Seven sets of functions describe each of the connected perimeter components, including the two topologically distinct twister perimeters. The labeling of these functions is straightforward. For instance, *area_edge()* is the area function that returns the interface surface area for an edge element and *volume_twister1()* is the volume function that returns the volume fraction for the twister-I element, as depicted in Fig. 2.11.

The arguments for the area functions are the crossings in the i , j , k directions. Since a 3D cell has twelve edges, with three sets of four edges parallel to each of the axes, we need to specify for each crossing the edge on which it is found. To do so we associate a boolean ordered pair to each crossing that conforms to the standard model and gives the location of the edge with respect to the other two directions. Thus, the location of the i , j and k crossings are provided by the ordered boolean pairs (ij, ik) , (ji, jk) and (ki, kj) . Each element of the pair has a value of 0 or 1 which, in accordance to the standard model, give the location of the edge with respect to the two other directions. For instance, if $(ij, ik) = (0, 1)$ is given as the pair associated with an i crossing, then this crossing must occur on the upper front edge, where $j = 0$ and $k = 1$. Similarly, if $(ki, kj) = (0, 1)$ is a k crossing, then it must occur on the left-back edge, where $i = 0$ and $j = 1$.

For cases with multiple crossings in the same direction, we add numbers at the end of the pairs. In the edge case, for instance, we have two j crossings

which are identified by the pairs $(ji1, jk1)$ and $(ji2, jk2)$.

Below we present the input arguments needed for each area function.

1. *float* area_corner(BLK_CRX*, *ij, ik, ji, jk, ki, kj*);
 - boolean[2](*ij, ik*): crossing in the *i* direction.
 - boolean[2](*ji, jk*): crossing in the *j* direction.
 - boolean[2](*ki, kj*): crossing in the *k* direction.
2. *float* area_edge(BLK_CRX*, *ij1, ik1, ij2, ik2, ji1, jk1, ji2, jk2*);
 - boolean[2](*ij1, ik1*): first crossing in the *i* direction.
 - boolean[2](*ij2, ik2*): second crossing in the *i* direction.
 - boolean[2](*ji1, jk1*): first crossing in the *j* direction.
 - boolean[2](*ji2, jk2*): second crossing in the *j* direction.
3. *float* area_plane(BLK_CRX*, *ij1, ik1, ij2, ik2, ij3, ik3, ij4, ik4*);
 - boolean[2](*ij1, ik1*): first crossing in the *i* direction.
 - boolean[2](*ij2, ik2*): second crossing in the *i* direction.
 - boolean[2](*ij3, ik3*): third crossing in the *i* direction.
 - boolean[2](*ij4, ik4*): fourth crossing in the *i* direction.
4. *float* area_glider(BLK_CRX*, *ij1, ik1, ij2, ik2, ij3, ik3, ji, jk, ki, kj*);
 - boolean[2](*ij1, ik1*): first crossing in the *i* direction.
 - boolean[2](*ij2, ik2*): second crossing in the *i* direction.

- `boolean[2](ji3, jk3)`: third crossing in the i direction.
 - `boolean[2](ji, jk)`: crossing in the j direction.
 - `boolean[2](ki, kj)`: crossing in the k direction.
5. *float* `area_hexagon(BLK_CRX*, ij1, ik1, ij2, ik2, ji1, jk1, ji2, jk2, ki1, kj1, ki2, kj2)`;
- `boolean[2](ij1, ik1)`: first crossing in the i direction.
 - `boolean[2](ij2, ik2)`: second crossing in the i direction.
 - `boolean[2](ji1, jk1)`: first crossing in the j direction.
 - `boolean[2](ji2, jk2)`: second crossing in the j direction.
 - `boolean[2](ki1, kj1)`: first crossing in the k direction.
 - `boolean[2](ki2, kj2)`: second crossing in the k direction.
6. *float* `area_twister1(BLK_CRX*, ij1, ik1, ij2, ik2, ji1, jk1, ji2, jk2, ki1, kj1, ki2, kj2)`;
- `boolean[2](ij1, ik1)`: first crossing in the i direction.
 - `boolean[2](ij2, ik2)`: second crossing in the i direction.
 - `boolean[2](ji1, jk1)`: first crossing in the j direction.
 - `boolean[2](ji2, jk2)`: second crossing in the j direction.
 - `boolean[2](ki1, kj1)`: first crossing in the k direction.
 - `boolean[2](ki2, kj2)`: second crossing in the k direction.

7. *float* area_twister2(BLK_CRX*, *ij1, ik1, ij2, ik2, ji1, jk1, ji2, jk2, ki1, kj1, ki2, kj2*);

- `boolean[2](ij1, ik1)`: first crossing in the *i* direction.
- `boolean[2](ij2, ik2)`: second crossing in the *i* direction.
- `boolean[2](ji1, jk1)`: first crossing in the *j* direction.
- `boolean[2](ji2, jk2)`: second crossing in the *j* direction.
- `boolean[2](ki1, kj1)`: first crossing in the *k* direction.
- `boolean[2](ki2, kj2)`: second crossing in the *k* direction.

BLK_CRX* is a structure that contains information about the crossings.

The arguments for the volume functions are the same as those of the area functions (*i.e.*, crossings in the *i, j, k* directions), with the addition of the black corners in the corner 2-colorings. We specify below the black corner input for these functions in accordance with Fig. 2.12.

1. *float* volume_corner(BLK_CRX*, ... , crn8);
2. *float* volume_edge(BLK_CRX*, ... , crn8, crn7);
3. *float* volume_plane(BLK_CRX*, ... , crn7, crn8, crn5, crn6);
4. *float* volume_glider(BLK_CRX*, ... , crn6, crn8, crn7);
5. *float* volume_hexagon(BLK_CRX*, ... , crn6, crn7, crn4, crn8);
6. *float* volume_twister1(BLK_CRX*, ... , crn5, crn6, crn8, crn4);
7. *float* volume_twister2(BLK_CRX*, ... , crn5, crn7, crn8, crn4);

2.6 Conclusions

In this chapter, we develop the averaged surface area and volume fraction formulas. These formulas support the average $\langle X_k \rangle$ of the indicator function X_k which is denoted as β_k (chapter 1). By using these formulas, we obtain the excellent agreement in the numerical closure simulation problem which is introduced in chapter 1 [4].

Chapter 3

Single Mode Surface Area in 3D

Rayleigh-Taylor Instability

The material in this chapter is based on the paper [39]. It is a pleasure to thank my coauthors for their contributions to this joint work. We emphasize here the single mode surface area in 3D. The simulation uses concentration equations and nonzero transport. We observed chaotic interface behavior even for this single mode simulation, in the sense that the interface appears to have an area proportional to Δx^{-1} , with respect to its mesh (non) convergence (i.e. rate of divergence) properties.

3.1 Introduction

We consider the Navier-Stokes equations with transport for a mixture of two compressible species,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0 , \quad (3.1)$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v} + p \delta_{ij}) = \nabla \cdot \mathbf{d} , \quad (3.2)$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (E + p) \mathbf{v} = \nabla \cdot \kappa \nabla T + \nabla \cdot \mathbf{d} \cdot \mathbf{v} \quad (3.3)$$

$$\frac{\partial \rho \psi}{\partial t} + \nabla \cdot \rho \psi \mathbf{v} = \nabla \cdot \rho \mu \nabla \psi . \quad (3.4)$$

The dependent variables ρ , \mathbf{v} , p , E , ψ , κ and μ , denote, respectively, the total mass density, the velocity, the pressure, total energy, the species mass fraction, the coefficient of heat conductivity, molecular mass diffusion.

The equation of state (EOS) is defined for each of the species as a gamma law gas, and according to [2], the mixture EOS is a gamma law gas also. \mathbf{d} is the viscous stress $\mathbf{d} = 2\nu[\mathbf{S} - \frac{1}{3}\text{tr}(\mathbf{S})\mathbf{I}]$ and S_{ij} is strain rate tensor,

$$S_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) . \quad (3.5)$$

In the reference book of Williams [51], there is a more detail approximation theory of multifluid viscosity.

We consider compressible flows coupled the concentration equation actively into the flow dynamics. The mixing problems which we study, are driven by acceleration-driven forces. The classical Rayleigh-Taylor instability with mixing regime is defined by steady acceleration of a density discontinu-

ities. The Atwood number $A = (\rho_2 - \rho_1)/(\rho_2 + \rho_1)$, with $A > 0$ is an important parameter to measure the effective buoyancy and thus the acceleration of the flow. See the overview [47]. In this single mode Rayleigh-Taylor 3D simulation, we use the numerical tool *FronTier* which is based on a front tracking algorithm with a high quality treatment of a fluid interface. In detail, we use locally grid based method, one of the front tracking methods. A front (a codimension 1 grid), specifies a fluid discontinuity location. Through a regular rectangular grid, the front moves freely. At the front, Riemann solutions, which are constructed in a normal direction, provide the physics based dynamics to move the front at each time step. The algorithm is conservative for interior cells, not cut by the front, but it is not conservative at the front. Additionally, a recent comparison study of this method (*FronTier*) to other interface methods (RAGE and PROMETHEUS) is shown in Chapter 4.

3.2 Multiscale Modeling of Single Mode 3D RT Instability

We simulate 3D single mode Rayleigh-Taylor with nonzero transport and obtain the surface area. It was already observed [37] that the interface for a related but distinct 2D Richtmyer-Meshkov instability has an length proportional to Δx^{-1} , with respect to its mesh (non) convergence (i.e. rate of divergence) properties.

We show in Fig. 3.1, 3.2, 3.3, 3.4 several time steps in the evolving flow. Here, we used transport coefficients for water (NS viscosity : $0.00085105 \text{ cm}^2/\text{ms}$,

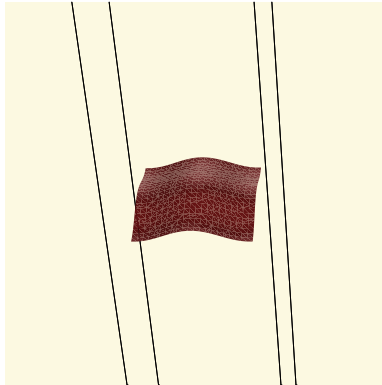


Figure 3.1: Plot of the interface between the heavy (above) and light (below) fluids at initial time for Rayleigh-Taylor fluid instability. For this study, density ratio is 2:1, the Atwood number is $1/3$, the peak to peak amplitude of initial disturbance is 0.06, grid size is $20 \times 20 \times 200$ and computational domain $1 \text{ cm} \times 1 \text{ cm} \times 10 \text{ cm}$.

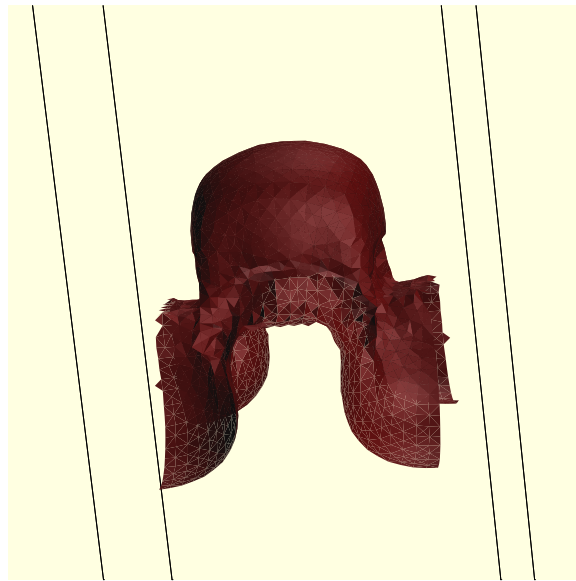


Figure 3.2: Plot of early time for Rayleigh-Taylor fluid instability. Flow and grid parameters as in Fig. 3.1.

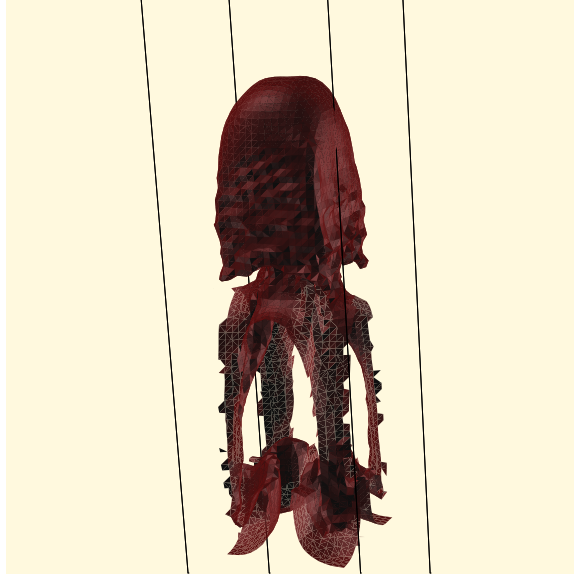


Figure 3.3: Plot of middle time for Rayleigh-Taylor fluid instability. Flow and grid parameters as in Fig. 3.1.

NS mass diffusion : $0.00016366 \text{ cm}^2/ms$, NS thermal conductivity : $0.00112; \text{ cm}^2/ms$).

In the interface surface area Fig. 3.5, we observed that the surface area is divergent with time (expressed unphysical units) in three mesh refinements. We convert the surface area to mesh units using (3.6).

$$\begin{aligned} \frac{(\text{interface surface})}{(\text{mixing zone volume})} &= \frac{\frac{(\text{physical surface area})}{(\Delta X * \Delta Y)}}{\frac{(h_{max} - h_{min}) * \text{domain X} * \text{domain Y}}{\Delta X * \Delta Y * \Delta Z}} \\ &= \frac{(\text{physical surface area}) * \Delta Z}{(h_{max} - h_{min}) * \text{domain X} * \text{domain Y}} . (3.6) \end{aligned}$$

Here h_{max} is maximum height and h_{min} is minimum height in the mixing zone.

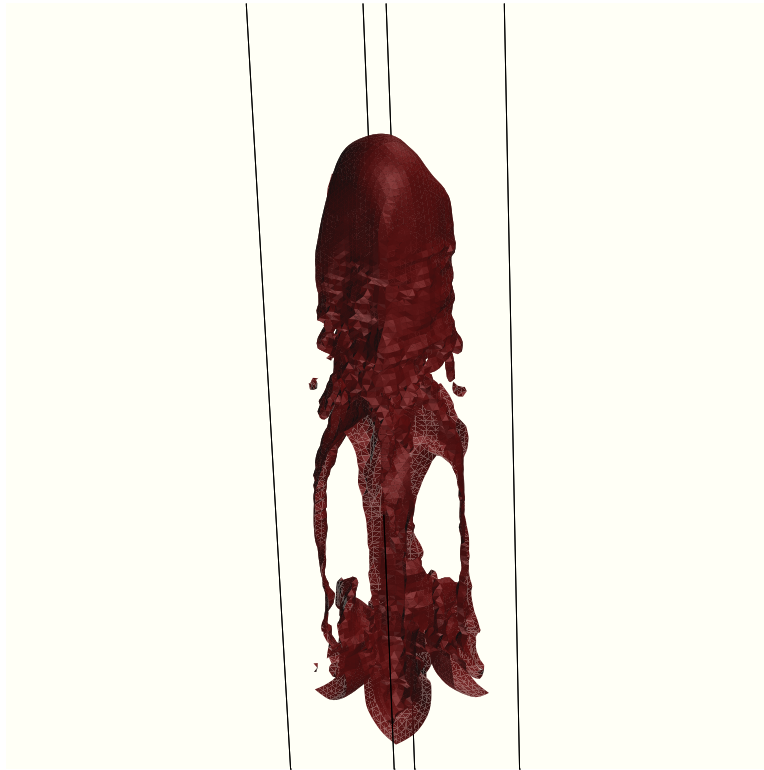


Figure 3.4: Plot of late time for Rayleigh-Taylor fluid instability. Flow and grid parameters as in Fig. 3.1.

We observe the chaotic interface behavior in Fig. 3.6. By (3.6), at late time, the mesh fraction of surface area per mixing zone mesh block ranges from 24 % to 14 %. We interpret these members as indicating that the flow is a mixture of chaotic patches and non chaotic patches.

The flow morphology of a single mode Rayleigh-Taylor disturbance is far less chaotic than a comparison multimode 2D Richtmyer-Meshkov flow, but there is still vortex shedding from the mushroom caps. The flow is more or less chaotic in these vortex shedding region at late time. So, we study the chaotic flow near the bottom of the mushroom caps and the top of the spike tips in more detail [39].

The mesh level surface fraction is time independent. In the later time, the mesh level surface fraction is about 20% relative to the mixing zone itself.

The divergent nature prototypical error analysis of the interface without transport physics is proposed as a formula form [39]:

$$\text{Error} = C_1 \times \Delta x \times (\text{Interface Area}) = C_1 C_2 , \quad (3.7)$$

C_i are $\mathcal{O}(1)$ constants independent of Δx . C_1 is related to numerical mass diffusion and might be taken as 3.0 for a typical numerical algorithm [16]. In the single mode 3D problem and for grid levels considered here, C_2 has a value in the range 0.14 to 0.24.

To study the convergence and mesh refinement, we define $Re = VL / \langle \nu \rangle$, ν is the kinematic viscosity, and $\langle \cdot \rangle$ is an ensemble average. V is the turbulent fluid velocity, $V = \sqrt{\langle \delta V^2 \rangle}$. Here, $\delta V_z = V_z - \langle V_z \rangle$ is the fluctuating part of the velocity, and L is the mixing zone height ($h_{\max} - h_{\min}$). In the simulation,

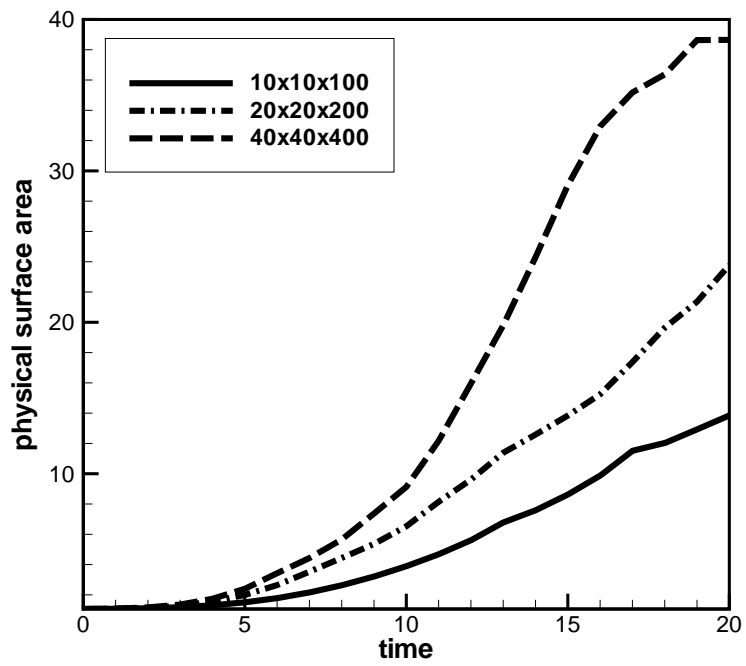


Figure 3.5: Plot of the interface surface area vs. time using physical units. Results for three mesh levels are displayed for the identical 3D single mode RT instability.

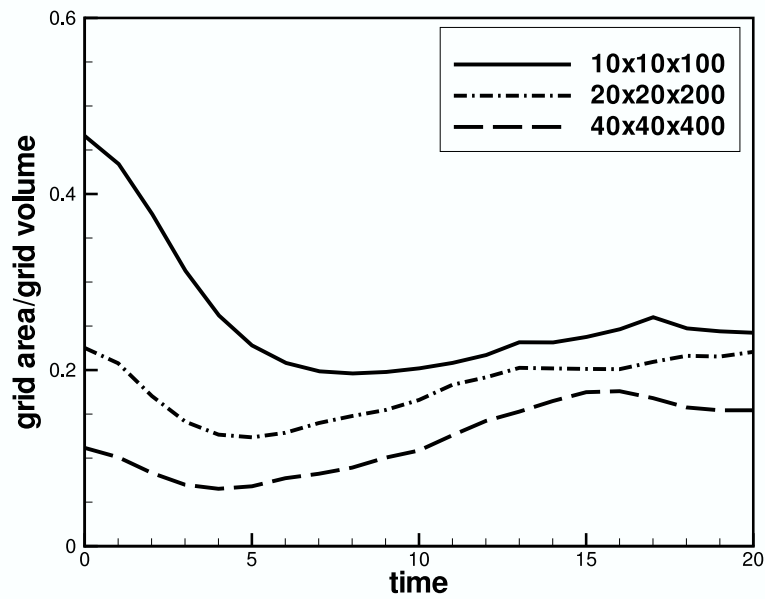


Figure 3.6: Plot of the interface area divided by the mixing zone volume vs. time. Both the area and the volume are measured in mesh units (the quantity $[\text{physical area} / \text{physical volume}] \times \Delta x$ is plotted). Results for three mesh levels are displayed for a 3D single mode RT instability.

grid size	Re	Re_{mesh}	$\lambda_{K_{mesh}}$
10 x 10 x 100	731.50	32.51	1.22
20 x 20 x 200	937.04	21.09	1.71
40 x 40 x 400	1212.36	12.55	2.80

Table 3.1: Table relating Re and λ_K to Re_{mesh} at $t = 11.0$ for several mesh levels. Kolmogorov scale $\lambda_{K_{mesh}} = \lambda_K/\Delta x$. Schmidt number $Sc = 5.2$.

we used a constant dynamic viscosity. We also use the mesh Reynolds number $Re_{mesh} = V\Delta x/\langle\nu\rangle$ and the Schmidt number $Sc = \nu/\mu$ as a new dimensionless parameter, where μ is the coefficient of molecular mass diffusion. The Kolmogorov length scale $\lambda_K = (\nu^3/\epsilon)^{1/4}$ or the viscous inner scale, approximately $50\lambda_K$, is a measure of the lengths at which viscosity plays a role, and this is related to the level of mesh refinement needed for a numerical simulation. For comparison, Δx for the finest grid is comparable to λ_K and well below the inner viscous scale, indicating that the calculation is in close to fully resolved simulation. We need to do more mesh refinement to get fully resolved simulation. Here ϵ is the dissipation rate,

$$\epsilon = \frac{\nu}{2} \|S_{ij}\|_2^2, \quad (3.8)$$

where S_{ij} is (3.5). See Table 3.1 relating Re and λ_K to Re_{mesh} .

3.3 Conclusions

In this chapter, we studied the simulation of a single mode Rayleigh-Taylor instability in 3D. By the implementation of surface area function, we can

observe the chaotic behavior of the interface flow. The flow morphology is far less chaotic than a comparison 2D multimode Richtmyer-Meshkov flow. There is still vortex shedding from the mushroom caps, at which locations the flow is more or less chaotic. So, we can study the chaotic flow of the bottom of the mushroom caps and the top of the spike tips in more detail. Here, we studied the Kolmogorov length scale which is relative to the level of mesh refinement for numerical simulation. Finally, in the future work, we need Prandtl and Batchelor scales which are need to resolve fully for numerical simulation.

Chapter 4

Comparison of 1D Eulerian ICF Simulations

In this chapter, we propose the comparison study of several compressible fluid dynamics codes in a 1D geometry suggestive of an ICF application. We study several algorithms and conduct a systematic mesh refinement study. The principal difficulty in the test problem selected is a strong shock passage through two narrowly spaced density layers, leading to multiple reflected waves. On account of these difficulties, numerical convergence requires extensive mesh refinement. Such refinement is within reach in 1D, which thus provides a convenient testing platform for the convergence requirements which would not be practical for a 2D or 3D simulation. The codes compared are *FronTier*, RAGE and PROMETHEUS, and within this framework, we compare a variety of algorithms, including PPM, TVD, Godunov-split and MUSCL. We explore both tracked and untracked simulations, where the tracking (if applied) is limited to the initially defined density discontinuities. Our first main conclusion is the overall similarity of the convergence properties. Within this similarity, we note several trends. *FronTier* excels on moderate

grids, while PPM excels on very fine ones.

4.1 Introduction

We study the identification and the nature of solution errors in a problem with multiple shock-contact interactions in planar geometry. The problem setup is suggested by an ICF capsule. The simulations are considered in 1D and model the Euler equations (zero transport). The concern over solution accuracy in the ICF simulations has led to mandates of formal efforts to assure solution accuracy. In this chapter, we analyze multiple-shock contact interactions in a 1D planar geometry. We compare the three code frameworks and different several numerical algorithms.

We consider the one dimensional Euler equations for a compressible fluid,

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} = 0 \quad (4.1)$$

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + P)}{\partial x} = 0 \quad (4.2)$$

$$\frac{\partial E}{\partial t} + \frac{\partial(u(E + P))}{\partial x} = 0 , \quad (4.3)$$

with $E = \rho e_1(P, \rho) + \rho u^2/2$. ρ is the fluid mass density, u is the fluid velocity, E is the total energy per unit volume, e is the internal energy per unit mass for the fluid, and P is the pressure. P is defined as a function of ρ and e using a γ -law equation of state with $\gamma=1.4$.

First, we consider the initialization. The heavy fluid is located in two intervals, $[0.625, 0.6875]$ and $[0.75, 0.8125]$ within the computational domain $[0, 1]$.

The region outside these intervals is occupied by the light fluid. We choose $\rho_h = 4$ or 8 or 16 as the heavy fluid density and $\rho_l = 1$ as the light fluid density. Refer to Fig. 4.1. An incoming shock is located at $x = 0.875$. The $P=1$ on $[0,0.875]$ and $P=20$ on $[0.875,1.0]$. The shock Hugoniot relations yield a post shock density $\rho = 4.65384626$ and a post shock velocity $u = -3.86229968$. All other initial velocities are zero. Reflection boundary conditions are imposed at $x = 0$ and flow through boundary conditions at $x = 1.0$. The final time 0.5 was chosen to allow the reflected lead shock wave to traverse about half of the computational domain, well past the propagated heavy density layers. For comparison, convergence properties for a single-shock contact interaction case in spherical geometry with and without tracking is considered in S.Dutta, et al.(2005) and reference therein [15]. This chapter organized as follows. In Section 4.2, we introduce three code frameworks: *FronTier*, RAGE and PROMETHEUS with several numerical algorithms. In Section 4.3, we analyze the waves and their interactions. In Section 4.4, we study L_1 -norm comparison among these numerical methods. In Section 4.5, we analyze nature of the solution errors. In section 4.6, we define and study the convergence rate. The convergence order is similar with rescaling for $\rho = 4, 8, 16$. In section 4.7, we conclude this chapter with a summary of our results and future work. Our main conclusion is the overall similarity of the convergence properties. Within this similarity, we note several trends. *FronTier* excels on moderate grids, while PPM excels on very fine ones. Finally, we suggest the future work, *FronTier* with PPM might be the best.

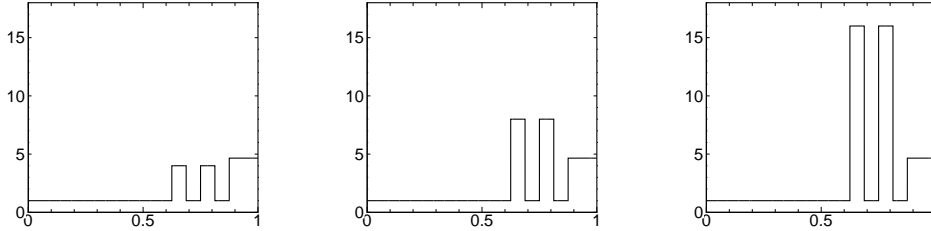


Figure 4.1: Plot of initial density vs. position. $\rho_l = 1$. Left to right, $\rho_h = 4$, $\rho_h = 8$ and $\rho_h = 16$.

4.2 Numerical Methods

4.2.1 *FronTier*

FronTier is a hydrodynamic code developed at Stony Brook University and Los Alamos National Laboratory. It supports a variety of numerical CFD algorithms. It has undergone extensive verification and validation [26, 31, 37]. In the *FronTier* code, we use a split 1st order Godunov scheme, the 2nd order Monotonic Upstream-Centered Scheme for Conservation Law (MUSCL) [12], the Total Variation Diminishing (TVD) and the TVD schemes with artificial compressibility (TVDAC) [38] algorithms. TVD and TVDAC are second order in dx and first order in dt . In the *FronTier*, we use the front tracking method.

4.2.2 RAGE

RAGE uses a high-order Godunov method to solve the Eulerian equations of gas dynamics. It uses a piecewise linear reconstruction method similar to MUSCL and a two-shock approximation to the Riemann problem solution. The code is accurate to second order in both space and time. An interface

preserver is included to keep contact discontinuities and material boundaries sharp. The code has an adaptive mesh capability, but all the simulations presented here were performed on a fixed grid. For complete details, see Gittings, et al. (2009) and references therein [22].

4.2.3 PROMETHEUS

PROMETHEUS is a code that was written primarily for computational astrophysics applications (Fryxell, Müller, and Arnett 1989 [17]). It uses the Piecewise-Parabolic Method (PPM) (Colella and Woodward 1984 [52]) for solving the Eulerian equations of gas dynamics. PPM is formally second-order accurate in both space and time, although many of the critical steps in the algorithm are performed at a higher order of accuracy to reduce the most important sources of error. The method is a high-order Godunov solver that uses a piecewise-parabolic reconstruction to obtain higher resolution than those of the MUSCL type methods. It uses a two-shock approximation to the Riemann problem solution. PPM incorporates smart dissipation algorithms to maintain sharp shocks with minimal oscillations and contains a procedure for preventing the diffusive spread of contact discontinuities. The PROMETHEUS hydrodynamic solver is also the basis for the hydrodynamic solver in the ASCI Flash code at the University of Chicago (Fryxell, et al. 2000 [18]). The hydrodynamic method has been subjected to extensive verification and validation tests (e.g. Woodward and Colella 1984 [13]; Calder, et al. 2002 [5]).

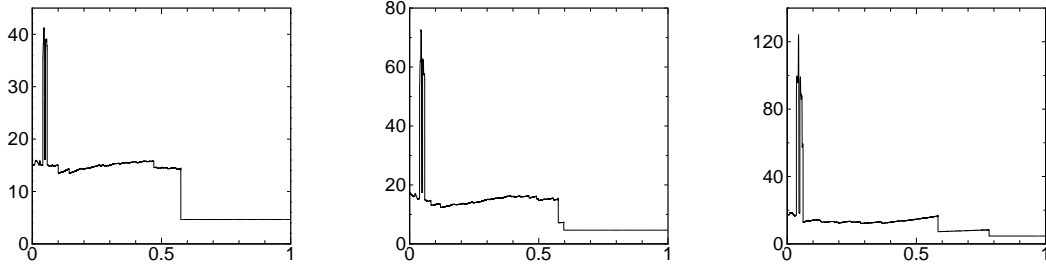


Figure 4.2: Plot of final time density vs. position for the RAGE simulation. Left to right: $\rho_h = 4, 8, 16$. The results for *FronTier* and PROMETHEUS are visually identical.

4.3 Waves and Interactions

We obtained nearly identical results using *FronTier*, RAGE and PROMETHEUS in the finest grid of 25,600 cells for each of the ρ initial values. See Figs. 4.2 and 4.3.

To analyze the waves and their interactions, we use a wave filter [23] which locates shocks waves and contacts with subgrid accuracy. This tool is based on solving a Riemann problem joining states at nearby locations $x \pm a\Delta x$, for each mesh point x and each time step. Here the parameter a is chosen as 3 for untracked contacts and 2 for shocks. The wave filter only finds the stronger among the very large number of waves generated in this problem. When a shock crosses a contact, three waves are generated. When two shocks of the opposite families cross, a new contact wave is generated. The wave filter finds more waves on finer grids, and accordingly we examine the finest grid case, grid size 25600. The four original contacts are tracked for the *FronTier*

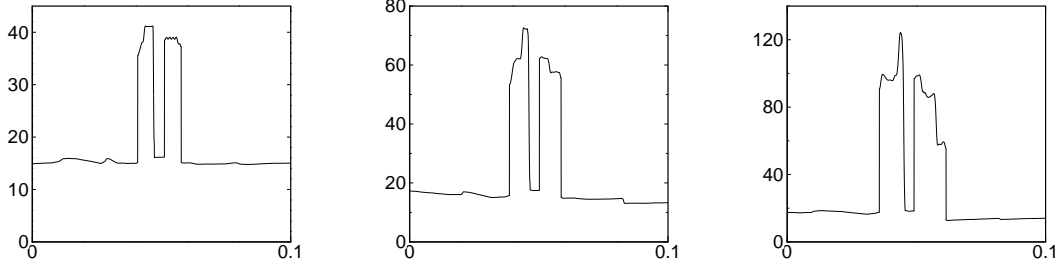


Figure 4.3: Enlarged view of the region near the origin in Fig. 4.2. Plots of density vs. position x , for $x \in [0, 0.1]$. Left to right: $\rho_h = 4, 8, 16$ using *FronTier*. The results for RAGE and PROMETHEUS are visually similar.

simulations. The location of these waves is determined within the solution itself and the wave filter is not needed in these cases.

New shocks formed during the passage of the primary shock through the two layers are identified by the wave filter. The first wave interaction occurs at $t = 0.02$, between the incoming shock and the outer edge of the outer density layer. We denote by S_T and S_R the transmitted and reflected shock from this interaction. S_T meets the next contact, giving rise to shocks S_{TT} and S_{TR} . A similar terminology is applied to subsequent interactions. These interacting waves are indicated schematically in Fig. 4.4. The first new contact, NC_1 , is generated by the interactions of S_{TRR} and S_{TTRT} at time $t = 0.06$, a second new contact, NC_2 , is generated by the interactions of S_{TRRT} and S_{TTRRR} at time $t = 0.07$ and third new contact, NC_3 , is generated by the interactions of S_{TTRRT} and S_{TTTR} at time $t = 0.08$. See the detailed density vs. position plots Figs. 4.5, 4.6. New contacts waves as well as old ones are found by the wave filter. All waves contribute to the numerical error. By $t = 0.08$, the primary

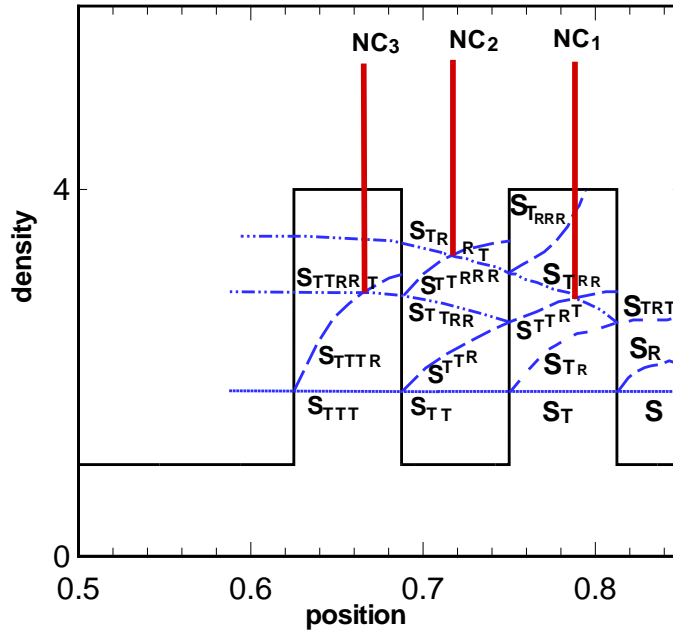


Figure 4.4: Shock-contact wave interaction schema. S : shock, T : transmitted, R : reflected, NC : new contact.

shock has passed all four contacts. Around $t = 0.20$, the primary shock is reflected at the left wall. After $t = 0.28$, the primary reflected shock has traversed the four original contacts. Around $t = 0.38$, two outgoing reflected shocks, S_R and S_{TRT} have merged. At $t = 0.44$, the primary shock from wall reflection and these outgoing reflected shocks met.

To understand the wave paths, we plot the shock-contact-capturing plot Fig. 4.8. Some but not all of the above discussed waves are detected by the wave filter and displayed in the position plot Fig. 4.8. We have visually similar results with *FronTier*, RAGE and PROMETHEUS.

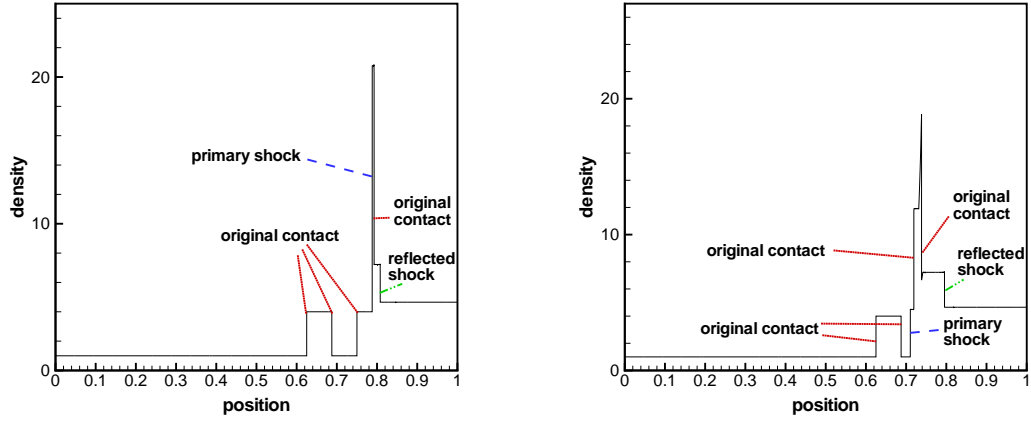


Figure 4.5: Density vs. position plot at time $t = 0.02$ (left) and $t = 0.04$ (right) with $\rho_h = 4$ in *FronTier*.

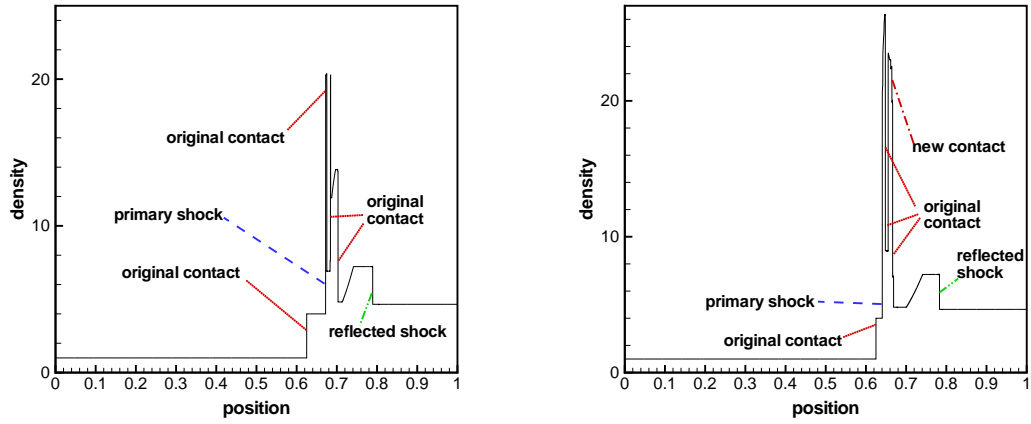


Figure 4.6: Density vs. position plot at time $t = 0.05$ (left) and $t = 0.06$ (right) with $\rho_h = 4$ in *FronTier*.

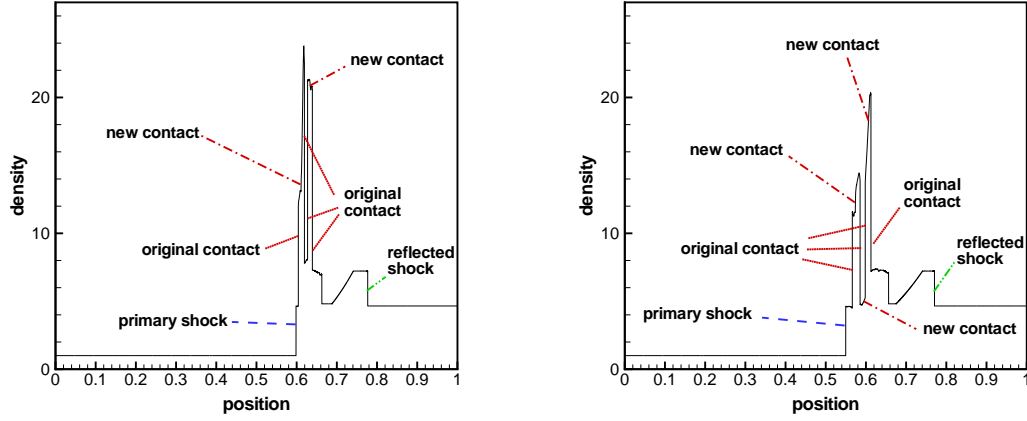


Figure 4.7: Density vs. position plot at time $t = 0.07$ (left) and $t = 0.08$ (right) with $\rho_h = 4$ in *FronTier*.

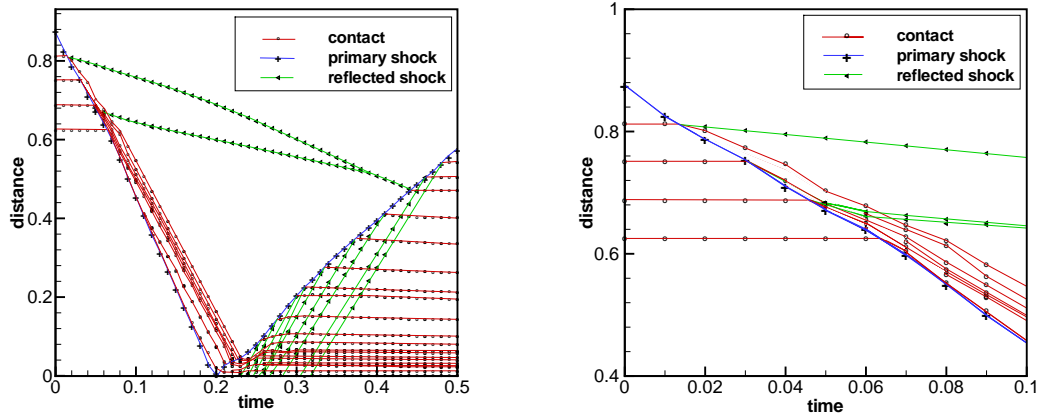


Figure 4.8: Wave position vs. time plot (left) and detailed plot between $t = 0$ and $t = 0.1$ with $\rho_h = 4$ (right) in *FronTier*. RAGE and PROMETHEUS is visually similar.

4.4 Convergence : L_1 -Norm Comparison

We define the L_1 -error norm, conceptually equal to

$$\|\rho^{\text{fine}} - \rho^{\text{coarse}}\|_{L_1}(t) = \int_0^1 |\rho^{\text{fine}} - \rho^{\text{coarse}}| dx . \quad (4.4)$$

We measure L_1 -errors based on the difference between the current solutions and the finest grid solutions with grid size 25600. All grid sizes considered are multiples of the finest grid size. For this reason, $\Delta x_{\text{coarse}}/\Delta x_{\text{fine}}$ is an integer. We can regard any coarse grid solution as defined on a fine grid, with piecewise constant data over blocks of $\Delta x_{\text{coarse}}/\Delta x_{\text{fine}}$ fine grid cells. The tracked solutions are an exception, with an additional discontinuity located within each of four coarse and fine grid cells for each of the four tracked contacts. Thus even, the tracked coarse grid solutions can also be regarded as tracked solutions defined on the fine grid. With both solutions defined on the same mesh, the definition of the L_1 norm is elementary.

For untracked solutions, the error is

$$\sum_{i=1}^{25600} |u_i^{\text{fine}} - u_i^{\text{coarse}}|/25600. \quad (4.5)$$

For the tracked case, the fine grid cells without tracked waves follow definition (4.5). For the fine grid cells containing a tracked wave in either the coarse grid or the fine grid are exceptional, but the general definition (4.4) still applies.

$$\Delta p_1 = |u_{\text{ahead}}^{\text{fine}} - u_{\text{ahead}}^{\text{coarse}}|,$$

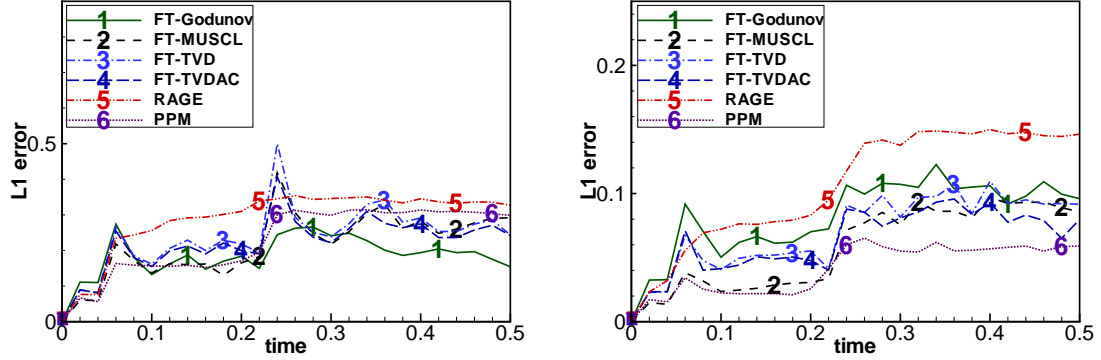


Figure 4.9: Left to right: Grid size 400 and 1600 L_1 -error with $\rho_h = 4$

$$\Delta p_2 = |u_{\text{behind}}^{\text{fine}} - u_{\text{ahead}}^{\text{coarse}}|,$$

$$\Delta p_3 = |u_{\text{behind}}^{\text{fine}} - u_{\text{behind}}^{\text{coarse}}|,$$

So, the L_1 -errors component in tracked contacts is $\sum \Delta p |x_{\text{contact}}^{\text{fine}} - x_{\text{contact}}^{\text{coarse}}|$

See the L_1 -errors plots Fig. 4.9 for $\rho = 4$.

4.5 Nature and Magnitude of Solution Errors

In this section, we study the origin of solution errors. The errors flow along the paths of the waves in the solution. Most of the error is associated in this sense with the contacts. The errors originate primarily in the shock-contact interactions, and for the untracked simulations, in the propagation of the contacts. See Figs. 4.8, 4.11, 4.12. To break down the error into individual errors, as carried by individual waves, we analyze the error associated with

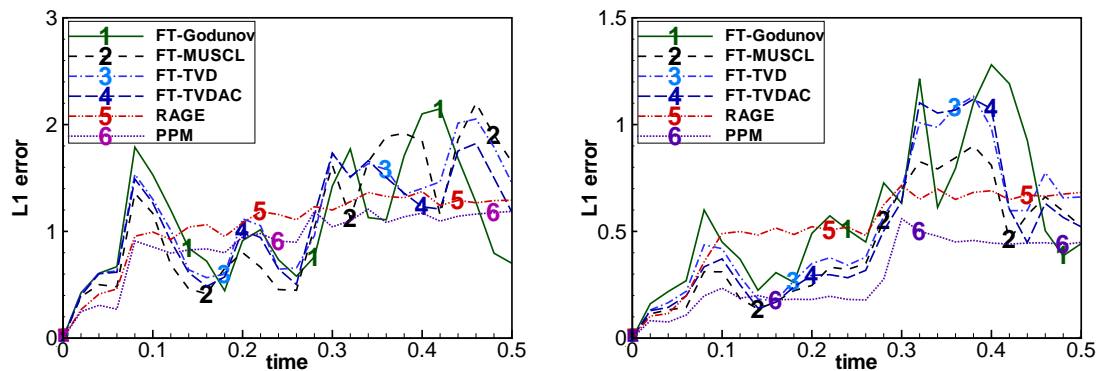


Figure 4.10: Left to right: Grid size 400 and 1600 L_1 -error with $\rho_h = 16$

specific wave types. With a wave position x given by the wave filter, or for tracked waves, by the solution itself, we calculate the local wave error (in units of mass) as

$$\int_{x-a\Delta x^{\text{coarse}}}^{x+a\Delta x^{\text{coarse}}} |\rho^{\text{fine}} - \rho^{\text{coarse}}| dx \quad (4.6)$$

at fixed time t . Note that x is not a grid cell center, and so (4.6) may involve integration over fractional fine grid cells. The parameter a is chosen to capture most of the error which propagates with the wave. We use values $a = 2$ for shocks, $a = 3$ for untracked contacts.

When the primary shock passes through the four original contacts after the primary shock wall reflection, the error increases strongly. One of our main conclusions is that the contact error comprises about 80% of the total error; of this the four original contacts account for from 30% to 50% of the total error. See Fig. 4.13 in *FronTier*-MUSCL. The time dependent error for shocks only is given in left Fig. 4.11 and for the four original contacts in right

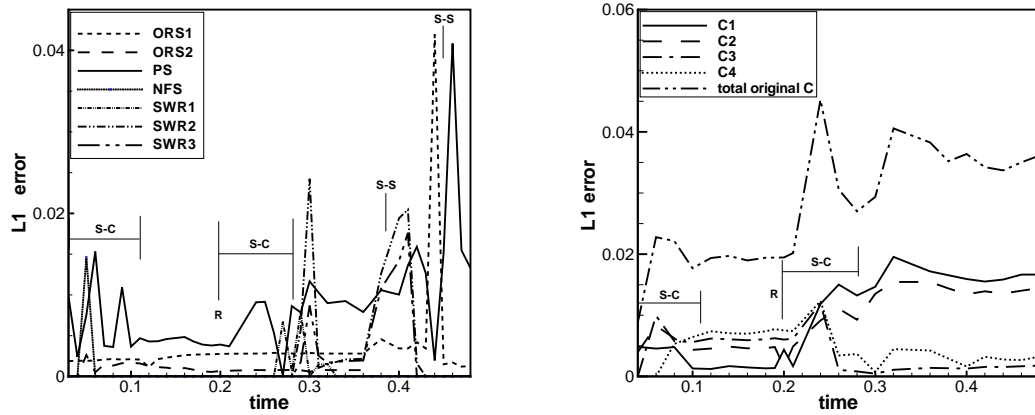


Figure 4.11: C: contact, S-C: shock-contact interaction. Left : Local wave error in units of mass vs. time with $\rho_h = 4$ and grid size 1600, *FronTier*-MUSCL. ORS: outgoing reflected shock (ORS1= S_R , ORS2= S_{TRT}), PS: primary shock, NFS: new forward shock, SWR: shock after wall reflection. Right : Local wave error for original four contacts, $\rho_h = 4$ and grid size 1600, *FronTier*-MUSCL. C: contact, S-C: shock-contact interaction, R: wall reflection, S-S: shock-shock interaction.

Fig. 4.11, 4.12.

4.6 Convergence Rates

The L_1 norm defines a norm for the error as a function of t . Since the error is generally increasing in t , we compare errors at the final time $t = 0.5$. We study the total error. See Table 4.1 for the total error.

We define a convergence rate, assuming that the convergence is in an asymptotic region, and thus is expressed as a constant times a power of Δx . On the basis of this assumption, and from comparing errors at two successive grid levels, one can determine the power, which is defined as the convergence

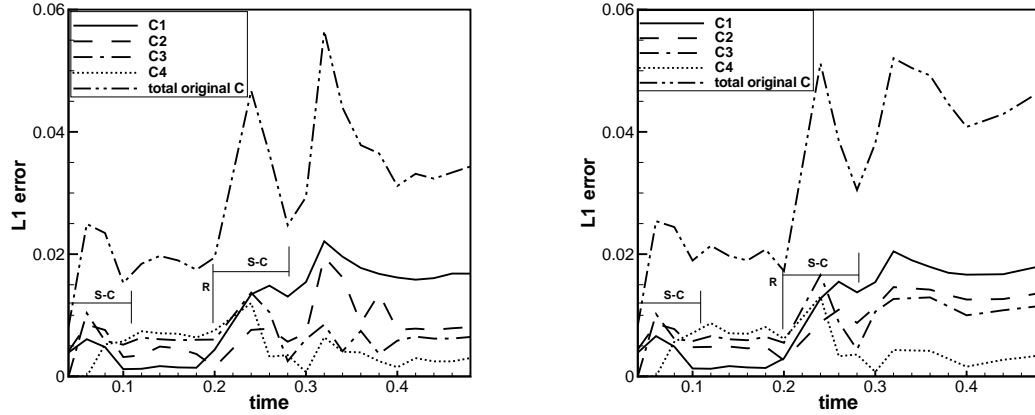


Figure 4.12: C: contact, S-C: shock-contact interaction, R: wall reflection, S-S: shock-shock interaction. Left : Local wave error for original four contacts, $\rho_h = 4$ and grid size 1600, RAGE. Right : Local wave error for original four contacts, $\rho_h = 4$ and grid size 1600, PPM.

order. Specifically, we write

$$f(\Delta x) = c\Delta x^\theta$$

as the error associated with the grid level Δx . Thus, we derive

$$\frac{f(\Delta x)}{f(\Delta x/2)} = \frac{\Delta x^\theta}{(\Delta x/2)^\theta} = 2^\theta$$

as the definition of the convergence order θ . We study the convergence order. See Table 4.2 for the convergence order. The L_1 -error and convergence order θ with rescaling for $\rho = 8, 16$ are similar, although convergence is uniformly slower as ρ increases.

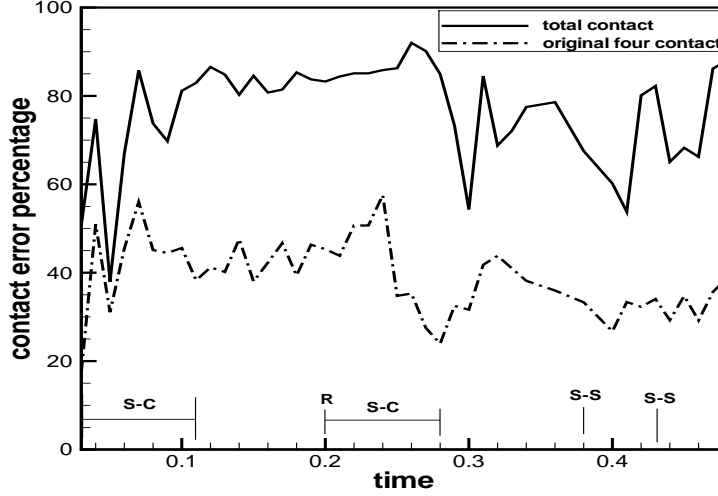


Figure 4.13: The solid line values at the top are the contact error as a percentage of the total error (with new and four original contacts) vs. time. The dash-dot lower line show the error carried by the original four contacts, expressed as a percentage of the total error vs. time. S-C: shock-contact interaction, R: wall reflection, S-S: shock-shock interaction. Here $\rho_h = 4$ and the number of grid cells is 1600.

	untracked		tracked <i>FronTier</i>			
	RAGE	PPM	Godunov	MUSCL	TVD	TVDAC
mesh	error	error	error	error	error	error
100	0.240	0.236	0.230	0.230	0.234	0.235
200	0.177	0.150	0.120	0.142	0.150	0.149
400	0.146	0.120	0.095	0.107	0.121	0.113
800	0.107	0.059	0.060	0.044	0.059	0.058
1600	0.054	0.021	0.040	0.029	0.035	0.032
6400	0.015	0.005	0.012	0.007	0.009	0.008

Table 4.1: Summary, L_1 -error from $t=0.0$ to $t=0.5$ with $\rho_h=4$ as t depends on grid size and numerical algorithm. The first order Godunov, MUSCL, TVD, TVDAC use front Tracking; RAGE and PPM are untracked.

	untracked		tracked <i>FronTier</i>			
	RAGE	PPM	Godunov	MUSCL	TVD	TVDAC
mesh vs. mesh	order	order	order	order	order	order
100 vs. 200	0.439	0.654	0.939	0.696	0.642	0.657
200 vs. 400	0.278	0.322	0.337	0.408	0.310	0.400
400 vs. 800	0.448	1.024	0.663	1.282	1.036	0.962
800 vs. 1600	0.987	1.490	0.585	0.602	0.753	0.858
1600 vs. 6400	0.924	1.035	0.869	1.025	0.980	1.00
100 vs. 6400	0.667	0.927	0.710	0.840	0.783	0.813

Table 4.2: Summary of grid size and convergence order with $\rho_h=4$. The first order Godunov, MUSCL, TVD, TVDAC are front Tracking; RAGE and PPM are untracked.

4.7 Conclusions

In this chapter, we compare several CFD algorithms using the code frame works *FronTier*, RAGE and PPM in 1D ICF simulations. On moderate grids, *FronTier* is better than PPM. On very fine grids, PPM is better than *FronTier*. However, more striking than the difference are the similarity of the results, even when comparing such different algorithms as a first order Godunov scheme and a TVD scheme with artificial compressibility. Finally, we suggest as future work, *FronTier* with PPM might be the best.

Bibliography

- [1] R. Abgrall and R. Sorell. Discrete equations for physical and numerical compressible multiphase mixtures. *J. Comp. Phys.*, 186:361–396, 2003.
- [2] T. Barberon and P. Helluy. Finite volume simulation of cavitating flow. *Computers and Fluids*, 34:832–858, 2005.
- [3] W. Bo, B. Cheng, J. Du, B. Fix, E. George, J. Glimm, J. Grove, X. Jia, H. Jin, H. Lee, Y. Li, X. Li, X. Liu, D. H. Sharp, L. Wu, and Yan Yu. Recent progress in the stochastic analysis of turbulent mixing. *Contemporary Mathematics*, 429:33–44, 2007. Stony Brook University Preprint Number SUNYSB-AMS-05-18, Los Alamos National Laboratory LAUR Number LAUR-05-7102.
- [4] W. Bo, H. Jin, D. Kim, X. Liu, H. Lee, N. Pestieau, Y. Yu, J. Glimm, and J. Grove. Multi phase closure models. *Computers & Mathematics with Applications*, 56:1291–1302, 2008. Stony Brook University Preprint Number SUNYSB-AMS-07-02.
- [5] A.C. Calder, B. Fryxell, T. Plewa, R. Rosner, L.J. Dursi, V.G. Weirs, T. Dupont, H.F. Robey, J.O. Kane, B.A. Remington, R.P. Drake, G. Dimonte, M. Zingale, F.X. Timmes, K. Olson, P. Ricker, P. MacNeice, and H.M. Tufo. On validating an astrophysical simulation code. *Astrophysical Journal ,Supplement*, 143:201, 2002.
- [6] B. Cheng. An overview of mix models. Technical report, Los Alamos National Laboratory, Los Alamos, NM 87545, 2005.
- [7] B. Cheng, J. Glimm, D. Saltz, and D. H. Sharp. Boundary conditions for a two pressure two phase flow model. *Physica D*, 133:84–105, 1999.
- [8] B. Cheng, J. Glimm, and D. H. Sharp. Density dependence of Rayleigh-Taylor and Richtmyer-Meshkov mixing fronts. *Phys. Lett. A*, 268:366–374, 2000.

- [9] B. Cheng, J. Glimm, and D. H. Sharp. A 3-D RNG bubble merger model for Rayleigh-Taylor mixing. *Chaos*, 12:267–274, 2002.
- [10] B. Cheng, J. Glimm, and D. H. Sharp. Dynamical evolution of the Rayleigh-Taylor and Richtmyer-Meshkov mixing fronts. *Phys. Rev. E*, 66:1–7, 2002. Paper No. 036312.
- [11] B. Cheng, J. Glimm, D. H. Sharp, and Y. Yu. A multiphase flow model for the unstable mixing of layered incompressible materials. *Phys. of Fluids*, 17, 2005. Paper No. 087102. LANL Preprint Number LA-UR-05-0078. Stony Brook University Preprint Number SUNYSB-AMS-05-01.
- [12] P. Colella. A direct Eulerian MUSCL scheme for gas dynamics. *SIAM Journal on Computing*, 6(1):104–117, 1985.
- [13] P. Colella and P. Woodward. The piecewise-parabolic method (ppm) for gas dynamical simulations. *J. Comp. Phys.*, 54:174, 1984.
- [14] D. A. Drew. Mathematical modeling of two-phase flow. *Ann. Rev. Fluid Mech.*, 15:261–291, 1983.
- [15] S. Dutta, E. George, J. Glimm, J. Grove, H. Jin, T. Lee, X. Li, D. H. Sharp, K. Ye, Y. Yu, Y. Zhang, and M. Zhao. Shock wave interactions in spherical and perturbed spherical geometries. *Nonlinear Analysis*, 63:644–652, 2005. University at Stony Brook preprint number SB-AMS-04-09 and LANL report No. LA-UR-04-2989.
- [16] S. Dutta, J. Glimm, J. W. Grove, D. H. Sharp, and Y. Zhang. Error comparison in tracked and untracked spherical simulations. *Computers and Mathematics with Applications*, 48:1733–1747, 2004. University at Stony Brook preprint number AMS-03-10 and LANL report No. LA-UR-03-2920.
- [17] B. Fryxell, E. Muller, and D. Arnett. Hydrodynamics and nuclear burning. *Max-Planck-Institut fur Astrophysik Report.*, 449, 1989.
- [18] B. Fryxell, K. Olson, P. Ricker, F.X. Timmes, M. Zingale, D.Q. Lamb, P. MacNeice, R. Rosner, J.W. Truran, and H. Tufo. Flash: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. *Astrophysical Journal ,Supplement*, 131:273, 2000.
- [19] E. George and J. Glimm. Self similarity of Rayleigh-Taylor mixing rates. *Phys. Fluids*, 17:054101, 2005. Stony Brook University Preprint number SUNYSB-AMS-04-05.

- [20] 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. E*, 2005. In Press. Stony Brook University Preprint number SUNYSB-AMS-05-11.
- [21] 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.
- [22] M. Gittings, R. Weaver, M. Clover, T. Betlach, N. Byrne, R. Coker, E. Dendy, R. Hueckstadt, K. New, W.R. Oakes, D. Ranta, and R. Stefan. The rage radiation-hydrodynamic code. *Computation Science and Discovery*, 2009. in press.
- [23] J. Glimm, J. W. Grove, Y. Kang, T. Lee, X. Li, D. H. Sharp, Y. Yu, K. Ye, and M. Zhao. Statistical riemann problems and a composition law for errors in numerical solutions of shock physics problems. *SISC*, 26:666–697, 2004. University at Stony Brook Preprint Number SB-AMS-03-11, Los Alamos National Laboratory number LA-UR-03-2921.
- [24] 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.
- [25] J. Glimm, H. Jin, M. Laforest, F. Tangerman, and Y. Zhang. A two pressure numerical model of two fluid mixing. *SIAM J. Multiscale Model. Simul.*, 1:458–484, 2003.
- [26] J. Glimm, W. B. Lindquist, O. McBryan, and L. Padmanabhan. A front tracking reservoir simulator, five-spot validation studies and the water coning problem. In *Frontiers in Applied Mathematics*, volume 1, pages 107–136. SIAM, Philadelphia, PA, 1983.
- [27] J. Glimm, D. Saltz, and D. H. Sharp. Renormalization group solution of two-phase flow equations for Rayleigh-Taylor mixing. *Phys. Lett. A*, 222:171–176, 1996.
- [28] J. Glimm, D. Saltz, and D. H. Sharp. Two-pressure two-phase flow. In G.-Q. Chen, Y. Li, and X. Zhu, editors, *Nonlinear Partial Differential Equations*, pages 124–148. World Scientific, Singapore, 1998.
- [29] J. Glimm, D. Saltz, and D. H. Sharp. Two-phase modeling of a fluid mixing layer. *J. Fluid Mech.*, 378:119–143, 1999.

- [30] J. M. Harris, J. L. Hirst, and M. J. Mossinghoff. *Combinatorics and Graph Theory*. Springer-Verlag, New York, U.S.A., 2000.
- [31] H. Jin and J. Glimm. Verification and validation for turbulent mixing. *Nonlinear Analysis*, 69:874–879, 2008. Stony Brook University Preprint SUNYSB-AMS-07-04.
- [32] H. Jin, J. Glimm, and D. H. Sharp. Two-pressure two-phase flow models. *Phys. Lett. A*, 2005. In press. Stony Brook University Preprint number SUNYSB-AMS-05-19 and Los Alamos National Laboratory LAUR Number LA-UR-05-7229.
- [33] H. Jin, X. F. Liu, T. Lu, B. Cheng, J. Glimm, and D. H. Sharp. Rayleigh-Taylor mixing rates for compressible flow. *Phys. Fluids*, 17:024104–1–024104–10, 2005. Stony Brook University Preprint number SUNYSB-AMS-04-06 and Los Alamos National Laboratory LAUR Number LA-04-1384.
- [34] D. Kim, J. N. Pestieau, and J. Glimm. Volume fractions and surface areas for 3-D grid cells cut by an interface. Technical Report SUNYSB-AMS-07-01, State Univ. of New York at Stony Brook, 2007.
- [35] Gabriel Lamé. Extrait d’une lettre de M. Lamé à M. Liouville sur cette question: Un polygone convexe étant donné, de combien de manières peut-on le partager en triangles au moyen de diagonales? *Journal de Mathématiques Pures et Appliquées (Journal de Liouville)*, 3:505–507, 1838.
- [36] D. Layzer. On the instability of superimposed fluids in a gravitational field. *Astrophys. J.*, 122:1–12, 1955.
- [37] H. Lee, H. Jin, Y. Yu, and J. Glimm. On validation of turbulent mixing simulations of Rayleigh-Taylor mixing. *Phys. Fluids*, 20:1–8, 2008. Stony Brook University Preprint SUNYSB-AMS-07-03.
- [38] X.-L. Li, B. X. Jin, and J. Glimm. Numerical study for the three dimensional Rayleigh-Taylor instability using the TVD/AC scheme and parallel computation. *J. Comp. Phys.*, 126:343–355, 1996.
- [39] H. Lim, Y. Yu, H. Jin, D. Kim, H. Lee, J. Glimm, X.-L. Li, and D.H. Sharp. Multiscale models for fluid mixing. *Comput. Methods Appl. Mech. Engrg.*, 197:3435–3444, 2008.

- [40] E. E. Meshkov. Instability of the interface of two gases accelerated by a shock wave. *Izv. Akad. Nauk SSSR, Mekh. Zhidk. Gaz.*, 5:151, 1969.
- [41] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, Cambridge, U.K., 1994.
- [42] G. Polya and R. C. Reade. *Combinatorial Enumeration of Groups, Graphs and Chemical Compounds*. Springer-Verlag, New York, U.S.A., 1987.
- [43] K. I. Read. Experimental investigation of turbulent mixing by Rayleigh-Taylor instability. *Physica D*, 12:45–58, 1984.
- [44] R. D. Richtmyer. Taylor instability in shock acceleration of compressible fluids. *Comm. Pure Appl. Math.*, 13:297–319, 1960.
- [45] R. Saurel and R. Abgrall. A multiphase godunov method for compressible multifluid and multiphase flows. *J. Comput. Phys.*, 150:425–467, 1999.
- [46] R. Saurel, O. Le Metayer, J. Massoni, and S. Gavrilyuk. Shock jump relations for multiphase mixtures with stiff mechanical properties. *Shock Waves*, 16(3):209–232, 2007.
- [47] D. H. Sharp. An overview of Rayleigh-Taylor instability. *Physica D*, 12:3–18, 1984.
- [48] V. S. Smeeton and D. L. Youngs. Experimental investigation of turbulent mixing by Rayleigh-Taylor instability (part 3). AWE Report Number 0 35/87, Atomic Weapons Establishment (AWE), 1987.
- [49] G. I. Taylor. The instability of liquid surfaces when accelerated in a direction perpendicular to their planes I. *Proc. R Soc. London A*, 201:192–196, 1950.
- [50] A. Tucker. *Applied Combinatorics*. John Wiley and Sons, New York, U.S.A., 1995.
- [51] Forman Williams. *Combustion Theory*. Addison-Wesley Co., Reading, 1965.
- [52] P. Woodward and P. Colella. The numerical solution of two-dimensional fluid flow with strong shocks. *J. Comp. Phys.*, 54:115, 1984.

- [53] Y. Yu, M. Zhao, T. Lee, N. Pestieau, W. Bo, J. Glimm, and J. W. Grove. Uncertainty quantification for chaotic computational fluid dynamics. *J. Comp. Phys.*, 2005. Stony Brook Preprint number SB-AMS-05-16 and LANL preprint number LA-UR-05-6212.