INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Modeling Interface Advection in the Die Casting Processes

by

Liang Zhou, B.Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of

Master of Engineering

Department of Mechanical and Aerospace Engineering
Ottawa-Carleton Institute
for Mechanical and Aerospace Engineering

Carleton University
Ottawa, Ontario
Canada
December, 1998

© 1998
1998, Liang Zhou

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-36914-5

The undersigned hereby recommend to
The Faculty of Graduate Studies and Research
acceptance of the thesis,

Modeling Interface Advection in the Die Casting Processes

submitted by

Liang Zhou, B. Eng.
in partial fulfilment of the requirements for
the degree of Master of Engineering

A handwritten signature in cursive script, appearing to read "P. Hesmily", positioned above a horizontal line.

Chair, Department of
Mechanical and Aerospace Engineering

A large, stylized handwritten signature in cursive script, appearing to read "J. Goldak", positioned above a horizontal line.

Thesis Supervisor

Carleton University
January 1999

Abstract

In a casting process, which is characterized by topologically-complex metal/air interfaces having high density ratios and high surface tension, how to model the interfacial flow kinematically and dynamically is a very important task. This thesis implements D.L. Youngs' PLIC method for simulating the interfacial flow and Miller's integration method for evaluating the surface tension forces. The PLIC model includes two steps: interface construction and interface advection. The surface tension model includes two steps: computing the surface tension force per unit area arising from local interface curvature and local (tangential) variations in the surface tension coefficient, and replacing the forces with a smoothly varying volumetric force by Peskin's delta function. Numerical results are presented in two and three dimensions to illustrate our models.

Acknowledgments

I would like to express my deep gratitude to my supervisor Prof. John A. Goldak for his knowledge, guidance and supervision during my studies. Many thanks to the colleagues in MMO group for invaluable help and supporting, specially to Wenqin Shao, Andrea Zypchen, Jianguo Zhou, Daniel Downey, Marco Antonio Pego Guerra, Jason Luo, Victor Aldea, Simei Gu, and Rong Wang. The financial support of the Material and Manufacturing Ontario(MMO) (formerly Manufacturing Research Corporation of Ontario) and Carleton University is highly appreciated. Finally, I would like to express my deep gratitude to my aunt Youdong Zhou, my parents, my grandparents and uncle Maoshi Gu for years of love and support.

Contents

1	Introduction	1
1.1	Importance of modeling free surface	1
1.2	Numerical simulation	3
1.2.1	Incompressible flow	4
1.2.2	Interface approximation	5
1.2.3	Finite element method	6
1.3	Scope of work	10
2	Literature Review	11
2.1	Front tracking	12
2.2	Level Set method	13
2.3	Volume-Of-Fluid Approach	15
2.3.1	SLIC	17

2.3.2	PLIC	18
3	Interface Advection	21
3.1	Interface construction	23
3.1.1	Calculation of the normal to the interface	23
3.1.2	Volume fraction matching	32
3.1.3	Volume Of liquid from the donor cell to the acceptor cell	44
3.2	Interface Advection	65
4	Surface tension on free surfaces	71
4.1	The surface tension force on a triangulated curved surface . .	72
4.2	Triangulated algorithm on the surface	77
4.3	Peskin's Delta Function algorithm	78
5	Numerical Results	86
5.1	Rigid body motion of a sphere	86
5.1.1	VOF initialization	87
5.1.2	Sphere translation results	88
5.2	Rotation of disc	100
5.2.1	Vof initialization	101

5.2.2	Disc rotation results	101
5.3	Surface Tension modeling results	113
6	Conclusions and Future Work	119
6.1	Conclusions	119
6.2	Future work	121
7	References	122

List of Figures

1.1	The free surface Π	3
1.2	The stresses acting on Π	3
1.3	FEM Computation Domain.	7
1.4	The mass flow rate across $\partial\Omega$	8
2.1	Reconstruction interface using SLIC approximation (advection in x-direction).	18
2.2	Reconstruction interface using PLIC approximation.	19
3.1	$3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in +x direction.	25
3.2	$3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in -x direction.	26

3.3	$3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in +y direction.	27
3.4	$3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in -y direction.	28
3.5	$3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in +z direction.	29
3.6	$3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in -z direction.	30
3.7	Local coordinates used in interface construction.	34
3.8	Triangular intersection.	37
3.9	Quadrilateral Section A.	39
3.10	Pentagonal Section.	41
3.11	Hexagonal Section.	42
3.12	Quadrilateral Section B.	43
3.13	The Flux Across Cell.	46
3.14	The Flux Across Cell: Triangular Section Case.	48
3.15	The Flux Across Cell: Quadrilateral Section A Case 1.	50
3.16	The Flux Across Cell: Quadrilateral Section A Case 2 Case 3.	51
3.17	The Flux Across Cell: Pentagonal Section Case.	54

3.18	The Flux Across Cell: Hexagonal Section Case.	58
3.19	The Flux Across Cell: Quadrilateral Section B Case.	63
4.1	Definition of tangent vector $\mathbf{t}$	74
4.2	m nodes share node i in counterclockwise order.	75
4.3	One-dimensional delta function.	81
5.1	Sphere translation in the x-direction (time step = 0).	90
5.2	Sphere translation in the x-direction (time step = 20).	91
5.3	Sphere translation in the x-direction (time step = 40).	92
5.4	Sphere translation in the x, y-direction (time step = 0).	94
5.5	Sphere translation in the x, y-direction (time step = 20).	95
5.6	Sphere translation in the x, y-direction (time step = 40).	96
5.7	Sphere translation in the x, y, and z-direction (time step = 0).	97
5.8	Sphere translation in the x, y, and z-direction (time step = 20).	98
5.9	Sphere translation in the x, y, and z-direction (time step = 40).	99
5.10	Disc rotation in 2-D (time step = 0).	103
5.11	Disc rotation in 2-D (time step = 100).	104
5.12	Disc rotation in 2-D (time step = 200).	105
5.13	Disc rotation in 3-D (time step = 0).	107

5.14	Disc rotation in 3-D (time step = 25).	108
5.15	Disc rotation in 3-D (time step = 50).	109
5.16	Disc rotation in 3-D (time step = 75).	110
5.17	Disc rotation in 3-D (time step = 100).	111
5.18	Disc rotation in 3-D (time step = 200).	112
5.19	unit cube with 8 triangles on each face.	114
5.20	The surface tension forces on the nodes on triangulation of the unit cube.	116

List of Tables

5.1	The variation with time of the total VOF within the sphere during translation in the x-direction.	89
5.2	The variation with time of the total VOF within the sphere during the translation in the x and y-direction.	93
5.3	The variation with time of the total VOF within the sphere during the translation in the x, y and z-direction.	100
5.4	The variation with time of the total VOF within the disc (2D)	102
5.5	The variation with time of the total VOF within the disc (3D).	106
5.6	The surface tension force on 8 corners on the unit cube.	115
5.7	The surface tension force on 6 nodes in the center of each face on the unit cube.	117

5.8	The surface tension force on 12 nodes on the middle of each	
	edge on the unit cube.	118

Chapter 1

Introduction

1.1 Importance of modeling free surface

Fluid flows with free surfaces or material interfaces occur in a large number of natural and technological processes. In industrial applications, casting processes, mold filling, thin film processes, and fluid jetting devices are a few of the areas in which material interfaces occur. Campbell [2] points out that casting is a very economic method of forming a component. Casting processes encompass a wide breadth of complex physical phenomena, ranging from fluid flow and free surface dynamics (Mold filling) to the heat transfer and phase change process (parts cooling and solidification). Mould filling in-

involves the insertion of approximately incompressible liquid metal alloys into complex 3-D moulds that are constructed from sand, wax, stainless steel, graphite, etc. The insertion usually takes place via gravity-pouring or pressure injection into a pre-designed system of runners, sprues, and gates. The mould-filling process is characterized by topologically-complex metal/air interfaces having high density ratios (> 1400) and high surface tension ($10^2 - 10^3$ dynes/cm) [10]. Therefore, how to model interfacial flow kinematically and dynamically becomes a very important task for scientists. For example, in die casting processes, under certain pressure and temperature conditions, how does the molten metal injected into the mould behave before it starts solidification? What is the velocity field, pressure field of the liquid? Campbell [2] mentioned that when the temperature is too low, phase change might occur; when the cross-section of a mould is too narrow (in general less than approximately 2 mm), the metal will be subject to the resistance of surface tension effects which can be safely ignored in larger sections, thus the metal may not be able to fill thin parts of the mould. Aerofoils on propellers and turbine blades are typical examples.

The fluid system can be shown schematically as Figure 1.1. A small volume of this fluid system is denoted by Ω , and Γ is the boundary of the volume.

Figure 1.1: The free surface Π .

Figure 1.2: The stresses acting on Π .

Ω and Γ are fixed in space. The free-surface will be denoted by Π . Across the boundary separating the two fluids, the velocity u is continuous. The stresses acting on the free surface are shown in Figure 1.2.

1.2 Numerical simulation

Numerical simulations are, in principle, ideally suited to study the free surfaces. However, in practice, the numerical computation of flows with free surfaces is difficult, especially if the interface separates fluids of dramatically

different densities such as air and molten metal in casting processes.

1.2.1 Incompressible flow

It will be assumed that both fluids (metal liquid and air) are incompressible and Navier-Stokes equations govern the motion in each fluid, then the assumption $\frac{D\rho}{Dt} = 0$ is made, where $\frac{D}{Dt} = \frac{\partial}{\partial t} + u \cdot \nabla$, is the material derivative.

Then the following constraint for Navier-Stokes equation is obtained:

$$\rho \operatorname{div}(u) = 0 \quad (1.1)$$

$$\operatorname{div}(u) = 0 \quad (1.2)$$

Therefore, the Navier-Stokes equations for incompressible flow is:

$$\rho \left[\frac{\partial u}{\partial t} + (u \cdot \nabla)u \right] = -\nabla p + \mu \operatorname{div}(\nabla u) \quad (1.3)$$

Where

u is velocity

p is pressure

ρ is density

t is time

μ is the coefficient of kinematic viscosity

There are a variety of highly efficient numerical algorithms for the solution of incompressible Navier-Stokes equations. Since the Mrco group solved this problem some time ago, this thesis will not discuss it further. For further information, the interested reader should consult reference papers such as [3],[4],[5],[8],[9].

1.2.2 Interface approximation

There are two basic approaches to approximate flows with an interface: “capturing” and “tracking”. In interface capturing, the basic approach is to use a finite volume discretization of the whole domain, while modifying the numerical approximation to minimize numerical difficulties. This generally involves using a high (usually second) order scheme and adding artificial viscosity around the front to diffuse it slightly to avoid oscillations. Shock-capturing has become considerably more sophisticated because of the introduction of schemes that explicitly enforce monotonicity through a nonlinear step, and at same time maintain high order “almost everywhere”. These methods gen-

erally serve well for shocks but work less well for contact discontinuities. Furthermore, relatively fine grids are still needed, thus limiting most unsteady flow simulations in two or three dimensions. An alternative to front-capturing is interface tracking, where additional computational elements are introduced explicitly to keep track of the front. This generally reduces by a considerable amount the resolution needed to keep the front sharp, and can eliminate numerical diffusion altogether. However, tracking is best suited for well-defined fronts that are easily identifiable in the initial conditions. There are three basic methods for tracking the interface (which will be discussed in detail in Chapter 2):

1. Front tracking method
2. The level set of some function (Level set methods)
3. Collection of volume fractions (volume-of-fluid method)

1.2.3 Finite element method

The finite element method (FEM) is a method of piecewise approximation in which the approximating function is formed by a linear combination of simple functions.

Figure 1.3: FEM Computation Domain.

Every FEM solver must contain: (see Figure 1.3)

1. Computaional domain plus PDE (Partial Differential Equation).
2. Initial Condition at time= 0.
3. Boundary Condition, which normally include Dirichlet B.C and Neumann B.C.
4. Material Parameters.

Note that: all the boundary condition must be applied on nodes in our discrete formulation.

Partial Differential Equation

Figure 1.4: The mass flow rate across $\partial\Omega$.

The PDE we solved here is the law of conservation of mass.

The rate of change of mass in region Ω is:

$$\frac{d}{dt}m(\Omega, t) = \frac{d}{dt} \int_{\Omega} \rho(x, t) dV = \int_{\Omega} \frac{\partial \rho}{\partial t}(x, t) dV \quad (1.4)$$

Let $\partial\Omega$ denotes that the boundary of Ω , n denote the unit outward normal defined at the point of $\partial\Omega$. and let dA denote the area element of $\partial\Omega$. and the mass flow rate per unit area is $\rho u \cdot n$. (See Figure 1.4)

As we know, the rate of increase of mass in Ω equals the rate at which mass is crossing $\partial\Omega$ in the inward direction:

$$\frac{d}{dt} \int_{\Omega} \rho dV = - \int_{\partial\Omega} \rho u \cdot n dA \quad (1.5)$$

Eqn. 1.5 is known as the integral form of the law of conservation of mass.

and is equivalent to:

$$\int_{\Omega} [\frac{\partial \rho}{\partial t} + \text{div}(\rho u)] dV = 0 \quad (1.6)$$

since this is true for all Ω , it is equivalent to:

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho u) = 0 \quad (1.7)$$

Eqn. 1.7 is the differential form of the law of conservation of mass. also known as the continuity equation.

Initial condition:

The given VOF (volume of fluid in the cell) distribution and velocity field.

Boundary condition:

The boundary condition we consider here is only Dirichlet B.C:

$$\text{inlet flow: } u \cdot n < 0.0$$

which is the mathematical BC for a first order advection equation.

Material parameter:

The density of fluid, liquid metal and air in this case.

1.3 Scope of work

Our current focus is on the development of numerical methods needed to model the free surface flow during the mould filling stage of a casting process. Accurate modeling of interfacial flows benefits from high fidelity algorithms for the kinematics and dynamics of interface. Algorithms for interface kinematics must address the discrete representation of the interface and its advection through the computational domain. Algorithms for interface dynamics must model physics specific to and localized at the interface [10].

In the present work, a numerical algorithm is developed for the analysis of volume tracking of arbitrary topology interfaces on fixed grids in two or three dimension based on Youngs' method [18]: piecewise linear interface calculation or PLIC. This is intended to replace the VOF algorithm the Mrco group uses now. More detail about Youngs' method is presented and several mistakes in Youngs' paper are pointed out. The analysis of the effect of surface tension on the interface is also touched on.

Computer algorithms are implemented to accomplish this analysis. and the applicability of this approach is demonstrated by some numerical tests.

Chapter 2

Literature Review

There are two basic approaches to simulating an interfacial flow: “capturing” and “tracking”. In front capturing methods, a data structure is defined in the entire computational domain and any discontinuities in the domain are captured and forced to propagate at the interface velocity. The disadvantage of this method is that many errors are accumulated while evolving the field. Numerical diffusion often occurs, thus the discontinuities representing the interface can rapidly disappear.

Interface tracking is one alternative to the capturing method. Interface tracking describes the interface by using additional computational elements. This can be accomplished directly by specifying the coordinates of the front

in some manner (front tracking) or indirectly using the volume fraction in each control volume (volume tracking). There are three front tracking techniques in which considerable progress has been made: front tracking, level set method, and volume-of-fluid (VOF) methods. However, since VOF methods are reasonably accurate and relatively easy to implement they are widely utilized.

Interface tracking methods usually solve the following equation:

$$\partial_t \rho + \nabla \cdot (u\rho) = 0 \quad (2.1)$$

where u is the velocity and ρ is density.

For incompressible flow, $\nabla \cdot u = 0$, equation 2.1 changes to:

$$\partial_t \rho + u \cdot \nabla \rho = 0 \quad (2.2)$$

2.1 Front tracking

The front tracking method represents the interface directly using Lagrangian markers. The markers are connected to form a front which moves inside a stationary Eulerian mesh. When the interface moves and deforms, interface points are added, deleted and reconnected [6]. (The idea of passing interface

information between the moving Lagrangian interface and the stationary Eulerian mesh is borrowed from the immersed interface method [11]. [15].) Sharp interfaces distribute the information at the interface over mesh points near the interface by using a smooth distribution function. Thus the front has finite thickness to provide stability and smoothness and the numerical diffusion is reduced. Although the front tracking method is very successful in dealing with arbitrarily topology in 2-Dimensions, implementation in 3-Dimensions is not easy.

2.2 Level Set method

The level set method was introduced by Osher and Sethian [14] a decade ago. Since then, its use has grown considerably. This interfacial model has been used in many fields, including casting processing, and is extremely popular because the mathematical formalism attracts leading numerical analysts.

In a level set method, the interface is represented as the level set of some function ϕ . ϕ is defined as $\phi(\mathbf{x}, t = 0) = \pm d$, where d is the distance from $\mathbf{x}$ to interface. The sign of d must be positive if $\mathbf{x}$ is outside the initial interface and negative if $\mathbf{x}$ is inside. When ϕ equals 0, the zero level set

coincides with the initial interface. A dynamic equation which describes the moving interface is [14]:

$$\frac{\partial \phi}{\partial t} + F|\nabla \phi| = 0 \quad (2.3)$$

Where F is the speed of the interface in the outward normal direction. For most interfacial flow, F is a curvature-dependent speed, i.e. $u \cdot \hat{n}$ — the flow velocity normal to interface, where $\hat{n}$ is the unit interface normal:

$$\hat{n} = \frac{\nabla \phi}{|\nabla \phi|} \quad (2.4)$$

The level set approach has two inherent strengths. The first is that the level set function ϕ helps compute the interface normal and curvature quite easily, which are two important tasks in free surface modeling. The second advantage of this approach is that when the shape of the front changes dramatically no special procedures are required. At each time step, the value of the function ϕ is updated, and ϕ remains zero at the location of the interface at each new time.

If the level set function ϕ is found to be conserved, this need not imply the conservation of the mass. A loss of mass has been found in many test cases for this method.

2.3 Volume-Of-Fluid Approach

Volume-Of-Fluid (VOF) approach is the most widely used tracking method for modeling free surface flows, because it is relatively easy to implement. it is topologically robust, and it is a well-documented algorithm. All VOF methods begin by computing the volume fraction of the fluid in each element in the computational domain based on the interface geometry. The volume fraction, VOF, is usually defined as the ratio of the volume of fluid in the cell to the total cell volume. Hence, $VOF = 0$ when the cell is empty of fluid. $VOF = 1$ when the cell is filled with fluid, and $0 < VOF < 1$ for a mixed cell containing a liquid/air interface. So, instead of using the true geometry of the interface, the volume fraction is used to track the interface indirectly through a discrete continuity equation. The fundamental property of all volume-of-fluid methods is that they are based on discretizing the volume (or the volume fraction, VOF) of the fluids. As we know, conservation of mass is equivalent to conservation of volume in an incompressible flow, the motion of the interface in such a flow could be modeled by solving the conservation of the VOF. Therefore, a conservative finite difference method could be used to update the volume fractions, and adjusting for the numerical errors. the

volume/mass of each fluid in the computation is conserved.

Since many slightly different interface topologies result in identical volume fractions, the precise location of the interface is somewhat uncertain at a given time step. A piecewise polynomial curve is used to describe the interface topology in each mixed cell. The order of the polynomial depends on the method, and the higher the order of the equation, the more accurate the method. The volume fractions of the surrounding cells are used to determine the coefficients of the polynomial curve. The volume fraction cut out by the piecewise curve must match the cell VOF precisely. This is done by adjusting the location of this approximate piecewise interface.

In order to maintain the discontinuity of the interface, the volumetric fluxes are calculated geometrically by tracing back along the fluid path characteristics, and determining the amount of fluid extracted through the face during the time step (i.e., area under the curve). The volume fractions are then updated for the next time step by solving the discrete continuity (advection) equation [17].

Price and Rowe [17] argue that the VOF formulation inherently conserves mass and eliminates numerical diffusion normal to the front, while at the same time it naturally adapts to the break up or merging of interfaces. One of

the earliest VOF algorithms represented the interface simply by a horizontal or vertical line that truncated an area equal to the fluid volume fraction in the cell (Simple line interface calculation or SLIC) [13]. Youngs [18] advanced the VOF approach with a method commonly referred to as PLIC (piecewise linear interface calculation). PLIC applies sloped piecewise lines to reconstruct the interface. The assumed interface geometry, interface construction, and volume flux calculation are key features of a VOF method. What follows is a discussion of those features in the SLIC and PLIC methods, respectively.

2.3.1 SLIC

The original simple line interface calculation (or SLIC) in 2D uses either a horizontal or vertical line to represent the interface (see Figure 2.1). The existence of fluid in the two neighboring cells (in the coordinate direction being advected) exclusively determines the outward normal of this line, since the reconstruction and advection steps are intrinsically one-dimensional. Multi-dimensional problems are treated as a number of one-dimensional cases, via Strang splitting. When the fluid advects in a different direction, the orientation of the line is allowed to vary. Over simplification may cause unphysical

Figure 2.1: Reconstruction interface using SLIC approximation (advection in x-direction).

interpretation of materials to occur. The SLIC method typically creates unphysical numerical flotsam and jetsam, meaning that small bits of material tend to be ejected unphysically from the interface, separating from the main body of fluid. Since the SLIC method is very easy to implement it is still widely used, in spite of its disadvantages.

2.3.2 PLIC

The piecewise linear interface calculation (or PLIC) reconstructs the interface using sloped piecewise lines in 2D (see Figure 2.2). The accuracy of the PLIC

Figure 2.2: Reconstruction interface using PLIC approximation.

reconstruction depends entirely on the method used to compute the slope of the lines.

Unlike the SLIC method, the piecewise approximation doesn't align the lines in the coordinate directions. In other words, the PLIC approximation is multi-dimensional and is therefore unrelated to the advecting mesh direction when operator splitting is applied. However, this reconstruction step must be repeated in each direction (i.e., two reconstruction steps in 2-D, three steps in 3-D).

From figure 2.1 and figure 2.2, we can tell that the geometric assumptions made for the interface reconstruction are very important: a piecewise linear

reconstruction yields a more accurate interface representation, relative to a piecewise constant reconstruction.

The PLIC algorithm is very powerful at solving for the large scale features of an interface, but it has difficulty in tracking the finer scale features. Which means that the PLIC method has a tendency to smooth the high curvature regions (also called “numerical surface tension”). The presentation of numerical surface tension is due to the assumptions inherent in the reconstruction. i.e. when the piecewise linear interface approximation is not adequate. If the mesh is refined or the order of the polynomial is increased from second order to third, then the numerical surface tension in high curvature regions can be reduced. But PPIC (piecewise parabolic interface calculation) are found to be more expensive than PLIC methods (costing more CPU time) [16].

Chapter 3

Interface Advection

There are two basic tasks to be solved by the interface advecting method [7]. First, the interfaces need to be reconstructed by using the given values of VOF in elements. Second, the VOF in each element needs to be updated using data from the velocity field and interface representation.

The template for our PLIC implementation, is based on Youngs' method [18]. When the VOF and velocity field of fluid are given, the template for each element in the computation domain is the following:

1. Calculate the outward normal to the interface. The interface orienta-

tion is estimated based on the values of VOF on a 27-point stencil. A new co-ordinate system (x', y', z') is created depended on the advection direction. The co-ordinate system (x', y', z') is rotated from global coordinates system, so that $+x'$ is always the advection direction.

2. Transform the cell to a unit cube in a local co-ordinate system $(\hat{x}, \hat{y}, \hat{z})$. The cube here is parallel to the (x', y', z') but scaled to a unit cube.
3. Transform the unit cube to co-ordinate system (x_1, x_2, x_3) , such that the vertex $(0,0,0)$ of the unit cube is the vertex closest to the interface and direction cosines n_1, n_2, n_3 are associated with axes x_1, x_2 and x_3 . In this co-ordinate system, the distance from $(0, 0, 0)$ to the interface is found according to the interface type. Thus the position of the interface is adjusted to match the VOF. This is known as *VOF matching*.
4. Compute volume fluxes advected from the donor cell to the acceptor cell based on interface type. The co-ordinate system (x'_1, x'_2, x'_3) is used, such that the face $+x$ or $-x$ is the exit face $(+x)$ of the donor element.
5. Advect the volume of fluid from donor to acceptor cell for this advection

direction using Strang splitting.

Cells are flagged as interface cells if $\eta \leq f \leq 1 - \eta$, where η is some small number(e.g., 10^{-8}). Those cells which are adjacent to the above cells are also activated as interface cells.

3.1 Interface construction

The piecewise linear interface approximation in each element is computed based on VOF values in a $3 \times 3 \times 3$ neighborhood. The interface equation is:

$$n_x x + n_y y + n_z z = d \quad (3.1)$$

where n_x , n_y , and n_z are the direction cosine components of outward unit normal to the plane. d is the distance of the closest vertex of the cell to the plane interface.

3.1.1 Calculation of the normal to the interface

The normal to the interface $\bar{n}$ uniquely determines the orientation of plane. The results of the PLIC method are found to be quite sensitive to the normal to the interface $\bar{n}$, so an accurate estimation of n often is very important

for good performance. A wide, symmetric stencil for $\bar{n}$ is necessary for a reasonable estimation of the interface orientation. In Youngs' paper [18], a 27-point stencil is used.

For each element, the VOF values of its $3 \times 3 \times 3$ neighborhood are found. Note that the following formula deals with the advection in +x direction only. i.e., f_{13} is always the VOF in the donor cell, and f_{14} is always the VOF in the acceptor cell. In order to deal with advection in other directions, a co-ordinate system (x', y', z') is rotated from global coordinates system and the VOF f_0 to f_{26} are reset (See Figures 3.1, 3.2, 3.3, 3.4, 3.5, 3.6).

The normal to the plane is defined as:

$$\bar{n} = -\frac{\nabla f}{|\nabla f|} \quad (3.2)$$

where ∇f is the gradient of VOF computed from the given VOF in a $3 \times 3 \times 3$ neighborhood,

$$\frac{\partial f}{\partial x} = \frac{f_E - f_W}{2\Delta x} \quad (3.3)$$

$$\frac{\partial f}{\partial y} = \frac{f_N - f_S}{2\Delta y} \quad (3.4)$$

$$\frac{\partial f}{\partial z} = \frac{f_T - f_B}{2\Delta z} \quad (3.5)$$

Figure 3.1: $3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in +x direction.

Figure 3.2: $3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in -x direction.

Figure 3.3: $3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in $+y$ direction.

Figure 3.4: $3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in -y direction.

Figure 3.5: $3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in +z direction.

Figure 3.6: $3 \times 3 \times 3$ neighborhood used for calculating the normal to the interface for advection in -z direction.

For a uniform mesh, the general form to calculate f_E is:

$$f_E = \frac{f_2 + f_8 + f_{20} + f_{26} + \beta(f_5 + f_{11} + f_{17} + f_{23}) + \gamma f_{14}}{4 + 4\beta + \gamma} \quad (3.6)$$

The general form to calculate f_W is:

$$f_W = \frac{f_0 + f_6 + f_{18} + f_{24} + \beta(f_3 + f_9 + f_{15} + f_{21}) + \gamma f_{12}}{4 + 4\beta + \gamma} \quad (3.7)$$

The general form to calculate f_N is:

$$f_N = \frac{f_6 + f_8 + f_{24} + f_{26} + \beta(f_7 + f_{15} + f_{17} + f_{25}) + \gamma f_{16}}{4 + 4\beta + \gamma} \quad (3.8)$$

The general form to calculate f_S is:

$$f_S = \frac{f_0 + f_2 + f_{18} + f_{20} + \beta(f_1 + f_9 + f_{11} + f_{19}) + \gamma f_{10}}{4 + 4\beta + \gamma} \quad (3.9)$$

The general form to calculate f_T is:

$$f_T = \frac{f_{18} + f_{20} + f_{24} + f_{26} + \beta(f_{19} + f_{21} + f_{23} + f_{25}) + \gamma f_{22}}{4 + 4\beta + \gamma} \quad (3.10)$$

The general form to calculate f_B is:

$$f_B = \frac{f_0 + f_2 + f_6 + f_8 + \beta(f_1 + f_3 + f_5 + f_7) + \gamma f_4}{4 + 4\beta + \gamma} \quad (3.11)$$

where β is chosen as 2 and γ is chosen as 4 according Youngs' paper [18].

3.1.2 Volume fraction matching

After orientation of the interface, i.e. after the normal to the interface $\bar{n}$ has been computed, the distance of the plane from the origin is computed to match the VOF in this cell.

If the origin of the cell is (x_o, y_o, z_o) , and the element size is $\delta x, \delta y, \delta z$, the cell occupies the volume:

$$\begin{aligned}x_o < x < x_o + \delta x \\y_o < y < y_o + \delta y \\z_o < z < z_o + \delta z\end{aligned}\tag{3.12}$$

The cell is transformed to a local co-ordinate system $(\hat{x}, \hat{y}, \hat{z})$ in which the origin of the system is the point (x_o, y_o, z_o) . The cell becomes a unit cube but still parallel to the (x', y', z') co-ordinate system.

The calculation of the interface distance is a nonlinear problem because the interface can intersect the boundary of the cell in 3, 4, 5 or 6 points and be a triangle, quadrilateral, pentagon or hexagon [7]. We try to follow the Youngs' method to solve each of these cases. Since any of the eight vertices of the cell could be closest to the interface in physical condition, a way

should be found to reduce the number of cases that need to be considered. First, the distance of point (0, 0, 0) from the plane through one the corners of the cell is calculated to determine which corner is the closest one to the interface. Second, if the VOF of the cell under consideration is greater than 1/2, the sign of the $\bar{n}$ is reversed and the interface position is adjusted to match the volume fraction: $1 - volumeOfFractionInCell$.

The numbering system of the eight corner of the cell is shown in Figure 3.7.

The distance of point (0, 0, 0) from the interface through one the corner of the cell is calculated as:

$$distance = \hat{n}_x \hat{x} + \hat{n}_y \hat{y} + \hat{n}_z \hat{z} \quad (3.13)$$

Where $\hat{n}_x$, $\hat{n}_y$, $\hat{n}_z$ are the component of the normal to the interface. $\hat{x}$, $\hat{y}$, $\hat{z}$ is relative to the co-ordinate of the corner under calculation.

Then

m_0 = value of corner corresponding to the smallest value of distance.

m_1 = value of corner corresponding to the second smallest value of distance.

m_2 = value of corner corresponding to the third smallest value of distance.

Figure 3.7: Local coordinates used in interface construction.

The new local co-ordinate (x_1, x_2, x_3) is used to match the volume fraction. thus the vertex $(0, 0, 0)$ of the unit cube is the closest to the interface and the direction cosines n_1, n_2, n_3 are associated with axes x_1, x_2, x_3 :

origin: p_0

x_1 axis: p_0p_1

x_2 axis: p_0p_2

x_3 axis: $p_0p'_3$

The calculation for the fourth corner of the $p_0p_1p_2$ is:

$$m_3 = m_1 + m_2 - m_0 \quad (3.14)$$

p'_3 is at the opposite end of the diagonal of the cube.

$$m'_3 = 9 + m_0 - m_1 - m_2 \quad (3.15)$$

The intercepts of the interface on the x_1, x_2 , and x_3 axes are:

$$h_1 = d/n_1$$

$$h_2 = d/n_2$$

$$h_3 = d/n_3 \quad (3.16)$$

where:

n_1 is the component of $\hat{n}$ along the x_1 axis

n_2 is the component of $\hat{n}$ along the x_2 axis

n_3 is the component of $\hat{n}$ along the x_3 axis

d is the distance of the point p_0 to the interface.

Since the origin of the new local co-ordinate system (x_1, x_2, x_3) is the closest vertex to the interface, n_1, n_2, n_3 are set to be the minimum, middle, and maximum values of $|\hat{n}_x|, |\hat{n}_y|, |\hat{n}_z|$, therefore $h_1 \geq h_2 \geq h_3$.

The distance of the point p_0 to the plane d is to match the volume fraction in the cell. The way of calculation the distance depends on the interface type.

A. Triangular Section (See Figure 3.8)

This situation occurs if:

$$6n_1n_2n_3f < n_1^3$$

$$f = \frac{1}{6}h_1h_2h_3 = \frac{d^3}{6n_1n_2n_3} \quad (3.17)$$

Therefore the distance from the origin point P_0 for triangular section is:

$$d = (6n_1n_2n_3)^{\frac{1}{3}} \quad (3.18)$$

B. Quadrilateral section A (See Figure 3.9)

This situation occurs if ¹:

¹There is an error on page 27 in Youngs's paper, it states $n_1^3 < 6n_1n_2n_3f <$

Figure 3.8: Triangular intersection.

$$n_1^3 < 6n_1n_2n_3f < n_2^3 - (n_2 - n_1)^3$$

$$\begin{aligned} f &= \frac{1}{6}h_1h_2h_3 \left\{ 1 - \left[\frac{h_1 - 1}{h_1} \right]^3 \right\} \\ &= \frac{1}{6n_1n_2n_3} \{ d^3 - (d - n_1)^3 \} \end{aligned} \quad (3.19)$$

Therefore the distance from the origin point P_0 for a quadrilateral section A is:

$$d = \frac{3n_1 + \sqrt{72n_2n_3f - 3n_1^2}}{6} \quad (3.20)$$

C. Pentagonal Section (See Figure 3.10)

This situation occurs if ² :

$$n_1 + n_2 > n_3 \text{ and } n_2^3 - (n_2 - n_1)^3 < 6n_1n_2n_3f < n_3^3 - (n_3 - n_1)^3 - (n_3 - n_2)^3$$

or

$$n_1 + n_2 < n_3 \text{ and } n_2^3 - (n_2 - n_1)^3 < 6n_1n_2n_3f < (n_1 + n_2)^3 - n_1^3 - n_2^3$$

$$f = \frac{1}{6}h_1h_2h_3 \left\{ 1 - \left[\frac{h_1 - 1}{h_1} \right]^3 - \left[\frac{h_2 - 1}{h_2} \right]^3 \right\}$$

$n_2^3 - (n_1 - n_2)^3$. But according to our calculation, it should be $n_1^3 < 6n_1n_2n_3f < n_2^3 - (n_2 - n_1)^3$

²There is an error on page 28 in Young's paper, it states that $n_2^3 - (n_1 - n_2)^3$. It is should be $n_2^3 - (n_2 - n_1)^3$.

Figure 3.9: Quadrilateral Section A.

$$= \frac{1}{6n_1n_2n_3} \{d^3 - (d - n_1)^3 - (d - n_2)^3\} \quad (3.21)$$

Therefore the distance from the origin point P_0 for a pentagonal section is computed by Newton-Raphson iteration.

D. Hexagonal Section (See Figure 3.11)

This situation occurs if:

$$n_1 + n_2 > n_3 \text{ and } 6n_1n_2n_3f > n_3^3 - (n_3 - n_1)^3 - (n_3 - n_2)^3$$

$$\begin{aligned} f &= \frac{1}{6}h_1h_2h_3 \left\{ 1 - \left[\frac{h_1 - 1}{h_1} \right]^3 - \left[\frac{h_2 - 1}{h_2} \right]^3 - \left[\frac{h_3 - 1}{h_3} \right]^3 \right\} \\ &= \frac{1}{6n_1n_2n_3} \{d^3 - (d - n_1)^3 - (d - n_2)^3 - (d - n_3)^3\} \end{aligned} \quad (3.22)$$

Therefore the distance from the origin point P_0 for hexagonal section is calculated by Newton-Raphson iteration.

E. Quadrilateral Section B (See Figure 3.12)

This situation occurs if:

$$n_1 + n_2 < n_3 \text{ and } 6n_1n_2n_3f > (n_1 + n_2)^3 - n_1^3 - n_2^3$$

$$f = \frac{1}{4}(p_1 + P_2 + p_3 + P_4) \quad (3.23)$$

Figure 3.10: Pentagonal Section.

Figure 3.11: Hexagonal Section.

Figure 3.12: Quadrilateral Section B.

Since:

$$\begin{aligned}
n_1 + p_1 n_3 &= d \\
n_1 + n_2 + p_2 n_3 &= d \\
n_2 + P_3 n_3 &= d \\
p_4 n_3 &= d
\end{aligned} \tag{3.24}$$

Therefore the distance from the origin point P_0 for quadrilateral section B is:

$$d = \frac{1}{2}(n_1 + n_2) + f n_3 \tag{3.25}$$

3.1.3 Volume Of liquid from the donor cell to the acceptor cell

Before the volume fraction advection step, the volume of liquid from donor cell to the acceptor cell based on interface type must be calculated. The co-ordinate system (x'_1, x'_2, x'_3) is used here, such that the face +x or -x is the exit face (+x in x', y', z' co-ordinate system) of the donor element.

In Figure 3.13:

V = volume of donor cell

δV = volume of liquid in the slice from the donor cell to the acceptor cell

= velocity on the exit face $\times$ face area $\times$ timeStepsize

$\epsilon = \delta V / V$

which is the distance from the plane drawn through the unit cell parallel to the exit face to the exit face. ϵ_1 is the volume of fluid in the slice over the volume of cell (beneath $A'BCDE'$) in Figure 3.13.

How to calculate the ϵ_1 depends on three things. What is the interface type? Which face is the exit face? What is the value of the slice box length ϵ ? To determine the interface type please see the previous section on volume fraction matching. As we know in the local (x_1, x_2, x_3) co-ordinate system, the origin point p_0 is the closest corner of the cell to the interface. We need to decide which face in the (x_1, x_2, x_3) co-ordinate system is the exit face (x-max) in the $(\hat{x}, \hat{y}, \hat{z})$ co-ordinate system. By taking advantage of the way of setting up the numbering system of the $(\hat{x}, \hat{y}, \hat{z})$ co-ordinate system (see Figure 3.7), we can identify the exit face by:

Figure 3.13: The Flux Across Cell.

$$m_2 + m_3' = 5 \quad \text{exit face is } x_1 \text{ max}$$

$$m_3' + m_1 = 5 \quad \text{exit face is } x_2 \text{ max}$$

$$m_1 + m_2 = 5 \quad \text{exit face is } x_3 \text{ max}$$

$$m_2 + m_3' = 13 \quad \text{exit face is } x_1 \text{ min}$$

$$m_3' + m_1 = 13 \quad \text{exit face is } x_2 \text{ min}$$

$$m_1 + m_2 = 13 \quad \text{exit face is } x_3 \text{ min}$$

A. Triangle section (see Figure 3.14)

Exit on Face A:

$$\begin{aligned} \epsilon_1 &= f \quad \text{if } \epsilon > p \\ \epsilon_1 &= f \left\{ 1 - \left[\frac{p - \epsilon}{p} \right]^3 \right\} \quad \text{if } \epsilon < p \end{aligned} \quad (3.26)$$

Exit on Face B³:

$$\begin{aligned} \epsilon_1 &= 0 \quad \text{if } \epsilon < 1 - p \\ \epsilon_1 &= f \left\{ \frac{\epsilon - 1 + p}{p} \right\}^3 \quad \text{if } \epsilon > 1 - p \end{aligned} \quad (3.27)$$

where the value of p is depend on the exit face:

$$p = \frac{d}{n_1} \quad \text{if exit face is } x_1 \text{ max or } x_1 \text{ min}$$

³There is an error on page 32 in Youngs' paper, it states: $\epsilon_1 = f \left\{ \frac{\epsilon - 1 - p}{p} \right\}^3$ and it should be $\epsilon_1 = f \left\{ \frac{\epsilon - 1 + p}{p} \right\}^3$

Figure 3.14: The Flux Across Cell: Triangular Section Case.

$$\begin{aligned}
 p &= \frac{d}{n_2} \quad \text{if exit face is } x_2 \text{ max or } x_2 \text{ min} \\
 p &= \frac{d}{n_3} \quad \text{if exit face is } x_3 \text{ max or } x_3 \text{ min}
 \end{aligned} \tag{3.28}$$

B. Quadrilateral section A (See Figure 3.15 and 3.16)

Case 1: The interface intersects the cell side as a triangle (See Figure 3.15).

Case 1A: Exit face x_1 max.

$$p = p_1 + \epsilon(p_2 - p_1)$$

$$\begin{aligned}
q &= q_1 + \epsilon(q_2 - q_1) \\
\epsilon_1 &= \frac{1}{6}\epsilon(p_1q_1 + pq_1 + pq)
\end{aligned} \tag{3.29}$$

The formula to calculate p_1, p_2, q_1, q_2 is the following (exit face x_1 max):

$$\begin{aligned}
p_1 &= \frac{d - n_1}{n_2} \\
p_2 &= \frac{d}{n_2} \\
q_1 &= \frac{d - n_1}{n_3} \\
q_2 &= \frac{d}{n_3}
\end{aligned} \tag{3.30}$$

Case 1B: Exit face x_1 min.

$$\begin{aligned}
\epsilon &= 1 - \epsilon \\
\epsilon_1 &= f - \epsilon_1 \quad \text{where } \epsilon_1 \text{ if found by exit face } x_1 \text{ max}
\end{aligned} \tag{3.31}$$

Case 2: The interface intersects the cell side as a quadrilateral (See Figure 3.16 exit face A):

p_1, p_2, q_1, q_2 is calculated by:

$$\begin{aligned}
p_1 &= \frac{d}{n_2} \\
p_2 &= \frac{d - n_1}{n_2} \\
q_1 &= \frac{d}{n_3}
\end{aligned}$$

Figure 3.15: The Flux Across Cell: Quadrilateral Section A Case 1.

Figure 3.16: The Flux Across Cell: Quadrilateral Section A Case 2 Case 3.

$$q_2 = \frac{d - n_1}{n_3} \quad (3.32)$$

if the exit face is x_2 min:

$$\begin{aligned} \epsilon_1 &= f && \text{if } \epsilon > p_1 \\ \epsilon_1 &= f - \frac{1}{6} \frac{(p_1 - \epsilon)^3 q_1}{p_1(p_1 - p_2)} && \text{if } p_1 > \epsilon > p_2 \\ \epsilon_1 &= \frac{1}{4} \epsilon \left\{ q_1 \frac{(2p_1 - \epsilon)}{p_1} + q_2 \frac{(2p_2 - \epsilon)}{p_2} \right\} && \text{if } p_2 > \epsilon \end{aligned} \quad (3.33)$$

if the exit face is x_3 min:

$$\begin{aligned} \epsilon_1 &= f && \text{if } \epsilon > q_1 \\ \epsilon_1 &= f - \frac{1}{6} \frac{(q_1 - \epsilon)^3 p_1}{q_1(q_1 - q_2)} && \text{if } q_1 > \epsilon > q_2 \\ \epsilon_1 &= \frac{1}{4} \epsilon \left\{ p_1 \frac{(2q_1 - \epsilon)}{q_1} + p_2 \frac{(2q_2 - \epsilon)}{q_2} \right\} && \text{if } q_2 > \epsilon \end{aligned} \quad (3.34)$$

Case 3: the interface does not intersect the cell side (See Figure 3.16 exit face on B):

p_1, p_2, q_1, q_2 is calculated by:

$$\begin{aligned} p_1 &= \frac{d}{n_2} \\ p_2 &= \frac{d - n_1}{n_2} \\ q_1 &= \frac{d}{n_3} \end{aligned}$$

$$q_2 = \frac{d - n_1}{n_3} \quad (3.35)$$

If exit face is x_2 max:

$$\begin{aligned} \epsilon_1 &= 0 && \text{if } \epsilon < 1 - p_1 \\ \epsilon_1 &= \frac{1}{6} \frac{(p_1 + \epsilon - 1)^3 q_1}{p_1(p_1 - p_2)} && \text{if } 1 - p_1 < \epsilon < 1 - p_2 \\ \epsilon_1 &= f - \frac{1 - \epsilon}{4} \left\{ q_1 \frac{(2p_1 + \epsilon - 1)}{p_1} + q_2 \frac{(2p_2 + \epsilon - 1)}{p_2} \right\} && \text{if } 1 - p_2 < \epsilon \quad (3.36) \end{aligned}$$

If exit face is x_3 max:

$$\begin{aligned} \epsilon_1 &= 0 && \text{if } \epsilon < 1 - q_1 \\ \epsilon_1 &= \frac{1}{6} \frac{(q_1 + \epsilon - 1)^3 p_1}{q_1(q_1 - q_2)} && \text{if } 1 - q_1 < \epsilon < 1 - q_2 \\ \epsilon_1 &= f - \frac{1 - \epsilon}{4} \left\{ p_1 \frac{(2q_1 + \epsilon - 1)}{q_1} + p_2 \frac{(2q_2 + \epsilon - 1)}{q_2} \right\} && \text{if } 1 - q_2 < \epsilon \quad (3.37) \end{aligned}$$

C. Pentagonal Section (See Figure 3.17):

p, q, r, s, h is found by:

$$\begin{aligned} p &= \frac{d - n_1}{n_3} \\ q &= \frac{d - n_1}{n_2} \\ r &= \frac{d - n_2}{n_3} \\ s &= \frac{d - n_2}{n_2} \\ h &= \frac{d}{n_3} \end{aligned} \quad (3.38)$$

Figure 3.17: The Flux Across Cell: Pentagonal Section Case.

If exit face is x_1 max:

$$\begin{aligned}
\epsilon_1 &= \frac{1}{6}pq\epsilon(1 + \alpha + \alpha^2) \quad \text{if } \epsilon < 1 - s \\
\alpha &= 1 + \frac{\epsilon(h-p)}{p} \\
\epsilon_1 &= f - \frac{1}{4}(1 - \epsilon)\{r + h + (s + \epsilon - 1)\frac{r}{s} \\
&\quad + p + \epsilon(h - p)\} \quad \text{if } \epsilon > 1 - s \quad (3.39)
\end{aligned}$$

If exit face is x_1 min:

$$\begin{aligned}
\epsilon &= 1 - \epsilon \\
\epsilon_1 &= f - \epsilon_1 \quad \text{which is found by exit face } x_1 \text{ max} \quad (3.40)
\end{aligned}$$

If exit face is x_2 max:

$$\begin{aligned}
\epsilon_1 &= \frac{1}{6}rs\epsilon(1 + \alpha + \alpha^2) \quad \text{if } \epsilon < 1 - q \\
\alpha &= 1 + \frac{\epsilon(h-r)}{r} \\
\epsilon_1 &= f - \frac{1}{4}(1 - \epsilon)\{p + h + (q + \epsilon - 1)\frac{p}{q} \\
&\quad + r + \epsilon(h - r)\} \quad \text{if } \epsilon > 1 - q \quad (3.41)
\end{aligned}$$

If exit face is x_2 min:

$$\begin{aligned}
\epsilon &= 1 - \epsilon \\
\epsilon_1 &= f - \epsilon_1 \quad \text{which is found by exit face } x_2 \text{ max} \quad (3.42)
\end{aligned}$$

If exit face is x_3 max:

When $p > r$

$$\begin{aligned}
\epsilon_1 &= 0 && \text{if } \epsilon < 1 - h \\
\epsilon_1 &= \frac{1}{6} \frac{(h + \epsilon - 1)^3}{(h - p)(h - r)} && \text{if } 1 - h < \epsilon < 1 - p \\
\epsilon_1 &= \frac{1}{6} \frac{(h + \epsilon - 1)^2}{h - r} (1 + \alpha + \alpha^2) && \text{if } 1 - p < \epsilon < 1 - r \\
&&& \text{with } \alpha = \frac{p + \epsilon - 1}{h + \epsilon - 1} \\
\epsilon_1 &= \frac{1}{6} \frac{(h + \epsilon - 1)^3}{(h - p)(h - r)} (1 - \alpha^3 - \beta^3) && \text{if } 1 - r < \epsilon \\
&&& \text{with } \alpha = \frac{p + \epsilon - 1}{h + \epsilon - 1} \\
&&& \text{with } \beta = \frac{r + \epsilon - 1}{h - \epsilon - 1} \tag{3.43}
\end{aligned}$$

When $p < r$:

$$\begin{aligned}
\epsilon_1 &= 0 && \text{if } \epsilon < 1 - h \\
\epsilon_1 &= \frac{1}{6} \frac{(h + \epsilon - 1)^3}{(h - p)(h - r)} && \text{if } 1 - h < \epsilon < 1 - r \\
\epsilon_1 &= \frac{1}{6} \frac{(h + \epsilon - 1)^2}{h - p} (1 + \alpha + \alpha^2) && \text{if } 1 - r < \epsilon < 1 - p \\
&&& \text{with } \alpha = \frac{r + \epsilon - 1}{h + \epsilon - 1} \\
\epsilon_1 &= \frac{1}{6} \frac{(h + \epsilon - 1)^3}{(h - p)(h - r)} (1 - \alpha^3 - \beta^3) && \text{if } 1 - p < \epsilon \\
&&& \text{with } \alpha = \frac{r + \epsilon - 1}{h + \epsilon - 1} \\
&&& \text{with } \beta = \frac{p + \epsilon - 1}{h - \epsilon - 1} \tag{3.44}
\end{aligned}$$

If exit face is x_3 min:

$$\epsilon = 1 - \epsilon$$

$$\epsilon_1 = f - \epsilon_1 \quad \text{which is found by exit face } X_3 \text{ max} \quad (3.45)$$

D. Hexagonal Section (See Figure 3.18):

p, q, u, t, s, r is found by:

$$\begin{aligned} p &= \frac{d - n_2}{n_3} \\ q &= \frac{d - n_2}{n_1} \\ u &= \frac{d - n_1}{n_3} \\ t &= \frac{d - n_1}{n_2} \\ s &= \frac{d - n_3}{n_1} \\ r &= \frac{d - n_3}{n_2} \end{aligned} \quad (3.46)$$

Figure 3.18: The Flux Across Cell: Hexagonal Section Case.

If the exit face is x_1 max:

$$\begin{aligned}
\epsilon_1 &= \frac{1}{6}ut\epsilon(1 + \alpha + \alpha^2) && \text{if } \epsilon < 1 - q \\
&&& \text{with } \alpha = 1 + \epsilon \frac{(1-u)}{u(1-s)} \\
\epsilon_1 &= \frac{1}{6}ut(1 - q)(1 + \alpha + \alpha^2) + \\
&\quad \frac{1}{4}(p_1 + p_2 + p_3)(q + \epsilon - 1) && \text{if } 1 - q < \epsilon < 1 - s \\
&&& \text{with } p_1 = \frac{p}{q}(\epsilon + q - 1) \\
&&& \text{with } p_2 = u + \frac{(1-q)(1-u)}{1-s} \\
&&& \text{with } p_3 = u + \frac{\epsilon(1-u)}{1-s} \\
&&& \text{with } \alpha = \frac{p_2}{u} \\
\epsilon_1 &= f - (1 - \epsilon - V) && \text{if } 1 - s < \epsilon \\
&&& \text{with } V = \frac{1}{6}(1 - p)(1 - r) \\
&&& (1 - \epsilon)(1 + \alpha + \alpha^2) \\
&&& \text{with } \alpha = 1 + \frac{(1-\epsilon)p}{q(1-p)} \quad (3.47)
\end{aligned}$$

If the exit face is x_1 min:

$$\epsilon = 1 - \epsilon$$

$$\epsilon_1 = f - \epsilon_1 \quad \text{which is found by exit face } x_1 \text{ max} \quad (3.48)$$

If the exit face is x_2 max:

$$\begin{aligned} \epsilon_1 &= \frac{1}{6}pq\epsilon(1 + \alpha + \alpha^2) && \text{if } \epsilon < 1 - t \\ &&& \text{with } \alpha = 1 + \epsilon \frac{(1-p)}{p(1-r)} \\ \epsilon_1 &= \frac{1}{6}pq(1-t)(1 + \alpha + \alpha^2) + \\ &\quad \frac{1}{4}(p_1 + p_2 + p_3)(t + \epsilon - 1) && \text{if } 1 - t < \epsilon < 1 - r \\ &&& \text{with } p_1 = \frac{u}{t}(\epsilon + t - 1) \\ &&& \text{with } p_2 = p + \frac{(1-t)(1-p)}{1-r} \\ &&& \text{with } p_3 = p + \frac{\epsilon(1-p)}{1-r} \\ &&& \text{with } \alpha = \frac{p_2}{p} \\ \epsilon_1 &= f - (1 - \epsilon - V) && \text{if } 1 - r < \epsilon \\ &&& \text{with } V = \frac{1}{6}(1 - u)(1 - s) \\ &&& (1 - \epsilon)(1 + \alpha + \alpha^2) \\ &&& \text{with } \alpha = 1 + \frac{(1-\epsilon)u}{t(1-u)} \end{aligned} \quad (3.49)$$

If the exit face is x_2 min:

$$\epsilon = 1 - \epsilon$$

$$\epsilon_1 = f - \epsilon_1 \quad \text{which is found by exit face } x_2 \text{ max} \quad (3.50)$$

If the exit face is x_3 max:

$$\epsilon_1 = \frac{1}{6}sr\epsilon(1 + \alpha + \alpha^2) \quad \text{if } \epsilon < 1 - u$$

$$\text{with } \alpha = 1 + \epsilon \frac{(1-s)}{s(1-u)}$$

$$\epsilon_1 = \frac{1}{6}sr(1-p)(1 + \alpha + \alpha^2) +$$

$$\frac{1}{4}(p_1 + p_2 + p_3)(p + \epsilon - 1) \quad \text{if } 1 - u < \epsilon < 1 - p$$

$$\text{with } p_1 = \frac{q}{p}(\epsilon + p - 1)$$

$$\text{with } p_2 = s + \frac{(1-p)(1-s)}{1-u}$$

$$\text{with } p_3 = s + \frac{\epsilon(1-s)}{1-u}$$

$$\text{with } \alpha = \frac{p_2}{s}$$

$$\epsilon_1 = f - (1 - \epsilon - V) \quad \text{if } 1 - p < \epsilon$$

$$\text{with } V = \frac{1}{6}(1 - q)(1 - t)$$

$$(1 - \epsilon)(1 + \alpha + \alpha^2)$$

$$\text{with } \alpha = 1 + \frac{(1-\epsilon)q}{p(1-q)} \quad (3.51)$$

If the exit face is x_3 min:

$$\epsilon = 1 - \epsilon$$

$$\epsilon_1 = f - \epsilon_1 \quad \text{which is found by exit face } x_3 \text{ max} \quad (3.52)$$

E. Quadrilateral Section B (See Figure 3.19):

p_1, p_2, p_3, p_4 is calculated by:

$$\begin{aligned} p_1 &= \frac{d - n_1}{n_3} \\ p_2 &= \frac{d - n_1 - n_2}{n_3} \\ p_3 &= \frac{d - n_2}{n_3} \\ p_4 &= \frac{d}{n_3} \end{aligned} \quad (3.53)$$

If exit face is x_1 max:

$$\begin{aligned} p_2' &= p_2 + \epsilon(p_3 - p_2) \\ p_3' &= p_1 + \epsilon(p_4 - p_1) \\ \epsilon_1 &= \frac{1}{4}\epsilon(p_2 + p_1 + p_2' + p_3') \end{aligned} \quad (3.54)$$

If exit face is x_1 min:

$$\epsilon = 1 - \epsilon$$

$$\epsilon_1 = f - \epsilon_1 \quad \text{which is found by exit face } x_1 \text{ max} \quad (3.55)$$

Figure 3.19: The Flux Across Cell: Quadrilateral Section B Case.

If exit face is x_2 max:

$$\begin{aligned}
 p_2' &= p_2 + \epsilon(p_1 - p_2) \\
 p_3' &= p_3 + \epsilon(p_4 - p_3) \\
 \epsilon_1 &= \frac{1}{4}\epsilon(p_2 + p_3 + p_2' + p_3') \quad (3.56)
 \end{aligned}$$

If exit face is x_2 min:

$$\begin{aligned}
 \epsilon &= 1 - \epsilon \\
 \epsilon_1 &= f - \epsilon_1 \quad \text{which is found by exit face } x_2 \text{ max} \quad (3.57)
 \end{aligned}$$

If exit face is x_3 max:

$$\begin{aligned}
 \epsilon_1 &= \epsilon \quad \text{if } \epsilon < p_2 \\
 \epsilon_1 &= \epsilon - \frac{1}{6} \frac{(\epsilon - p_2)^3}{(p_1 - p_2)(p_3 - p_2)} \quad \text{if } p_2 < \epsilon < p_3 \\
 \epsilon_1 &= \epsilon - \frac{1}{6} \frac{\alpha^2}{p_1 - p_2} \left\{ 1 + \frac{\beta(\alpha + \beta)(p_1 - p_2)}{(p_4 - p_3)\alpha^2} \right\} \quad \text{if } p_3 < \epsilon < p_1 \\
 &\quad \text{with } \alpha = \epsilon - p_2 \\
 &\quad \text{with } \beta = \epsilon - p_3 \\
 \epsilon_1 &= f - \frac{1}{6} \frac{(p_4 - \epsilon)^3}{(p_4 - p_1)(p_4 - p_3)} \quad \text{if } p_1 < \epsilon < p_4 \\
 \epsilon_1 &= f \quad \text{if } \epsilon < p_4 \quad (3.58)
 \end{aligned}$$

If exit face is x_3 min:

$$\epsilon = 1 - \epsilon$$

$$\epsilon_1 = f - \epsilon_1 \quad \text{which is found by exit face } x_3 \text{ max} \quad (3.59)$$

3.2 Interface Advection

Let the size of the element to be:

δx = element length in x direction

δy = element length in y direction

δz = element length in z direction

Then the volume of fluid in the cell is:

$$f \delta x \delta y \delta z \quad (3.60)$$

The advection of interface is calculated by:

$$\frac{\partial \rho}{\partial t} + \bar{u} \nabla \rho = 0 \quad (3.61)$$

where ρ is density of the liquid.

The center of the cell is $(i + 1/2, j + 1/2, k + 1/2)$, thus

δV_i^- is the volume of the slice box on the -x face:

$$\delta V_{i,j+1/2,k+1/2}^- = u_{i,j+1/2,k+1/2} \delta y \delta z \delta t \quad (3.62)$$

δV_{i+1}^- is the volume of the slice box on the +x face:

$$\delta V_{i+1,j+1/2,k+1/2}^- = u_{i+1,j+1/2,k+1/2} \delta y \delta z \delta t \quad (3.63)$$

δV_j^- is the volume of the slice box on the -y face:

$$\delta V_{i+1/2,j,k+1/2}^- = u_{i+1/2,j,k+1/2} \delta x \delta z \delta t \quad (3.64)$$

δV_{j+1}^- is the volume of the slice box on the +y face:

$$\delta V_{i+1/2,j+1,k+1/2}^- = u_{i+1/2,j+1,k+1/2} \delta x \delta z \delta t \quad (3.65)$$

δV_k^- is the volume of the slice box on the -z face:

$$\delta V_{i+1/2,j+1/2,k}^- = u_{i+1/2,j+1/2,k} \delta x \delta y \delta t \quad (3.66)$$

δV_{k+1}^- is the volume of the slice box on the +z face:

$$\delta V_{i+1/2,j+1/2,k+1}^- = u_{i+1/2,j+1/2,k+1} \delta x \delta y \delta t \quad (3.67)$$

The initial volume of a cell when starting an advection time step is:

$$V^0 = \delta x \delta y \delta z \quad (3.68)$$

The final volume of a cell after completing an advection time step is:

$$V^1 = V^0 + \delta V_i - \delta V_{i+1} + \delta V_j - \delta V_{j+1} + \delta V_k - \delta V_{k+1} \quad (3.69)$$

For incompressible flow (div free velocity field):

$$\delta V_i - \delta V_{i+1} + \delta V_j - \delta V_{j+1} + \delta V_k - \delta V_{k+1} = 0 \quad (3.70)$$

Therefore

$$V^1 = V^0 \quad (3.71)$$

Note that in order to reduce error, the advection order is $x \mapsto y \mapsto z$ and $z \mapsto y \mapsto x$ on alternate time steps.

The first step of operator splitting is to update the volume of the cell.

For $x \mapsto y \mapsto z$ advection case:

$$\begin{aligned} V^1 \mapsto V^x &= V^1 + \delta V_i - \delta V_{i+1} \quad \text{x - rezone} \\ \mapsto V^y &= V^x + \delta V_j - \delta V_{j+1} \quad \text{y - rezone} \\ \mapsto V^z &= V^y + \delta V_k - \delta V_{k+1} \quad \text{z - rezone} \\ &= V^0 \end{aligned} \quad (3.72)$$

After x-rezone, V^x is stored with the element and after an y-rezone, V^y is stored with the element and similarly for a z-rezone.

The second step of operator splitting is to update the volume of fluid in each cell.

$$\begin{aligned}
f^x V^x &= f^0 V^1 + h_i \delta V_i - h_{i+1} \delta V_{i+1} \quad \text{x - rezone} \\
f^y V^y &= f^x V^x + h_j \delta V_j - h_{j+1} \delta V_{j+1} \quad \text{y - rezone} \\
f^z V^z &= f^y V^y + h_k \delta V_k - h_{k+1} \delta V_{k+1} \quad \text{z - rezone} \\
&= f^1 V^0 \tag{3.73}
\end{aligned}$$

Where

f^x ($0 \leq f^x \leq 1$) is the fraction of fluid in a cell with volume V^x .

f^y ($0 \leq f^y \leq 1$) is the fraction of fluid in a cell with volume V^y .

f^z ($0 \leq f^z \leq 1$) is the fraction of fluid in a cell with volume V^z .

f^1 ($0 \leq f^1 \leq 1$) is the final solution of vof for advection in this time step.

Therefore,

$f^x V^x$ is the volume of fluid (m^3) after x - rezone,

$f^y V^y$ is the volume of fluid (m^3) after y - rezone,

$f^z V^z$ is the volume of fluid (m^3) after z - rezone.

The values of h (volume fraction of fluid in the volume of slices) are found by using the volume fractions available at the beginning of each rezone step. The value of ϵ used in volume of fraction matching is computed by:

$$\begin{aligned}
\epsilon_i &= \frac{|\delta V_i|}{V_d^1} \\
\epsilon_j &= \frac{|\delta V_j|}{V_d^x} \\
\epsilon_k &= \frac{|\delta V_k|}{V_d^y}
\end{aligned} \tag{3.74}$$

Where d stands for the donor cell.

Chapter 4

Surface tension on free surfaces

In this chapter we describe a method for modeling interfacial surface tension, which is the force per unit area arising from local interface curvature and local tangential variations in the surface tension coefficient. In continuum surface tension formulations, interfacial surface phenomena are replaced with smoothly varying volumetric forces derived by integrating surface tension per unit area over the interface surface. Surface tension is expressed as a volumetric force $\mathbf{F}_s$ satisfying

$$\mathbf{F}_s(\mathbf{x}_{s,t}) = \int_{\Omega} \mathbf{f}_{s,t}(\mathbf{x}) \delta(\mathbf{x} - \mathbf{x}_{s,t}) dS \quad (4.1)$$

where $\mathbf{x}_s$ are the points on the interface, Ω is a surface in three dimensions, $\mathbf{f}_{s,t}$ is surface tension force per unit interfacial area, and $\delta(\mathbf{x})$ is a delta function.

4.1 The surface tension force on a triangulated curved surface

The surface tension force per unit interfacial area, $\mathbf{f}_{s,t}$, is given by [1]:

$$\mathbf{f}_{s,t} = \sigma \kappa \hat{\mathbf{n}} + \nabla_s \sigma \quad (4.2)$$

where σ is the surface tension coefficient, ∇_s is the surface gradient, given by [1]:

$$\nabla_s = (\hat{\mathbf{i}} - \hat{n}_x \hat{\mathbf{n}}) \partial_x + (\hat{\mathbf{j}} - \hat{n}_y \hat{\mathbf{n}}) \partial_y + (\hat{\mathbf{k}} - \hat{n}_z \hat{\mathbf{n}}) \partial_z \quad (4.3)$$

where $\hat{\mathbf{n}}$ is the interface unit normal, and κ is the mean interfacial curvature:

$$\kappa = -\nabla \cdot \hat{\mathbf{n}} \quad (4.4)$$

$\sigma \kappa \hat{\mathbf{n}}$, the first term in Eqn.4.2, is a force acting normal to the interface.

proportional to the curvature κ . $\nabla_s \sigma$, the second term in Eqn.4.2, is a force acting along the interface (tangentially) toward regions with higher surface tension coefficient values. The normal force tends to smooth and propagate regions of high curvature, whereas the tangential force tends to force fluid along the interface toward regions of higher σ . It is obvious that when the surface gradient is equal to zero, the tangential force is equal to zero too. In most studies, only normal force is modeled, i.e., the surface tension coefficient σ is taken to be constant.

When discontinuous interfaces are represented as finite thickness transition regions within which the VOF varies smoothly from zero to one, the normal of interface is usually calculated by:

$$\hat{\mathbf{n}} = \frac{\nabla f}{\|\nabla f\|} \quad (4.5)$$

Therefore, although considerable progress has been made in calculating the surface force on a curved surface, most of them are involved in using second derivatives, which very easily cause numerical problems (ill-behaved: discontinuous and/or singular). In order to avoid this, we try to use integration to compute the surface force, instead, based on Keith Miller's method

Figure 4.1: Definition of tangent vector t .

[12]. The key point of Miller's theory is that a force normal to a surface with magnitude equal to the curvature is equivalent to a surface tension force applied to the boundary of the surface. Now, consider that we have a piecewise linear surface consisting of triangular facets, then we could use the following integral identity:

$$\int_{\Omega} \kappa \hat{n} dS = \int_{\partial\Omega} t dl \quad (4.6)$$

That is, the integral of the sum of the principal curvatures κ , times the unit normal vector $\hat{n}$ over a piece of surface Ω , is equal to the line integral around the boundary with integrand equal to the unit tangent vector t , where t is defined to be (i) tangential to Ω , (ii) orthogonal to $d\mathbf{l}$, and (iii) pointing away from Ω [12] (see Figure 4.1).

The curvature on the piecewise linear surface could be treated as a dis-

Figure 4.2: m nodes share node i in counterclockwise order.

tribution, which means the curvature only exists on the edges of the facets and is zero elsewhere. The equation 4.6 can be modified by multiplying the integrand by a “hat function” α , which is unity at the i ’th node and decays to zero at its neighbors:

$$\int_{nbd\ i} \kappa \mathbf{n} \alpha dS = \frac{1}{2} \int_{\partial(nbd\ i)} \mathbf{t} dl \quad (4.7)$$

In order to write down the discrete form of 4.7, some notation needs to be introduced. Node i is named as $\mathbf{x}_0$, and those nodes $1, \dots, m$ which share node i in counterclockwise order are named as $\mathbf{x}_1, \dots, \mathbf{x}_m$ (see Figure 4.2).

We can define

1. The length vector of the triangle:

$$\mathbf{r}_j \equiv \mathbf{x}_j - \mathbf{x}_0, \quad (4.8)$$

$$\mathbf{l}_j \equiv \mathbf{x}_{j+1} - \mathbf{x}_j, \quad (4.9)$$

2. The area vector of the triangle, which is the cross product of the length vector r_j and L_j :

$$\mathbf{A}_j \equiv \frac{1}{2} \mathbf{r}_j \times \mathbf{x}_j, \quad (4.10)$$

3. The normalized area vector:

$$\hat{\mathbf{n}}_j \equiv \frac{\mathbf{A}_j}{\|\mathbf{A}_j\|}, \quad (4.11)$$

4. The tangential vector of the triangle, which is the normalized cross product of the normal area vector n_j and length vector L_j :

$$\mathbf{t}_j \equiv -\frac{\mathbf{n}_j \times \mathbf{l}_j}{\|\hat{\mathbf{n}}_j \times \mathbf{l}_j\|}, \quad (4.12)$$

where $\mathbf{A}_j$ is the area vector of the j 'th triangle surrounding node i . which has length equal to the area of the triangle and direction normal to the triangle.

With this notation, the surface tension force applied to the boundary of the surface is equal to (or Eqn. 4.7 is equivalent to):

$$\int_{nbdi} \kappa \hat{\mathbf{n}} \alpha dS = \frac{1}{2} \sum_{j=1}^m \mathbf{t}_j \|\mathbf{l}_j\| \quad (4.13)$$

4.2 Triangulated algorithm on the surface

In order to use Eqn. 4.13, we need to create a triangular mesh on the surface.

The following is the triangular algorithm we use.

1. For triangular mesh data information, the coordinates (x, y, z) for triangle nodes on the surface are stored in a double array and for each triangle the three nodes are stored in CCW order, i.e. the three nodes in the i 'th triangle are $\{a, b, c\}$. Based on this information, the list of triangles that share the same node are also found.
2. For each node in the list of triangles, first, find each triangle that has this node and store the pair of nodes on the “far” edge into a edge list. Second, find a CCW edge list which contains, first the node itself, then all the other nodes in the triangles sharing this node in CCW order.
3. The three components of the surface force at each node are calculated based on Keith Miller’s paper. The result is stored in the `surfaceTensionArray`.
 - A. Let x_0 be the coordinates of the node i on which we wish to compute the force.

- B. Get all triangles that share this node. For each triangle get the edge from x_0 to the next node, say j with coordinates x_j . Then calculate the length vector from x_0 to x_j by Eqn. 4.8.
- C. Next, get the last node in this triangle, say $j+1$ and compute the length vector the edge of this triangle that does not contain node i by Eqn. 4.9.
- D. Now compute the area vector, normalized area vector. and tangential vector according to Eqn. 4.10 to Eqn. 4.12 and compute the force vector from this triangle according Eqn. 4.13.
- E. Sum the contributions of the force vectors from all triangles on this node

4.3 Peskin's Delta Function algorithm

Ideally the surface tension force $\mathbf{F}_s(\mathbf{x}_s)$ is discontinuous since the interface along $\mathbf{x}_s$ is discontinuous. Interfaces are instead represented in the computational domain as *finite thickness transition regions* within which the VOF smoothly varies from zero to one. The surface tension force $\mathbf{F}_s(\mathbf{x}_s)$ must therefore also vary smoothly, being nonzero only within these transition re-

gions. Therefore, the discrete delta function should act as a smoother, which smooths the surface tension force within and near to the interface transition region. Thus the discrete numerical form used for the interface delta function is very important. It is found that the quality of the model relies heavily on the form used for the delta function.

In our surface tension model, Peskin's kernel $\delta(\mathbf{x})$ is used, which is a smoothed approximation of the three-dimensional Dirac δ - function [15]:

$$\delta(\mathbf{x}) = h^{-3} \phi\left(\frac{x_1}{h}\right) \phi\left(\frac{x_2}{h}\right) \phi\left(\frac{x_3}{h}\right) \quad (4.14)$$

where $\mathbf{x} = (x_1, x_2, x_3)$, h is element size, and ϕ has following properties:

1. ϕ is a continuous function.
2. $\phi(r) = 0$ for $|r| \geq 2$.
3. For all r ,

$$\sum_{j \text{ even}} \phi(r - j) = \sum_{j \text{ odd}} \phi(r - j) = \frac{1}{2} \quad (4.15)$$

4. For all r ,

$$\sum_j (r - j) \phi(r - j) = 0 \quad (4.16)$$

5. For all r ,

$$\sum_j (\phi(r-j))^2 = C \quad (4.17)$$

where C is independent of r .

Continuity of ϕ implies continuity of $\delta - function$. This avoids sudden changes as the free surface moves in space. The delta function derived by Peskin [15] is:

$$\phi(r) = \begin{cases} \frac{3-2|r|+\sqrt{1+4|r|-4r^2}}{8} & |r| \leq 1 \\ \frac{5-2|r|-\sqrt{-7+12|r|-4r^2}}{8} & 1 \leq |r| \leq 2 \\ 0 & \text{otherwise} \end{cases} \quad (4.18)$$

ϕ is an even function, and not only ϕ but also its derivatives are continuous. The function ϕ is plotted in Figure 4.3.

A Tcl script was written, which converts the forces per unit area on the interface into smoothly varying volumetric forces using the above delta function.

1. Get co-ordinate (x, y, z) for each node, we can call it nodeG, which stands for the nodes on the grid. NodeG is represented by indices (ijk), for example, $nodeG_{123}$ means this node is at the first column

Figure 4.3: One-dimensional delta function.

in the x-direction, the second column in the y-direction, and the third column in the z-direction.

2. Specify the co-ordinate (u, v, w) for each node on triangles on the free surface, we can call it nodeT, which stands for the nodes on triangles.
3. Each nodeT on triangle, can act on at most $4 \times 4 \times 4 = 64$ nodeGs. For each nodeT, get nodeG. For each nodeG, compute the delta function from the nodeG and add the force to that nodeG.

A. For each nodeT, find the $4 \times 4 \times 4 = 64$ nodeGs. which it acts on.

For those $4 \times 4 \times 4 = 64$ nodeGs:

The values of i could be:

$$\text{int}(u/dx) - 1, \text{int}(u/dx), \text{int}(u/dx) + 1, \text{int}(u/dx) + 2$$

we call it $i-1$, i , $i+1$, and $i+2$ respectively. where int means taking the integer part, u is the nodeT co-ordinate in the x-direction. dx is the element size in the x-direction.

The values of j could be:

$$\text{int}(v/dy) - 1, \text{int}(v/dy), \text{int}(v/dy) + 1, \text{int}(v/dy) + 2$$

we call it $j - 1$, j , $j + 1$, and $j + 2$ respectively. V is the nodeT

co-ordinate in the y-direction, dy is the element size in the y-direction.

The values of k could be:

$$int(w/dz) - 1, int(w/dz), int(w/dz) + 1, int(w/dz) + 2$$

we call it $k - 1$, k , $k + 1$, and $k + 2$ respectively. W is the nodeT co-ordinate in the z-direction, dz is the element size in the z-direction.

B. For each nodeG in $4 \times 4 \times 4 = 64$ nodeGs, find the distance (r_x, r_y, r_z) from the node to the interface.

For those nodeGs at the $i'th$ column in the x-direction:

$$r_x = remainder(u/dx) \quad (4.19)$$

For those nodeGs at the $i - 1'th$ column in the x-direction:

$$r_x = 1 + remainder(u/dx) \quad (4.20)$$

For those nodeGs at the $i + 1'th$ column in the x-direction:

$$r_x = 1 - remainder(u/dx) \quad (4.21)$$

For those nodeGs at the $i + 2'th$ column in the x-direction:

$$r_x = 2 - remainder(u/dx) \quad (4.22)$$

Where remainder means taking the decimal part.

Similarly r_y and r_z are computed.

C. Compute the 3D delta function.

Looping over $k - 1$, $j - 1$, $i - 1$, for each nodeG of 64 nodeGs for each direction, x, y, z, compute $\delta Function_X$, $\delta Function_Y$, and $\delta Function_Z$, respectively. e.g:

when $r_x \leq 1$:

$$\delta Function_X = 3.0 - 2.0r_x + \sqrt{1.0 + 4.0r_x - 4.0r_x^2}/8.0 \quad (4.23)$$

when $1 \leq r_x \leq 2$:

$$\delta Function_X = 5.0 - 2.0r_x + \sqrt{-7.0 + 12.0r_x + 4.0r_x^2}/8.0 \quad (4.24)$$

when $r_x > 2$:

$$\delta Function_X = 0.0 \quad (4.25)$$

Similarly $\delta Function_Y$ and $\delta Function_Z$ are computed.

Thus the delta function in 3-D is:

$$\begin{aligned} \delta Function_{3D} = & \delta Function_X * \delta Function_Y \\ & * \delta Function_Z / elementsize^3 \end{aligned} \quad (4.26)$$

D. Compute the forces according to the 3D delta function above.

Each nodeT can act on at most $4 \times 4 \times 4 = 64$ nodeGs. For each nodeT get the nodeGs. For each nodeG, compute the delta function and add the force to that nodeG

$$Force_x = force_u * deltaFunction_{3D} \quad (4.27)$$

$$Force_y = force_v * deltaFunction_{3D} \quad (4.28)$$

$$Force_z = force_w * deltaFunction_{3D} \quad (4.29)$$

where $force_u$, $force_v$, $force_w$ are the three component of surface tension force at nodeT in x-direction, y-direction, z-direction respectively and $Force_x$, $Force_y$, $Force_z$ are the three component of surface tension force at nodeG in x-direction, y-direction, z-direction respectively.

E. Sum all the forces on the nodeG. Since more than one nodeT on the triangle has contributed to the forces on the nodeG, therefore we should sum over all the force acting on this nodeG. Write the force into a double array .

Chapter 5

Numerical Results

The performance of modern interface tracking methods is commonly assessed by uniform translation and rigid body rotation tests. The method should pass translation and rotation tests with acceptable mass conservation and without significant distortion or degradation of fluid interfaces.

5.1 Rigid body motion of a sphere

Uniform and constant velocity fields are imposed on a sphere which is placed in the center of $20 \times 20 \times 20$ mesh, such that the sphere could translate along the x-direction, diagonally across the x- and y-direction, or diagonally across

the mesh (45° angle) respectively. The sphere should not change shape as the result of the movement.

5.1.1 VOF initialization

Let the center of sphere be (x_c, y_c, z_c) , therefore the distance of a point (x, y, z) to the surface of the sphere is:

$$D = \sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2} - r_0 \quad (5.1)$$

where

r_0 is the radius of the sphere.

For a given element, the maximum value of distance (D_{max}) to the surface of the sphere at the 8 corners and the minimum value of distance (D_{min}) to the surface of the sphere at the 8 corners are found:

If $D_{min} - 0.1\delta x > 0$, the whole cell is outside the fluid, where δx is the element size. The correction term $0.1\delta x$ is included to allow for the case when the surface intersects one of the faces of the cell but does not cross any of the cell corners [18].

Then set

$$f = 0 \quad (5.2)$$

If $D_{max} + 0.1\delta x < 0$, the whole cell is inside the fluid,

then set

$$f = 1 \quad (5.3)$$

Otherwise, the cell belongs to the fuzzy set ($0 < f < 1$)

If a cell is within the fuzzy set, the cell is divided into $10 \times 10 \times 10$ sub-cells.

For each sub-cell the D_{max} and the D_{min} are calculated.

If $D_{min} > 0$, the whole sub-cell is outside the fluid,

then

$$f_{subcell} = 0 \quad (5.4)$$

If $D_{max} < 0$, the whole sub-cell is inside the fluid,

then

$$f_{subcell} = 1.0 \quad (5.5)$$

Otherwise,

$$f_{subcell} = \frac{|D_{min}|}{D_{max} + |D_{min}|} \quad (5.6)$$

5.1.2 Sphere translation results

Sphere Geometry: Radius of Sphere: 0.25 m

Center of Sphere: (1.0, 1.0, 1.0)

Timestep	Total VOF(Exact Volume)	Total VOF(Calculated Volume)	Error %
0	65.473520604409273	65.473520604409273	0.000000000
20	65.473520604409273	65.473520604409231	0.65114e-14
30	65.473520604409273	65.473520604409106	2.60456e-13
40	65.473520604409273	65.473520604409075	3.03866e-13

Table 5.1: The variation with time of the total VOF within the sphere during translation in the x-direction.

Mesh Data: Number of Elements: $20 \times 20 \times 20$

Mesh Element Size: 0.1 m

Simulation Parameters: Time Step size: 0.1 s

Time Step: 40

Velocity field:

Case 1: Constant velocity in m/s in the x-direction

$$(V_x, V_y, V_z) = (0.2, 0.0, 0.0)$$

The result obtained is shown in Figures 5.1 to 5.3 and the variation with time of the total VOF within the sphere is shown in Table 5.1.

Figure 5.1: Sphere translation in the x-direction (time step = 0).

Figure 5.2: Sphere translation in the x-direction (time step = 20).

Figure 5.3: Sphere translation in the x-direction (time step = 40).

Timestep	Total VOF(Exact Volume)	Total VOF(Calculated Volume)	Error %
0	65.473520604409273	65.473520604409273	0.000000000
20	65.473520604409273	65.473520604409236	6.51143e-14
30	65.473520604409273	65.473520604409115	2.38751e-13
40	65.473520604409273	65.473520604409065	3.25571e-13

Table 5.2: The variation with time of the total VOF within the sphere during the translation in the x and y-direction.

Case 2: Constant velocity in m/s in the x, y-direction

$$(V_x, V_y, V_z) = (0.2, 0.2, 0.0)$$

The result obtained is shown in Figures 5.4 to 5.6 and the variation with time of the total VOF within the sphere is shown in Table 5.2.

Case 3: Constant velocity in m/s in the x, y, z-direction

$$(V_x, V_y, V_z) = (0.2, 0.2, 0.2)$$

The result obtained is shown in Figures 5.7 to 5.9 and the variation with time of the total VOF within the sphere is shown in Table 5.3.

Figure 5.4: Sphere translation in the x, y-direction (time step = 0).

Figure 5.5: Sphere translation in the x, y-direction (time step = 20).

Figure 5.6: Sphere translation in the x, y-direction (time step = 40).

Figure 5.7: Sphere translation in the x, y, and z-direction (time step = 0).

Figure 5.8: Sphere translation in the x, y, and z-direction (time step = 20).

Figure 5.9: Sphere translation in the x, y, and z-direction (time step ≈ 40).

Time	Total VOF(Exact Volume)	Total VOF(Calculated Volume)	Error %
0	65.473520604409273	65.473520604409273	0.000000000
20	65.473520604409273	65.473520604409231	6.51143e-14
30	65.473520604409273	65.473520604409103	2.60456e-13
40	65.473520604409273	65.473520604409053	3.47276e-13

Table 5.3: The variation with time of the total VOF within the sphere during the translation in the x, y and z-direction.

5.2 Rotation of disc

A rigid body rotation velocity field is imposed with center of rotation at the center of the domain. This field will cause all fluid elements to rotate around this center. The body will undergo one rotation in ϕ time units, and should not change shape as a result of this rotation.

5.2.1 Vof initialization

Let the center of the disc be (x_c, y_c, z_c) , the thickness of the disc be $2h$, the radius of disc be r_0 , thus the VOF distribution should be:

$$f = \begin{cases} 1 & \text{if } (x - x_c)^2 + (y - y_c)^2 < r_0^2 \\ & \text{and } |z - z_c| < h \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the distance of a point (x, y, z) to the surface of the disc is:

$$D = \max\{\sqrt{(x - x_c)^2 + (y - y_c)^2} - r_0, |z - z_c| - h\} \quad (5.7)$$

Once the distance of the point to the surface of the disc is determined. the VOF initialization of the disc is identical to that of the sphere (See 5.1.1).

5.2.2 Disc rotation results

Disc Geometry: Radius of Disc: 0.25 m

Center of Disc: (1.0, 1.0, 1.0)

Thickness of Disc: 0.1 m

Mesh Data: Number of Elements: $20 \times 20 \times 20$

Mesh Size: 0.1 m

Timestep	Total VOF(Exact Volume)	Total VOF(Calculated Volume)	Error %
0	37.170864664981991	37.170864664981991	0.0000000000
100	37.170864664981991	37.170864664981836	4.20542e-13
150	37.170864664981991	37.170864664981777	5.73468e-13
200	37.170864664981991	37.170864664981707	7.64623e-13

Table 5.4: The variation with time of the total VOF within the disc (2D)

Simulation Parameters: Time Step size: 0.0222 s

Time Steps: 200

Velocity field:

Case 1: Rotation in 2-D

$$v_x = -(y - y_c)$$

$$v_y = x - x_c$$

$$v_z = 0 \quad (5.8)$$

The result obtained is shown from Figures 5.10 to 5.12 and the variation with time of the total VOF within the sphere is shown in Table 5.4.

Case 2: Rotation in 3-D

$$\bar{v} = \bar{\omega} \times (\bar{r} - \bar{r}_o)$$

Figure 5.10: Disc rotation in 2-D (time step = 0).

Figure 5.11: Disc rotation in 2-D (time step ≈ 100).

Figure 5.12: Disc rotation in 2-D (time step = 200).

Timestep	Total VOF(Exact Volume)	Total VOF(Calculated Volume)	Error %
0	37.170864664981991	37.170864664981991	0.000000000
100	37.170864664981991	37.170864664981877	3.05850e-13
150	37.170864664981991	37.170864664981806	4.97005e-13
200	37.170864664981991	37.170864664981755	6.30814e-13

Table 5.5: The variation with time of the total VOF within the disc (3D).

$$\text{with } \omega = (1, 1, 0) \quad (5.9)$$

The angular velocity is:

$$\omega = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2} = \sqrt{2} \quad (5.10)$$

This is the same as:

$$\begin{aligned} v_x &= z - z_c \\ v_y &= -(z - z_c) \\ v_z &= (y - y_c) - (x - x_c) \end{aligned} \quad (5.11)$$

The result obtained is shown in Figures 5.13 to 5.18 and the variation with time of the total VOF within the sphere is shown in Table 5.5.

Figure 5.13: Disc rotation in 3-D (time step = 0).

Figure 5.14: Disc rotation in 3-D (time step = 25).

Figure 5.15: Disc rotation in 3-D (time step = 50).

Figure 5.16: Disc rotation in 3-D (time step = 75).

Figure 5.17: Disc rotation in 3-D (time step = 100).

Figure 5.18: Disc rotation in 3-D (time step = 200).

5.3 Surface Tension modeling results

A unit cube with 8 triangles on each face of the cube (see Figure 5.19) is placed in the center of $10 \times 10 \times 10$ mesh, inside the cube VOF is 1.0 and outside the cube VOF is 0.0. There are a total of 26 nodes and 48 triangles on the cube. The corners of the cube should have maximum force, the edges of cube should have large forces and the mid-face nodes should have zero forces.

Cube Geometry: Dimension of Cube: 1

Center of Cube: (4.5, 4.5, 4.5)

Mesh Data: Number of Elements: $10 \times 10 \times 10$

Mesh Size: 1

The results show that the eight corners of the cube have maximum forces (see Table 5.6), the six nodes at the center of each face on the cube have zero forces (see Table 5.7), and the 12 nodes in the middle of each edge of the cube have large forces (see Table 5.8). The forces on the unit cube are also shown in Figure 5.20.

Once the surface tension forces at 26 nodes of the unit cube are found. Peskin's delta function is used as a smoother to distribute the surface tension

Figure 5.19: unit cube with 8 triangles on each face.

Node	surfaceForceU	surfaceForceV	surfaceForceW	modulus of Surface force
0	-2.0	-2.0	-2.0	3.4641
1	2.0	-2.0	-2.0	3.4641
2	-2.0	2.0	-2.0	3.4641
3	2.0	2.0	-2.0	3.4641
4	-2.0	-2.0	2.0	3.4641
5	2.0	-2.0	2.0	3.4641
6	-2.0	2.0	2.0	3.4641
7	2.0	2.0	2.0	3.4641

Table 5.6: The surface tension force on 8 corners on the unit cube.

Figure 5.20: The surface tension forces on the nodes on triangulation of the unit cube.

Node	surfaceForceU	surfaceForceV	surfaceForceW	modulus of Surface force
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0

Table 5.7: The surface tension force on 6 nodes in the center of each face on the unit cube.

force within and near to the interface transition region. It shows from the result that $5 \times 5 \times 5$ nodeGs in the center of the $10 \times 10 \times 10$ mesh have been affected by the surface tension forces on the unit cube. This is exactly what we expected, since, as we mentioned before, for each node on the cube, it can act on at most $4 \times 4 \times 4 = 64$ nodeGs. Also it is found that for each nodeT, the sum of the forces at the $4 \times 4 \times 4 = 64$ nodeGs, which the nodeT influences, is equal to the surface tension force at this nodeT. This is required by Peskin's delta function.

Node	surfaceForceU	surfaceForceV	surfaceForceW	modulus of Surface force
14	-1.0	-1.0	0.0	1.414
15	1.0	-1.0	0.0	1.414
16	1.0	1.0	0.0	1.414
17	-1.0	1.0	0.0	1.414
18	-1.0	0.0	-1.0	1.414
19	0.0	-1.0	-1.0	1.414
20	1.0	0.0	-1.0	1.414
21	0.0	1.0	-1.0	1.414
22	-1.0	0.0	1.0	1.414
23	0.0	-1.0	1.0	1.414
24	1.0	0.0	1.0	1.414
25	0.0	1.0	1.0	1.414

Table 5.8: The surface tension force on 12 nodes on the middle of each edge on the unit cube.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Accurate modeling of interfacial flows, such as liquid metal filling of moulds in casting processes, must have high-fidelity, robust algorithms for interface kinematics (tracking) and dynamics (e.g., surface tension). Our latest algorithmic developments for interfacial flows have been described: PLIC VOF method for interface tracking and a surface tension method for interfacial surface tension.

After several interface tracking methods were reviewed, the piecewise linear interface calculation VOF method has been chosen to implement our

PLIC model. This follows Youngs' method [18]. This algorithm is a true interface tracking algorithm, with fluxes derived geometrically rather than algebraically. The two basic tasks to be solved by the interface advecting method are interface construction and the volume of fluid advancing by Strang splitting. The interface construction includes the interface normal estimation, VOF matching, and volume of liquid from the donor cell to the acceptor cell computation, where the last two are determined by the interface type. The key issues for the PLIC method are found. The results of PLIC method are quite sensitive to the normal to the interface $\bar{n}$, so an accurate estimate of n often is very important for good performance. Numerical surface tension is demonstrated in 3-D disc rotation. How to minimize the numerical surface tension will be a subject of future work.

The purpose of our surface tension method is try to model physical processes specific to and localized at fluid interfaces. Our surface tension method falls under the general class of immersed interface methods [15]. In order to avoid numerical problems caused by the second derivative, Keith Miller's integration method [12] is used to calculate the surface tension force per unit area of the interface in our surface tension model. Keith Miller states that a force normal to a surface with a magnitude equal to curvature is equivalent

to a surface tension force applied to the boundary of the surface. Peskin's delta function is used here to act as a smoother, which distributes the surface tension force near the interface transition region. The test case of a unit cube with 8 triangles on each face show that the surface tension method is very promising, but further tests will need to be done in future work.

6.2 Future work

Our interfacial flow model could be extended. The following are recommended for future work.

1. Combine the PLIC method and the surface tension method to build an overall interfacial flow model, which is topologically robust and could solved interface kinematics and dynamics accurately.
2. The surface tension method needs further work and more tests to elucidate the strengths and weaknesses of the method.

Bibliography

- [1] J. U. Brackbill and D. B. Kothe. A Continuum Method for modeling Surface Tension. *Journal of Computational Physics*. Vol. 100:335–354. 1992.
- [2] J. Campbell. *Castings*. Jordan Hill Oxford: Butterworth-Heinemann Ltd., 1991.
- [3] A. J. Chorin. A Numerical Solution of the Navier-Stokes Equations. Vol. 22:745–762, 1967.
- [4] T. Chung. *Finite Element Analysis in Fluid Dynamics*. New York: McGraw- Hill, 1978.
- [5] N. M. Dai. *Automating the Analysis of Complex Physical Systems. The Virtual Foundry*. PhD thesis, Carleton University, Ottawa. Canada.

August 1994.

- [6] B. J. Daly. Numerical study of two fluid Rayleigh-Taylor instability. *Physics of Fluids*, Vol. 11:15–30, 1967.
- [7] J. A. Goldak. Interface Advection with PLIC and PPIC. unpublished. 1998.
- [8] P. M. Gresho. A Modified Finite Element Method for Solving the Time-Dependent solution of the incompressible Navier-Stokes equations in two- and three- dimensions. *International Journal for Numerical Methods in Fluids*, Vol. 4:619–640, 1984.
- [9] E. Hughes. Computation of Incompressible Navier-Stokes Flows. Internal Communication, Carleton University, 1992.
- [10] D. B. Kothe and W. J. Rider. Volume Tracking of Interfaces Having Surface Tension in Two and Three Dimensions. <http://www.c3.lanl.gov/wjr/pubs.html>, 1996.
- [11] R. J. Leveque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *Journal on Numerical Methods in Fluids*, Vol. 31:1019–1044, 1994.

- [12] K. Miller and A. Kuprat. Private communication. 1998.
- [13] W. F. Noh and P. R. Woodward. SLIC (simple line interface method). *Lecture Notes in physics 59*, edited by A.I. van de Vooren and P.J. Zandbergen:330–340, 1976.
- [14] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulation. Vol. 79:12–49, 1988.
- [15] C. S. Peskin and D. M. McQueen. Fluid Dynamics of The Heart and Its Valves. *Case Studies in Mathematical Modeling - Ecology. Physiology. and Cell biology*, edited by Othemer, H.G.:309–337, 1996.
- [16] G. R. Price and G. T. Reader. A Piecewise Parabolic Interface Calculation For Volume Tracking. *Computational Fluid Dynamics*, Vol. 3:71–78, 1998.
- [17] G. R. Price and R. D. Rowe. An Extended Piecewise Linear Volume-of-fluid Algorithm for Reconstructing Thin Interfaces. *Computational Fluid Dynamics*, Vol. 11:9–14, 1997.

- [18] D. L. Youngs. *An Interface Tracking Method for a 3D Eulerian Hydrodynamics Code*. Atomic Weapons Research Establishment, Aldermaston. Reading, RG7 4PR, U.K., Ref. AWRE/44/92/35, 1984.

IMAGE EVALUATION TEST TARGET (QA-3)

APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved