Stony Brook University

Applications of Front Tracking to Multiple Scientific Problems

A Dissertation Presented

by

Xicheng Jia

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

August 2007

Stony Brook University

The Graduate School

Xicheng Jia

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

Xiaolin Li
Professor of Applied Mathematics
Dissertation Advisor

James Glimm
Professor of Applied Mathematics
Chairman of Defense

Yan Yu
Assistant Professor of Applied Mathematics
Inside Member

Roman Samulyak
Brookhaven National Laboratory Computational Center
Outside Member

This dissertation is accepted by the Graduate School.

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Applications of Front Tracking to Multiple Scientific Problems

by

Xicheng Jia

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

2007

We present several new developments of the Front Tracking method to solve scientific and engineering problems including constructing 3-D grid cells with multiple components, propagating 2-D boundary nodes through contact angle, and simulating the cell migration behaviors. The purpose of our study is to extend the Front Tracking method into more research fields where the tracking of high resolution discontuinities is essential.

Key Words: Front Tracking, multiple components, contact angle, cell migrations.

To my parents with all my love

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my advisor, Professor Xiaolin Li, for the exciting and challenging topics and for his guidence and continuous support for my research. His insight and experience on the computational mathematics and physics helped me reorganize many of my previous thoughts and learn some important facts about the scientific philoshophy.

Thanks to Professor James Glimm, his working enthusiam and ways of deep thinking are always the role model for people who want to be different.

Thanks to my parents, Xuedun Jia and Hean Xie and my sister Xiping Jia, for their continued understanding and support through years. Thanks to my friends Taewon Lee, Dayoung Lee, Hyunsun Lee and Yuanhua Li for all your support and friendship.

Chapter 1

Introduction

Computational Fluid Dynamics(CFD) is one of the most important branches of the continuum media physics. It combines the knowledge of the classical fluid dynamics with the growing computational technologies, and has been rapidly developed in the recent forty years.

The Front Tracking method, accompanied by the *FronTier* code, is an unique method to attain sharp solutions for physical problems featuring with discontinuities. In this paper, we explain new algorithms and extend the Front Tracking method into several new applications.

1.1 Equation systems

The *FronTier* code is a comprehensive software package which contains the solvers for various physical problems. For a gas dynamic problem, the underlying mathematical model of the Front Tracking method is based on the Euler equation which follows three conservation laws (mass, momentum, and energy conservations) and the equation of states (EOS). For example, the Euler equation for one-dimensional compressible fluid flow is as follows:

$$\frac{\partial \overrightarrow{U}}{\partial t} + \frac{\partial \overrightarrow{f}}{\partial x} = 0 \tag{1.1}$$

where

$$\overrightarrow{U} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix} \qquad and \qquad \overrightarrow{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e+p) \end{bmatrix} \tag{1.2}$$

u is the velocity of the fluid parcel, p is the pressure, ρ is the fluid density, and e is the energy. $\frac{\partial \overrightarrow{U}}{\partial t}$ is the time-dependent term which describes the local rate of change with time for variables in U, and $\frac{\partial \overrightarrow{f}}{\partial x}$ is the flux convection term. It is worth noting that for most of the time-dependent finite difference schemes, the convection term is the most important part to find the accurate position and the strength of the captured discontinuities.

In case the viscosity is important in the studied physics, a source term is introduced into the equation system to specify the viscosity-related influences. Such equation system is known as the Navier-Stokes equation, and the basic one-dimensional N-S equation is Equation 1.3, where $\overrightarrow{VIS}$ is the viscosity source term.

$$\frac{\partial \overrightarrow{U}}{\partial t} + \frac{\partial \overrightarrow{f}}{\partial x} = \overrightarrow{VIS} \tag{1.3}$$

1.2 Numerical Methods

Finite difference is the main method used in the *FronTier* package as the PDE solver. Several frequently used schemes *i.e.* MUSCL, Lax-Wendroff and PPM, are also joined into the *FronTier* solver. On the neighborhood of the discontinuities, Reimann solver is applied to compute the propagated positions of the discontinuities.

For a given problem, the computational domain is discretized into grids. For each grid point, the modeling equations are solved to compute its physical parameters by using variables at its neighboring grid points.

For most finite difference schemes, there are common factors that may influence the accuracy and robustness of the simulations:

1. How to handle the discontinuity, especially the contact discontinuity. In the neighborhood of these physical waves, the physical properties may have a sharp gradient and two adjacent grid points may follow completely different equations.

2. How to handle the boundary conditions. For any time-dependent finite difference method, a small error on the boundary can be propagated inward and accumulate to influence the overall numerical solution.

Front Tracking method is a Godunov-type scheme which solves Reimann problems on grid blocks for flux calculation and applied a direct way to explicitly track discontinuities. More details of this method are discussed in §1.3.

1.3 Discontinuities and Reimann Solution

Knowing the equations at the discontinuities where one or more physical properties change dramatically within several neighboring grid cells is a primary challenge for most numerical simulations. This is the major reason the numerical solution becomes unstable or even incorrect if not treated properly. In fluid dynamics, there are two types of discontinuities: shock and contact.

1.3.1 Shock, Rarefaction and Contact

Sod's classical one-dimensional Shock-Tube experiment is a good example to explain the differences between the shock, the rarefaction and the contact.

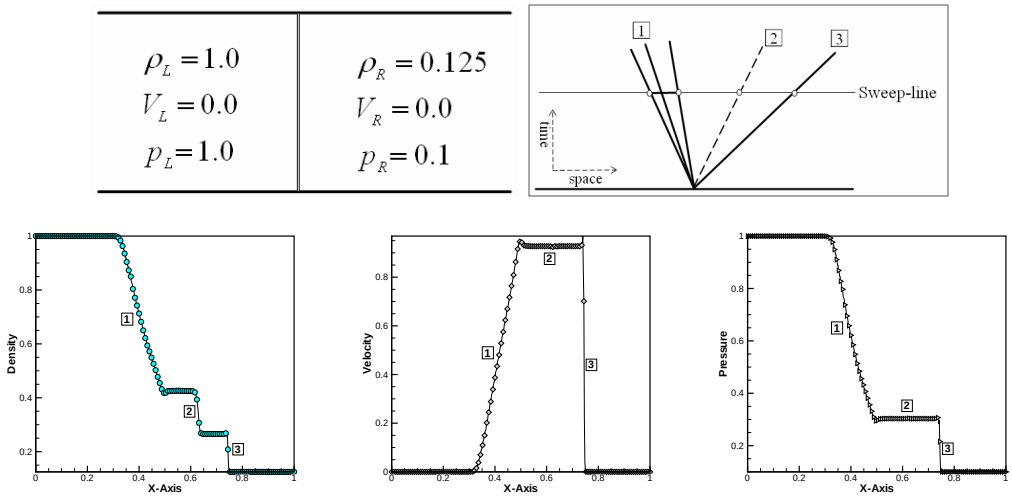


Figure 1.1: 1-D shock tube experiment: Top: Left: initialization, Right: types of discontinuity. Bottom: density, velocity, pressure distributions.

See Figure 1.1 Top Left, at initial time, a box is separated by a thin film and gas in each side of the film has very different physical properties.

At $time = 0$, the film is broken, and various discontinuities evolve and spread along the box. Three types of discontinuities can be found at the location 1, 2, and 3 as in Figure 1.1top-right. The density, velocity and pressure at $time = 0.14$ computed by a second order upwind TVD scheme(computational grid size is 100 along the x-direction, 1997 by Xicheng Jia) can be shown in the Figure 1.1 bottom.

At location 3 where there is a shock, the density, velocity and pressure all have very sharp gradients and the changes in the quantity span only two grids in width.

At location 1 where this is a rarefaction, although the density, velocity and pressure all have large gradients, the changes span more grid widths and are smoother.

Different from both the shock and the rarefaction, at location 2 where there is a contact, the density has a very sharp gradient and the change spans about 3 grid widths while both the pressure and velocity are continuous without change.

Figure 1.1 Top Right shows the discontinuities which spread along the spatial (horizontal) direction and the time (vertical) direction in Sod's experiment. If we use a horizontal straight line(keep time as a constant) to sweep along the spatial-axis, the resultant rarefaction is shown as a line segment, while both the shock and the contact are points.

1.3.2 Reimann Solution

Riemann problem is a fundamental problem to study the interactions among waves. Numerical methods involving Riemann solution, *i.e.* Godunov scheme, Glimm scheme et al., treat the flow as a set of small-scale waves, and Reimann problems can then be solved exactly between the adjacent regions of these waves.

Considering the one dimensional Euler equation as in Equation 1.1. In case of the discontinuities, the partial derivatives in Equation 1.1 are not defined, and the Rankine-Hugoniot condition is applied to obtain weak solution.

$$\overrightarrow{n} \cdot ([\overrightarrow{f}] - v[\overrightarrow{U}]) = 0 \tag{1.4}$$

where $[A] = A_+ - A_-$ (Note: A can be either $\overrightarrow{f}$ or $\overrightarrow{U}$ in Equation 1.4) is the jump in the quantity of the variable A across the discontinuity, v is the velocity of the discontinuity, and $\overrightarrow{n}$ is the unit vector normal to the discontinuity.

For a discontinuous problem, solutions of Equation 1.1 are not unique, and an entropy condition must be supplied to make the solution unique. According to the initial physical states at the left and the right sides of the discontinuity, Reimann

solutions can be divided into five cases, see Figure 1.2.

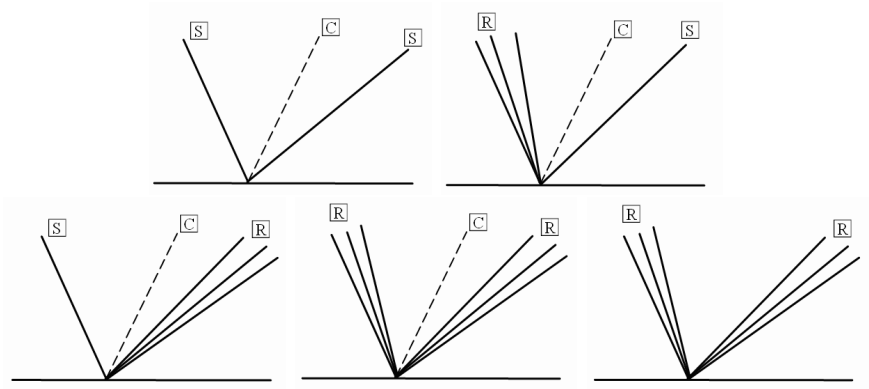


Figure 1.2: Five typical Riemann solutions: (a) Shock-Contact-Shock, (b) Rarefaction-Contact-Shock, (c) Shock-Contact-Rarefaction, (d) Rarefaction-Contact-Rarefaction, (e) Rarefaction-Vacuum-Rarefaction.

Each type of the five cases provides an unique description of the wave interactions(Fu 1993 [20]).

1.4 Front tracking Method and FronTier

Front Tracking method have been applied to the studies of fluid interface instabilities since the late 1980's([21, 22]). This method, combined with some classic CFD method *i.e.* MUSCL, produces high resolutions in studying the physical problems with discontinuities.

1.4.1 Tracking Methods

Three main tracking methods are used in the *FronTier*: grid-free, grid-based and the hybrid tracking methods(see [21]).

In the grid-free tracking method, the propagated interface does not necessarily have a relation with the finite difference grid. The propagation of the interface posi-

tions in this method is often more accurate but suffer from complexity in topological bifurcations.

The grid-based tracking method, on the other hand, adapted the reconstruction of the interface on Eulerian mesh blocks to simplify the topological bifurcations. This method is more robust than the grid-free method and therefore is often used in computing the fluid interface instabilities where topological changes are frequent.

The hybrid tracking method (locally grid-based tracking method) is a compromise of the above two methods which combined the accuracy of the grid-free method with the robustness of the grid-based method. Reconstruction is only used locally when and where bifurcation occurs. The hybrid method has been a major improvement to the previous tracking methods and its implementations keep an ongoing subject of the *FronTier* package(Du et al. 2006 [14]).

1.4.2 Applications

The applications of the Front Tracking method have included fluid interface instabilities, the shock implosions and fuel-injection jet et al. Aside from the original *FronTier* code, a general purpose software library is also available for scientific computation of various dynamic interfaces.

In our studies, we presented some recent development of the Front Tracking method and new applications including 3-D three-component interface constructions, the 2-D thin film formation, and cell migration problems et. al. Both the original *FronTier* package and the newest *FronTier* library packages are applied in the simulations.

Chapter 2

Multiple Component Problems

The Marching Cube algorithm [31] is an efficient way to numerically construct complex geometric structures by mapping all potential grid cell conformations into some isomorphically distinct patterns. Here, we extended the Marching Cube algorithm into the construction of three dimensional grid cells containing three different materials.

2.1 Marching Cube Algorithm

The Marching Cube algorithm was proposed by Lorensen and Cline in 1987 [31] and now is one of the standard methods for iso-surface generation. As one of the most useful surface modeling algorithms, the Marching Cube method provides a high-resolution 3D surface construction and has been widely used in many research fields, *e.g.* medical imagining and computational graphics where the applications are mainly based on the grid cells(Rajon et al. 2003 [50]).

2.1.1 Mechanism

The principle of the Marching Cube algorithm is that the crossings are located by determining which edges of the cube are intersected by the isosurface and then a set

of triangle patches are constructed by using these crossings. A surface representation is constructed by connecting the neighboring triangular patches from all cubes on the isosurface boundary.

Basically, the Marching Cube method uses the symmetric properties of the geometric structures of a grid cell and maps all possible combinations of the components on eight grid vertices into a look-up table. Based on the component values of these vertices, a set of grid-line crossings and grid-face crossings are constructed and connected to describe how a surface cuts through each cell in a three-dimensional spatial domain.

There are two main methods used in the Matching Cube algorithm to establish such mapping. For example, for a three-dimensional and two-component problem, the conformation mapping can be established by:

- use a look-up table of all 256 conformations which collect all potential combinations of corner status and then map them into 14 basic patterns. This is the faster way and take time complexity of O(1) (Bourke, 1994 [8]). The downside of this method is that the table of 256 potential conformations has to be found manually beforehand. For three-component problem, this becomes much more difficult and error-prone since the number of potential conformations becomes $3^8 = 6561$.

- use a look-up table of 14 patterns and let the computer programs do the rotations and complementations within the cube. This technique, termed as 24 rotations (for three-dimensional only), is a more generic way and easier to be implemented. The downside is that it uses a brute-force algorithm to loop through 24 potential rotations, and thus may take more computational time (up to $14 \times 24 = 336$ tests for each grid cell) to find the correct conformation

pattern.

The implementation of the Marching Cube algorithm is the foundation for the grid-based tracking method in *FronTier*. By matching over the entire computational domain, the cells are treated one at a time and then by using the component values of eight vertices on the grid cell, a conformation pattern is selected from the look-up table and the triangular patches are constructed by connecting related crossings, and then the interface is constructed by the linking of these triangles in the cell.

2.1.2 Terminologies

Prior to the introduction of building grid-based interfaces, some terminologies frequently used in three dimensional *FronTier* simulations need to be explained.

See Figure 2.1, if we divide the computational domain into grid blocks, each block is a cube which has eight vertices, six grid-faces, and twelve grid-lines(edges).

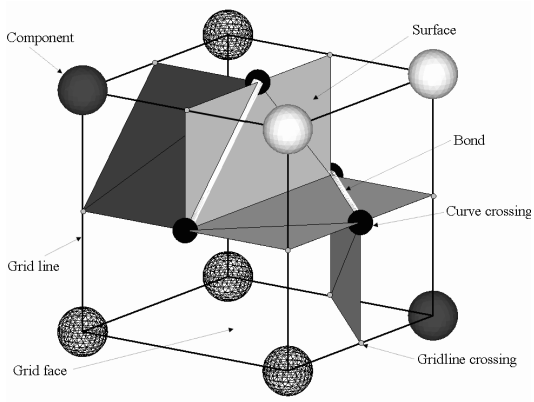


Figure 2.1: Technical terminologies in FronTier code.

In *FronTier*, an interface is an arbitrarily complex topological object consisting of a set of surfaces which separate the fluids. The surfaces divide the computational domain into a set of connected sub-domains which contain distinct materials. Each

surface on the interface is represented by a set of planar triangular elements, and different surfaces intersect only along curves. A curve is a set of connected bonds which are line segments connected by neighboring curve crossings on the grid-faces. Each curve is a continuous object which has start and end points, and a boundary point is called node. The most fundamental geometric object in *FronTier* is the point (the x, y, z coordinate set) which is used to describe a position in the space.

Most of the objects described above are oriented manifolds, and have topological properties *e.g.* positive-negative, incoming-outcoming, start-end and so on. For example, if we labeled three materials with comp-0, comp-1, and comp-2, the orientation of three surfaces follows a rotation of the components (See Table 2.1). Note: the orientation of the intersecting curve is unique and must be specified deliberately in order to cope with the surface orientations.

Surface number	Positive component	Negative component
0	$comp-0$	$comp-1$
1	$comp-1$	$comp-2$
2	$comp-2$	$comp-0$

Table 2.1: The orientations of the surfaces. The direction of a surface is from the negative component pointing to the positive component.

Figure 2.2 shows the definition of the orientations for both the surfaces and the intersecting curve. The orientation of the surfaces follows the right-hand rule, for example, the positive direction is from $comp-0$ to $comp-1$ for $surface-0$, from $comp-1$ to $comp-2$ for $surface-1$, and from $comp-2$ to $comp-0$ for $surface-2$. By this way, the positive direction of the bond where three surfaces intersect, keeps coherent, from $P1$ to $P2$.

In *FronTier*, Numbers 0 ans 1 are used to specify the positions of objects in a grid cell. For example, 0 is generally used to denote the lower boundary, and 1 is the

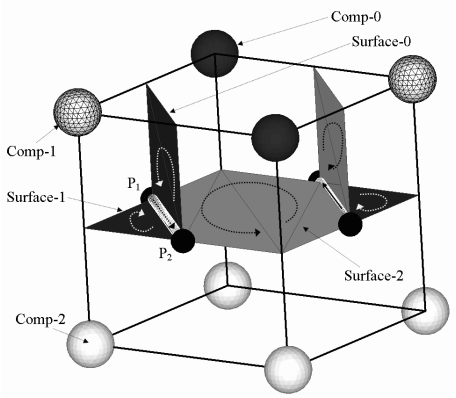


Figure 2.2: The orientations of the surfaces are given by the right-hand rule and then the direction of the bond can be determined accordingly.

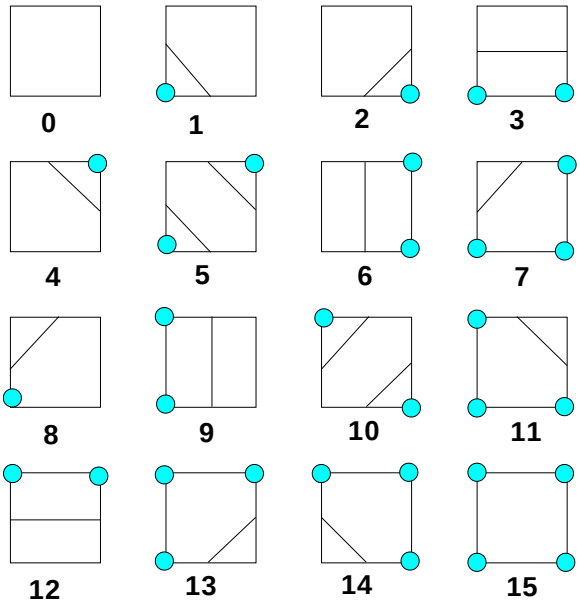
upper boundary. $(x, 0)$ means the lower boundary at x direction, and $(z, 1)$ means the upper boundary at z direction. Similarly, the 8 vertices may be expressed as (0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1) in a (x,y,z) coordinate system.

2.1.3 24 Rotations

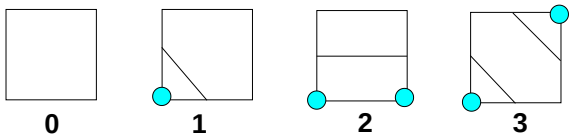
Because of the structure symmetries of the cube, any potential conformation of the n-dimensional cube can be mapped into a number of isomorphically distinct patterns by geometric transformations. The rotation group of the n-dimensional cube is one of such transformations for the mapping.

Given a simple 2-dimensional cube(square) of two-component problem as an example, the number of possible conformations is $2^4 = 16$, see Figure 2.3.

In Figure 2.3 top, conformation 1, 2, 4, 8 can be derived from one another by basic rotations and then grouped into a single pattern numbered 1 as shown in Figure 2.3 bottom. Moreover, conformation 14 is a binary complementation of the conformation 1, thus there is no need to create a new pattern for this configuration. By using a similar rotation and reflection technique, all 16 conformations can be mapped into



(a) 16 potential conformations.



(b) 4 basic patterns.

Figure 2.3: 2-dimensional cube conformations and patterns. Top: the 16 conformations for a 2-dimensional cube. Bottom: After applying rotations and reflections, the conformations can be reduced to four patterns

the four patterns of Figure 2.3 bottom.

Some principles can be made to work in constructing three-dimensional conformations, they are:

- Rotation over any of the 3 primary axis
- Reflecting the shape across any of the 3 primary axis
- Inverting the components of all corners and flipping the normals of polygons.

According to Malkevitch (2003[33]), the size of the rotation group for a n-dimensional cube is $2^n(n!)$. For example, the rotation group must has 8 rotations for a two dimensional cube(square), and 24 rotations for a three dimensional cube(cell).

Taking this into account we can resolve the original 256 potential conformations of a two-component problem in 3D cube down to a total of 16 combinations. With this number it is much easier to create predefined polygon sets for making the appropriate surface representations.

Note that Matching Cube algorithm does not consider the symmetric by reflection, it considers only rotations, and the reflection has to be treated independently. If we consider the reflection, the typical patterns for a two-component grid cell can be reduced to 14.

From Li et al.(2003 [30]), the number of the isomorphically distinct patterns for three-dimensional grid cells with different multiple component (from one to eight) problems are shown in Table 2.2.

NO. of Distinct Comp	1	2	3	4	5	6	7	8
NO. of patterns	1	14	57	94	60	19	3	1

Table 2.2: The maximum number of the typical patterns for three-dimensional grid cells containing different numbers of distinct components.

For a three dimensional grid cell which contains exactly three distinct components, there are $3^8 = 6561$ possible conformations. But by using our extended Marching Cube algorithm, these 6561 conformations can be represented by 57 isomorphically distinct patterns. We will discuss this in the next section.

2.2 Three-Component Problems

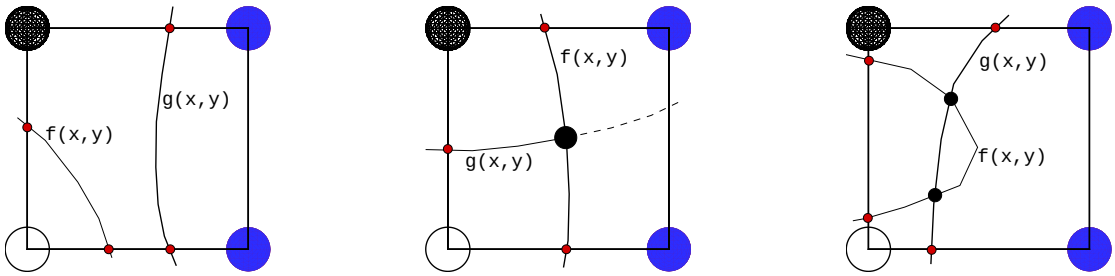
When we construct the interfaces on a grid cell containing at most two distinct components, there is no interaction between different surfaces. But in three-component problems, the situation becomes more complex. For example, there can be up to three surfaces within a single grid cell. Thus how to distinguish among different surfaces and how to assign consistent surface and curve orientations become the main concern.

To define a surface representation within a grid cell which contains exactly three different components, we first look at the problem from the mathematical point of views. Assuming components in the grid cell can be divided by two functions $z = f(x, y)$ and $z = g(x, y)$, the intersections of these two functions define the bonds that separate three components into distinct parts.

Based on each grid-face(see Figure 2.4 Left), the intersecting point of two functions $z = f(x, y)$ and $z = g(x, y)$ is called a curve crossing. Different from a grid-line crossing which is a point located on the grid line separating two distinct materials, a curve crossing is a point in the grid-face at which three different materials meet together.

Since the curve crossings may occur in any position of a grid-face, and given a grid-face conformation (Figure 2.4) by solving the above two equations, there can be a single curve crossing, no curve crossing, or multiple curve crossings.

If we consider all these potential conformations and 6 grid-faces in a single grid cell, the number of the typical patterns extracted from the Marching Cube algorithm will be added up to at least $3 \times 6 = 18$ times of the number in the original look-up patterns, which is too expensive in numerical computations and have to be simplified.



(a) no crossing (b) a single crossing (c) two crossings

Figure 2.4: Left: typical grid-face conformations when there is a single curve crossing(Left), no curve crossing(Middle) and multiple curve crossings(Right).

2.2.1 Curve Crossing Assumptions

After the conformation of a grid cell is specified and the component values on the vertices are fixed, the grid-line crossing can be uniquely determined by the following rules:

1. one and only one crossing must exist on a grid line with two distinct components at each end.

2. At most one crossing on each grid line.

These rules remove two possible curve-crossing cases in Figure 2.4 Middle, Right, but still have problems as shown in Figure 2.5. In order to further simplify our problem, some restrictions are made in defining a curve crossing.

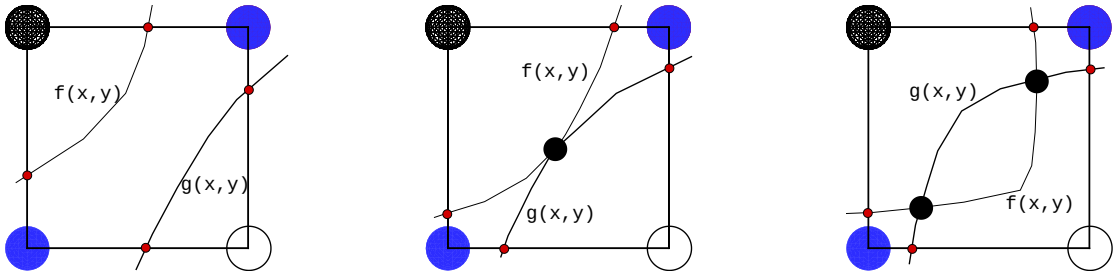
On each grid-face, a curve crossing is uniquely determined if and only if the following assumptions are satisfied:

1. Exactly three distinct components are found.

2. Two components at the diagonal positions must be different.

A sample grid-face which satisfies all above assumptions is shown in Figure 2.4 Left.

By the above assumption, there does not exist a curve crossing in a grid-face conformations(see Figure 2.5 Left). In some computational cases it is possible to have at least one curve crossing(see Figure 2.5 Middle, Right).



(a) no crossing (b) a single crossing (c) two crossings

Figure 2.5: grid-face conformations which must not have a curve crossing by the assumptions. Left: no curve crossing which is valid according to our assumptions. Middle and Right: deprecated cases where one or more curve crossings exist.

2.2.2 57 Pattern Construction

We construct the three-component interface on each grid cell as follows:

1. For every grid-line with two distinct components at each end, one and only one grid-line crossing must exist. A surface number (see Table 2.1) is specified to each grid-line crossing according to the values of the two components.

2. For each grid face, one and only one curve crossing must exist if it satisfies two assumptions introduced in 2.2.1.

NO. of Comp-0	NO. of Comp-1	NO. of Comp-2	NO. of case
1	1	6	3
1	2	5	7
1	3	4	13
2	2	4	16
2	3	3	18

Table 2.3: The 57 typical conformation patterns divided by the different numbers of distinct components.

3. Triangular patches are built by connecting all grid-line crossings having the same surface number with possible curve crossings. Triangulation using these crossing points should follow the same orientation of the given surface(see Figure 2.2).

4. One or more bonds are formed by connecting curve crossings.

By stitching the triangular patches from the neighboring grid cells, we can make surface representation, and we make the intersecting curve of three components by connecting all adjacent bonds into a continuous object.

As mentioned above, 6561 potential conformations exist in a grid cell containing three distinct components on 8 vertices. By the extended Marching Cube algorithm, any possible grid cell conformation can be described by one of the 57 isomorphically distinct patterns. According to the numbers of different components we can group 57 patterns into five categories, see Table 2.3.

Considering the possibilities of the reversal on the components for patterns with the set of 1(comp0)-1(comp1)-6(comp2), 2(comp0)-2(comp1)-4(comp2), and 2(comp0)-3(comp1)-3(comp2), there should be $3 + 16 + 18 = 37$ basic patterns which have to apply 24 rotations twice in order to cover all possible grid cell conformations.

2.3 Computational Results

To examine the feasibility of our assumptions in constructing the three-component grid cells, we conducted several numerical experiments on geometric problems where three-phase intersecting curves exist.

2.3.1 Pattern Constructions

Figure 2.6 shows some grid conformations computed by using our new algorithm. Both the surface representations and the bond are well constructed as expected.

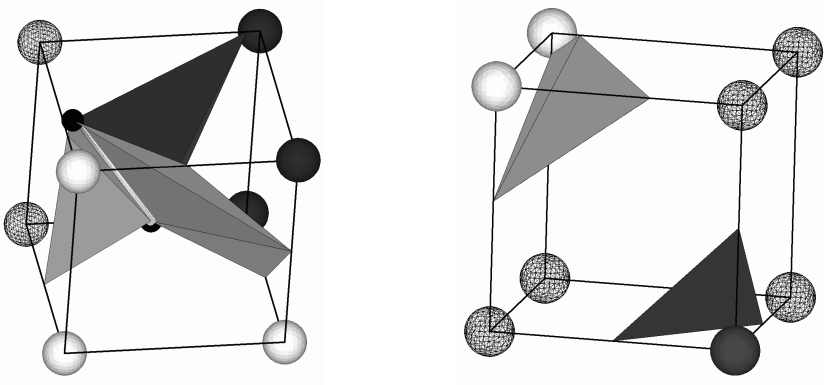


Figure 2.6: Two typical grid-cell conformations constructed by the new algorithm. Left: three surfaces and a bond. Right: two surfaces and no bond.

It's worth noting that the original Marching Cube algorithm uses the middle point of an edge to locate the position of a crossing point. In our computations, the coordinates of both grid-line crossing and curve crossing are computed by solving several equations used for initialization. there exist cases where some computed crossings are very close to the grid vertices, and the pre-constructed triangular patches from the isomorphically distinct patterns may not fit the real surfaces well.

Since the Matching Cube algorithm relies mainly on a small look-up table as

we have mentioned in §2.1, it may have cracks and inconsistent topology(Lewiner, 2003 [29]). For example, from Figure 2.7, we can find some tangled triangles and the intersection between surfaces are not smooth.

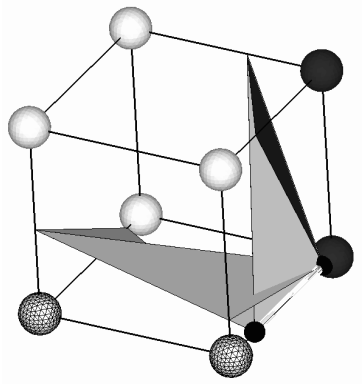


Figure 2.7: A computed grid conformation in which the bond is almost parallel to an adjacent grid edge and some triangular planes are tangled between each other.

2.3.2 An Ellipsoid Collides with a Plane

Figure 2.8 shows an ellipsoid drip into a horizontal free-boundary plane. Three components are the component inside the ellipsoid, the component outside the ellipsoid but above the horizontal plane, and the component outside the ellipsoid but below the horizontal plane. The ellipsoid is cut open in order to check details of the interaction from the inside. From Figure 2.8 Right, the curve is smooth enough to express the geometric structure of a three component surface intersection.

2.3.3 Two Ellipsoids Intersect

Figure 2.9 shows the collision of two ellipsoids. The smaller ellipsoid has been cut open to check the intersecting curve from the inside. The surface intersection is captured properly. Some glitches could be found on the intersecting curve, which

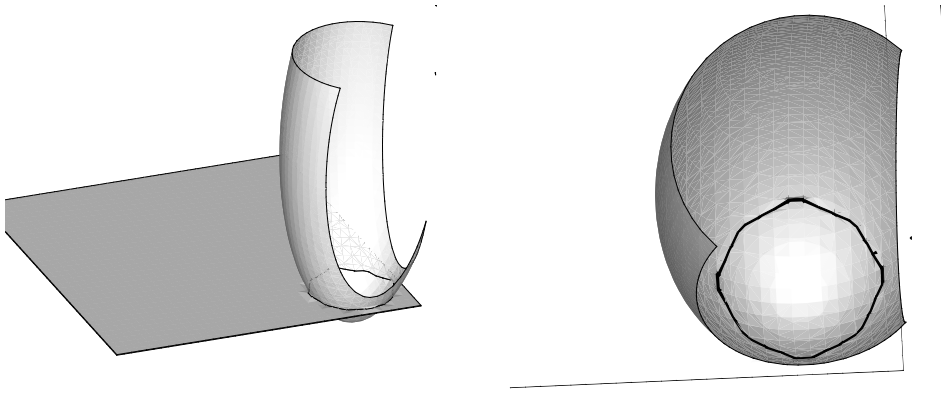


Figure 2.8: An ellipsoid collides with a horizontal free-boundary plane. Left: The interaction when part of the ellipsoid is below the plane. Right: grid details around the intersecting curve.

may be from the assumptions of defining the curve crossings or the pre-defined triangulations on the 57 grid cell patterns. Considering the computational spatial scale, these glitches are still acceptable.

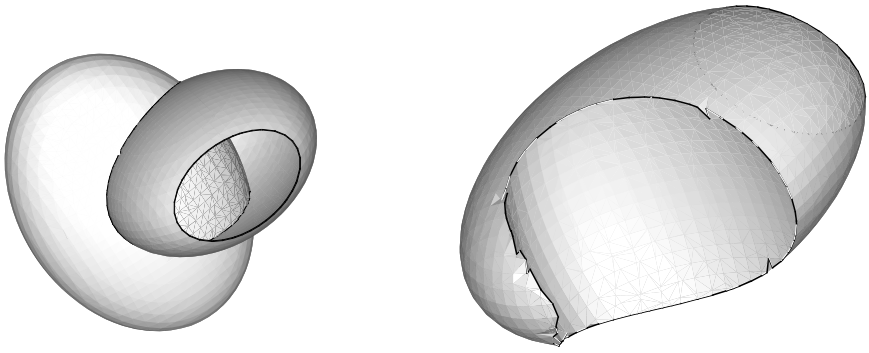


Figure 2.9: Two ellipsoids collide. Left: the intersecting curve shown from both inside and outside an ellipsoid. Right: some glitches found on the intersecting curve.

2.3.4 Three Dimensional Jet

In the initialization of three-dimensional fuel jet problem there is a boundary curve on which the fuel, the air, and the structure wall intersect, see the bold white line in Figure 2.10 Right and Figure 2.11 Right.

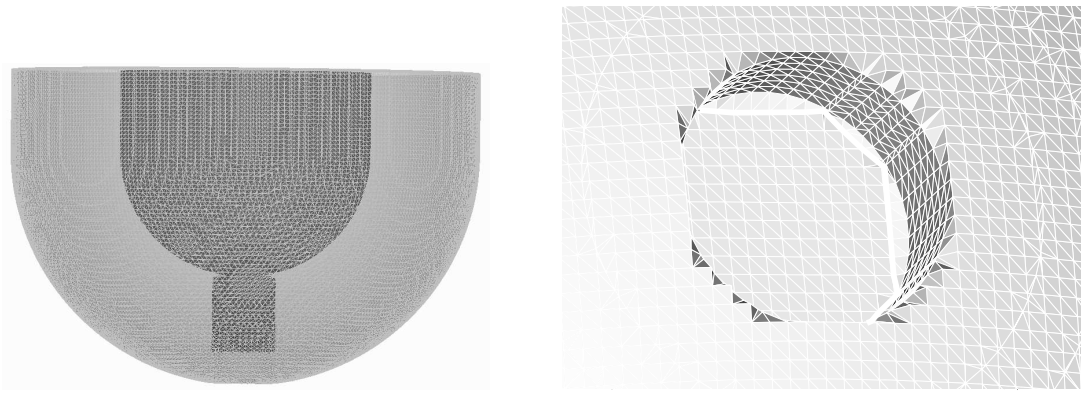


Figure 2.10: Initialization of 3-D jet problem(fine grid). Left: the initial grid when the contact is a plane inside the jet nozzle. Right: the grid details around jet nozzle.

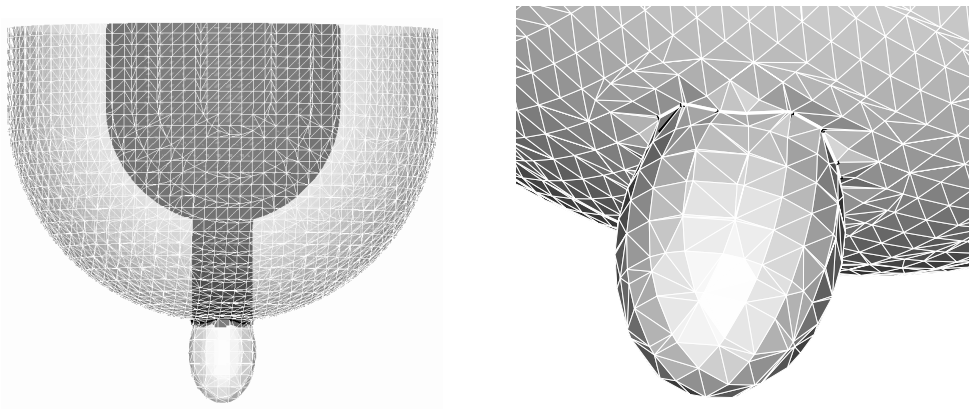


Figure 2.11: Initialization of 3-D jet problem(coarse grid). Left: the initial grid when the contact is an ellipsoidal surface outside the jet nozzle. Right: the grid details around the intersecting curve.

The computational grid applied in the computation is $150 \times 150 \times 150$ for the

fine grid and $50 \times 50 \times 50$ for the coarse grid. From the geometric plots, we can find that both the curve and surfaces were captured smoothly.

Note: the nozzle radius in the simulations is enlarged in order to display the structure of the jet clearly.

2.4 Summary

In our computation, we extended the applications of the Marching Cube algorithm into the three-component grid cell constructions. Some assumptions on defining the curve crossing are made to validate and simplify the implementation. The numerical experiments show that the method is appropriate in handling three dimensional grid cells containing multiple materials. Further work can be done to improve the triangulation method on the 57 patterns.

Chapter 3

The Contact Angle Problems

The contact angle plays the role of boundary conditions where more than three interfaces meet. Here we tried to establish some basic frames to study how the notions of the contact angle can be applied to describe a moving boundary node and how to implement them into the *FronTier* code.

3.1 The Contact Angle

The contact angle is a very important parameter in studying the interfacial physics which involves more than three different phases of materials. The experimental and theoretical research about the contact angle, *e.g.* dynamic wetting, spreading, had been extensively conducted (Dussan 1979 [15], Hocking 1976, 1995 [24, 25] et al.). One usual method of studying the contact angle is placing a drop of liquid on a flat horizontal plate and setting up an external device above the surface, the contact angle then can be measured either directly by drawing on an enlarged photograph a line tangent to gas/liquid interface at the contact line and using a protractor or indirectly by using the solution for the shape of the meniscus along with the measured height of the apex of the drop and its size. Some new measurements combine experimental equipment(*i.e.* goniometers) with specially prepared computer programs to get

better effect in studying the contact angles, *e.g.* Sklodowska et al. (1999 [55]) used such combination and measured the contact angle of a liquid drop on the surface of minerals.

Currently, there are two main challenges in attaining the exact contact angle. One is the micro-physical processes near the contact line of which identity is not yet established. Due to the complexity of particle motion in this region, most of the current models can be established only on the macroscopic processes. Hocking (1995) [25] derived the macroscopic contact angle in an evaporating drop, and found that the difference between microscopic and macroscopic contact angles is due to the physical process in the narrow region near the contact line where slip is important. In this immediate neighborhood of the contact-line, the application of the classic hydrodynamic assumptions such as no-slip boundary condition between a fluid and the solid surface gives rise to an unphysical stress singularity (Benintendi et al. 1999 [3]). This suggests that very close to the moving contact line, unique micro-physics processes other than classic hydrodynamic control the fluid motion (Chen et al. 1995, 1997 [10, 11]). Depending on different physical systems, a number of models for flow in the neighborhood of a contact line have been suggested to relieve such bifurcations, *e.g.* diffuse interface model, non-linear slip model, shear thinning model, models by finding the balance between surface tension and Van der Waals stresses (Eggers et al. 2004 [16]). But up to now, none of these models have been experimentally confirmed and the modeling to study the exact physics which govern the behaviors of the fluid in this contact-line region is still unknown (Dussan 1979 [15], Benintendi et al. 1999 [3]).

Research on the dynamic behaviors of the contact angle is another challenge in studying the interfacial physics. Until now, very little is known about the details of which one fluid displaces another at the moving contact line. From Tanner's law

[17], the local dynamic contact angle is dependent on the relative speed between the substrate and the contact line through a cubic function, *i.e.* $U \propto \alpha^3$. Theoretical justifications of this relation had been given by different research, *i.e.* Eggers et al. (2004 [16]) adapted a logarithmic correction on the speed in a model-dependent manner. Dussan (1979 [15]) studied the static and dynamic contact angle on several typical situations where fluid flow has a significant influence on the moving contact line. Besides the motion of the fluids, the value of the dynamic contact angle also depends on the geometry in which it's measured (Chen et al. 1997 [11]). Because the influence of the viscous forces on the shape of the object, the measured dynamic contact angle is not guaranteed to be the actual contact angle.

There are also many other factors that may influence the magnitude of the contact angle, *e.g.* the physical properties of different materials, the shapes and surface roughness of the solid wall, the environmental temperature and the pressure etc. Some further studies showed different surface effects and the environmental parameters on the contact angle. Hocking (1976) [24] examined a fluid displaces another on a rough surface by using an effective slip coefficient. Benintendi and Smith (1999) [3] studied the effects of the slip coefficient on the spreading of a thin axisymmetric liquid droplet under certain temperature gradients. The behavior of the contact angle on extremely hydrophilic surfaces or highly hydrophobic surfaces had also been studied by Partankar 2003[46].

According to our knowledge, because the value of the contact angle is sensitive not only to the physical properties of the nearby materials but also to the mutual interactions between interfaces, a generic data of contact angles are usually very hard to obtain. In our studies, the detailed mechanism of fluid flow in the contact-line region is not considered. A simplified macroscopic model is introduced into the *FronTier* code to simulate the propagation of the boundary node.

3.1.1 Thermodynamics

Since fluid flows in which the contact angle plays an important role involve the complicated free-boundary physics, knowing the contact angle is often the key to understand the flow details at the vicinity of the contact line, to specify the tangential surface tensions, and to establish a proper equilibrium system of momentum and mass conservations around the contact line.

For fluid flow containing multiple components, the contact angle can be a set of angles which describe the relative position and strength between interfaces along the contact line. In general, for a three-phase flow, there can be at most three contact angles at a given point of the contact line. In such case the three contact angles can be addressed in the same cross section, and the number of unknown contact angles is reduced to two according to the basic triangle rule.

A special case of the contact angles is on the boundary where two fluids intersect with at a solid wall. Since the solid wall is always fixed, only one angle is needed to describe the interfaces around the boundary contact line.

Currently, most experimental and theoretical studies about the contact angle are based on the angle between a physical wave and a solid wall boundary. The contact angle along the boundary is the topic of our study.

The classic Young-Laplace's formula(Equation 3.1) studied the static contact angle from the consideration of a thermodynamic equilibrium between the following three phases: the liquid phase, the vapor phase of the ambient(or another immiscible liquid phase) and the solid phase of the substrate (Finn 1986 [18], Neumann et al. 1996 [43]).

$$cos\alpha = \frac{\gamma_{sv} - \gamma_{ls}}{\gamma_{vl}} \tag{3.1}$$

where γ_{vl} is the surface tension between the liquid and the vapor, γ_{ls} and γ_{sv} denote the solid-vapor interfacial energy and the solid-liquid interfacial energy respectively.

Young's formula describes a simple relation on the shape of the liquid-vapor or liquid-liquid interface away from the solid-vapor and solid-liquid interfaces, which theoretically establishes an equilibrium condition on fluid flow problems involving multiple components.

After the contact angle at a boundary node is known, the new position of the boundary node can be specified accordingly and the influence of the contact angle on the conservation of the system can be evaluated by introducing a pressure jump into the momentum and energy equations. This pressure jump can be written as a function of the surface tension, the local mean curvature and the contact angle.

3.1.2 Dynamic Contact Angle

As we have known, the speed of the contact line has significant influence on the behavior of the contact interfaces . The contact angle used to describe the intersections in the contact line region under dynamic conditions is called the dynamic contact angle. Currently, there are two methods to study the dynamic contact angle.

One approach is to assume a constitutive relation between the static contact angle and the contact-line speed. For example, Hocking's (1995 [25]) established a relation between the dynamic contact angle and the static contact angle according to some fitted functions from the experimental data on non-evaporating system.

$$\alpha^3 - \alpha_0^3 = \frac{72\mu U}{\gamma}. \tag{3.2}$$

where α is the dynamic contact angle, α_0 is the static contact angle, μ is the

dynamic viscosity of the liquid-phase, and U is the velocity of the moving contact line.

This cubic function connects the mathematical description of the outer region of the flow field to the contact-line speed and roughly establishes a macroscopic relationship between the dynamic contact angle, the static contact angle and other physical parameters. The detailed micro-physics in the contact-line is masked and not required in computing the flow of the bulk droplet.

Another approach is to assume that the microscopic contact angle is always equal to its static value. For instance, according to Neumann et al.(1996 [43]), in a water drop moving along a horizontal plate, the measured dynamic contact angles for the flow with relatively low velocities are equal to static contact angles. In cases when the three-phase contact curve is inactive or moving in a very low speed, the contact angle can be treated as a function of the properties of the material and the local temperature which avoid using any dynamic property of the contact line.

As we mentioned in the previous section, the detailed studies on the dynamic contact angle is one of the most challenging research. Up to now, the experimental techniques to obtain the dynamic contact angle is very limited especially when the curve speed is large. In our studies, some simplified functions were adopted to test the implementation of our contact angle algorithm and the exact input of these dynamic angles is left for future studies.

3.1.3 Boundary Node Propagation

The contact angle exists only when more than three phase intersections occur at the same location. In *FronTier*, the interface points on such kind of contact line are either called CC-node (Figure 3.1 Left, Curve Crossing node) or B-node (Figure 3.1 Right, Boundary node), where CC-nodes are interface points between three physical

waves, and B-nodes are interface points between a physical wave and a non-physical boundary.

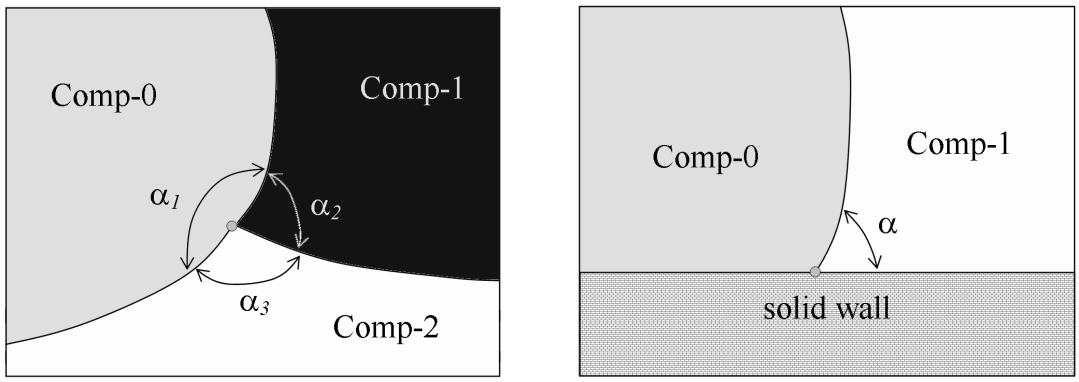


Figure 3.1: Two special nodes in Front Tracking. Left: Curve Crossing node where comp-0, comp-1 and comp-2 are all fluids and the contact angles α_1, α_2 and α_3 are changeable. Right: Boundary node where one of the three phases is the solid wall and only one contact angle α is needed.

As discussed in the previous sections, B-node is actually a special kind of CC-nodes and only one contact angle is needed to describe the relative relations between three interfaces.

In numerical simulations using *FronTier*, the computations with regard to B-node often involve some complex grid cells in which several interfaces may be tangled and need some special restructuring. The complex flow details within the contact-line region are neglected and a macroscopic contact angle is applied to simplify the problem.

Prior to the introducing of the contact angle into the *FronTier* code, there are two other methods to study the B-node propagation. see Figure 3.2

In Figure 3.2, if we assumed that A, B, C, and D are all interface points on the physical wave. At each time step, all the new position of these points (*i.e.* A', B', C', and D') are computed by solving the wave propagation equations(i.e. Riemann

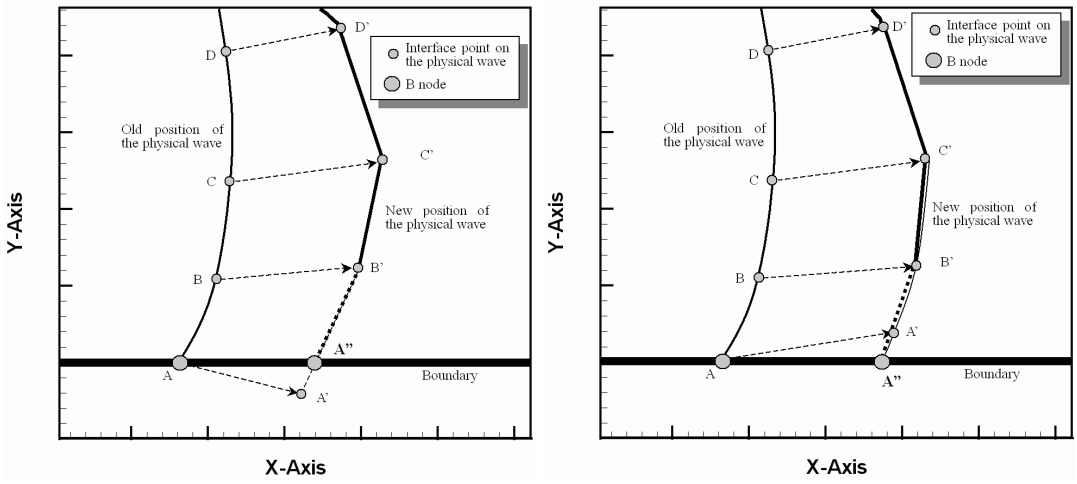


Figure 3.2: Existing ways to handle two-dimensional B-node propagation. Left: Find the crossing between the boundary and the computed interface. Right: Extend the computed interface to intersect with the boundary.

solution). However, the resulting point A' is just a temporary position, the final new position A" of the B-node will be specified according to the relative position of A' to the boundary line.

Two basic ways are applied in *FronTier* to specify the new position of a B-node:

1. If A' is outside the boundary, as in Figure 3.2 Left, then a straight line between A' and the closest interface point on the physical wave B' is made. The intersection between this straight line and the boundary is the new position of the B-node, *i.e.* A".

2. If A' is inside the physical boundary, see Figure 3.2 Right, there are several ways to extend the physical wave towards the boundary by using the existing nearby interface points to find the new position of the B-node A":

   - Extend the straight line along A' and B' and the intersection between this line and the boundary.

- Draw a circle by using three nearby points, A', B', C', and find the intersection between the circle and the boundary.
- use Huyghens' principle to the wavefront of a propagating wave of light.

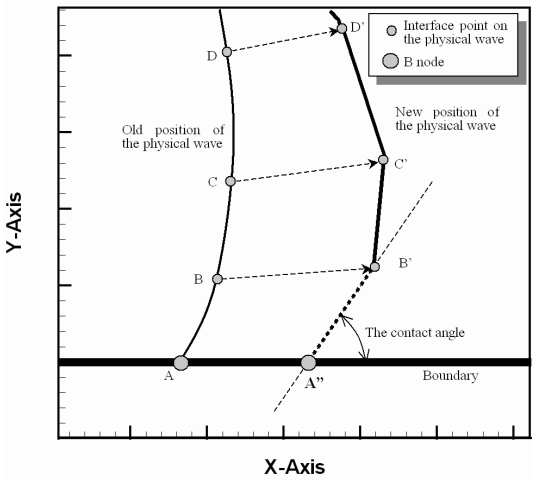


Figure 3.3: The new position of a propagated B-node A" is determined by drawing a line through the closest interface point (B') on the physical wave whose slope is the contact angle away from the boundary line, and finding its intersection with the boundary.

If we use the method of the contact angles, the temporary point A' is not necessary and the new position of the B-node must be obtained by using a geometric method instead of solving the physical equation. See Figure 3.3, for a two dimensional problem, given the value of the contact angle and the position of the closest interface point on the physical wave B', a straight line which passes through point B' and takes a value of the contact angle counterclockwisely away from the boundary line is made. The new position of the B-node is the intersection between this straight line and the boundary.

This method is exactly a simplified method to apply the contact angles since instead of using it on the contact line, we use it on a nearby interface point of the

physical wave. The drawback of this method is that it may introduce big errors when the simulations are based on some coarse grid or where the shear forces around the boundary have a sharp gradient.

3.2 Computational Results

There are many situations in which the contact angle plays a significant role. Here we applied the above contact angle method on the B-node propagation into several applications such as water drops onto a horizontal plate or a slanted plate, a thin film problem of the jet where a gravity-driven flow runs into a plane solid boundary.

3.2.1 Droplet Problems

The spreading of a water drop of volume V on a flat solid plate is a commonly used test case in studying the contact angle. Here we assumed that the surface of the plate is smooth enough so that we do not need to consider the roughness and surface resistance. Both the temperature and the pressure are under the standard environment.

The simulation is based on a two-dimensional 100×100 grids. Initially the water drop starts from a semi-circle and then spreads and changes its shape under the influence of the gravity. Because of the geometric symmetry only half of the semi-circular droplet is studied and the reflecting boundary condition is applied on the corresponding left boundary of the horizontal direction. The boundary condition for the solid wall(plate) is a Neumann boundary.

At initial time, there is one B-node as shown in Figure 3.4 and at this point the water drop, air and the solid wall intersect.

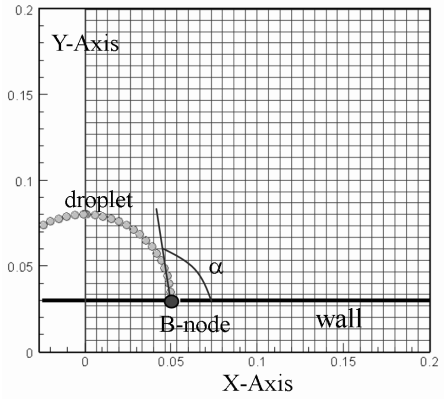


Figure 3.4: The contact angle exists at which the droplet, the air and the horizontal wall intersect. The blank area (no grid) in the left hand side is where the reflecting boundary condition are applied.

By Dussan's data (1979 [15]), the static contact angle α can be calculated upon getting the instantaneous radius r of the circular area where the drop wets the solid along with the volume V of the drop:

$$\frac{r^3}{V} = \frac{3(1+\cos\alpha\sin\alpha)}{\pi(1-\cos\alpha)(2+\cos\alpha)}. \tag{3.3}$$

Since the droplet moves very slowly, we applied the static contact angles instead of the dynamic contact angles in our simulations. We tested the values of the contact angle by using the above formula and compare them with the result by the B-node propagation. If we express r as the inverse of the local mean curvature at the B-node, the computed contact angle for the droplet is $1.5889(91^o)$ and the results are reasonably close to the real contact angle within some initial time and the computed contact angles are approaching to a fixed values *i.e.* 270^o. This may be introduced by the functions we used to compute the local mean curvature at B-node. The result from this simulation can be shown in Figure 3.5.

We further simulated a water drop spreading along a slanted plate. Figure 3.6

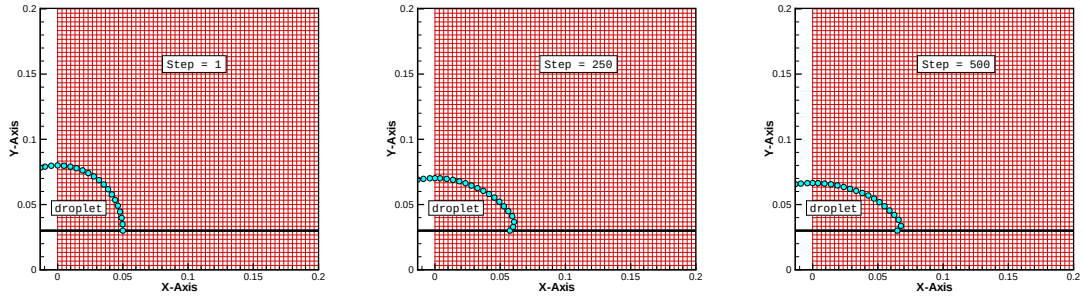


Figure 3.5: A droplet spreads on a horizontal plate at three different computational iteration steps(1, 250, and 500).

shows the boundary conditions we applied on this simulation. Both upper boundary at x and y directions are Dirichlet flow-through boundary and the lower boundary at the x direction is Reflect boundary condition. Newmann boundary is set on the solid wall.

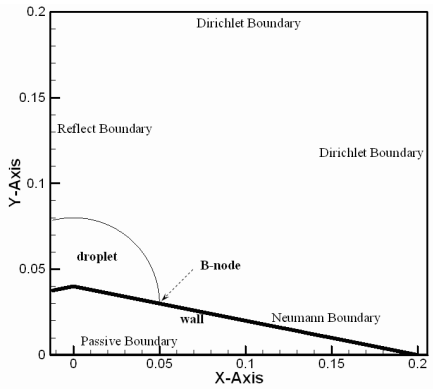


Figure 3.6: The boundary conditions to simulate a droplet spreading along a slanted plate.

Figure 3.7 shows the computational results of the above simulations at different time steps. Compared with the results from the original methods in *FronTier* code (see Figure 3.8), the results from the contact angle method takes nearly a constant

tangential angle at the boundary node while the original method takes a smaller angle. The overall positions and the shapes are roughly the same within a certain time.

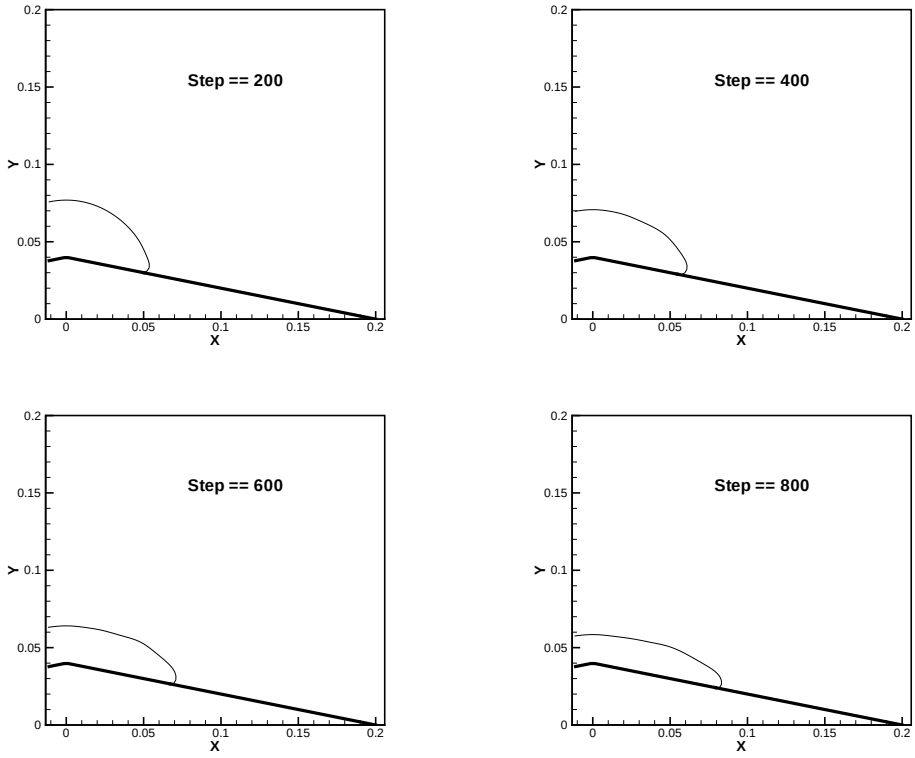


Figure 3.7: The numerical results of a droplet spreading on a slanted plate at four different computational iteration steps (200, 400, 600 and 800) by using the contact angle method to handle B-node propagation.

Figure 3.9 shows an elliptic water droplet falls from a certain height down to a slanted plate and then spread around. The shapes and the volumes of the droplet during the falling and spreading are reasonable from the qualitative point of view.

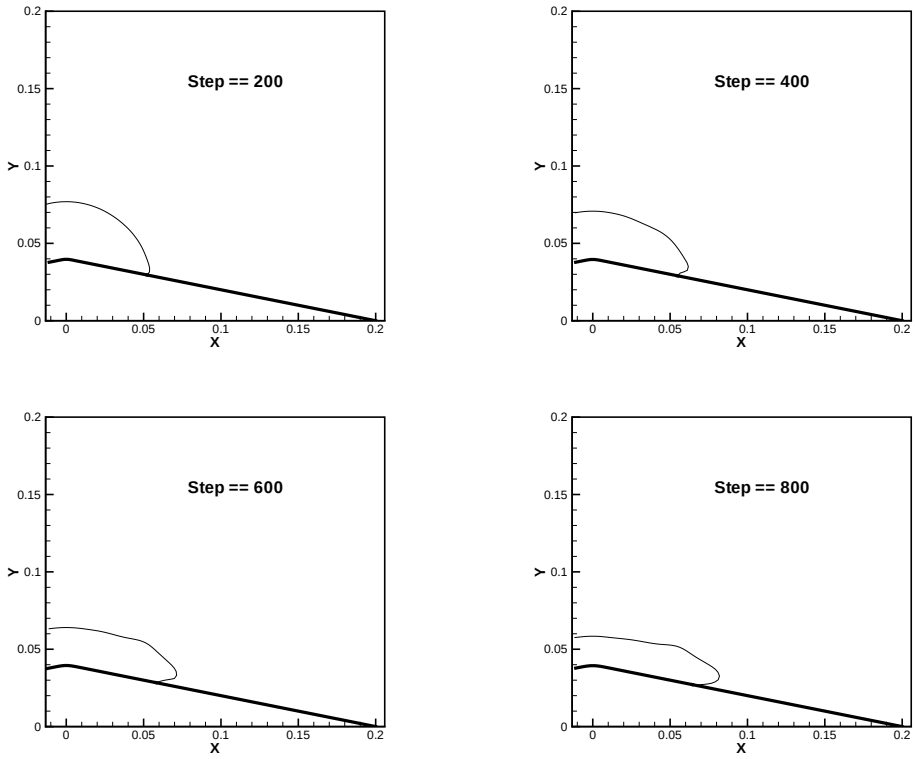


Figure 3.8: The numerical results of a droplet spreading on a slanted plate at four different computational iteration steps (200, 400, 600 and 800) by using the original method to handle B-node propagation.

3.2.2 Thin Film Problems

We applied the contact angle method to solve the B-node propagation in a thin film problem. See Figure 3.10 Left, water at 20^oC flows from the jet with the flow-rate of $1.7gr/s$ and then hits a slanted plate.

We applied four different grids and structures to initialize this problem, see Figure 3.11.

In case-1, the jet outlet is simplified to a Dirichlet flow-through boundary flow. The fluid at the nozzle keeps constant physical parameters throughout the simulation.

In case-2 and case-3, a full-structured jet is applied. The initial interface be-

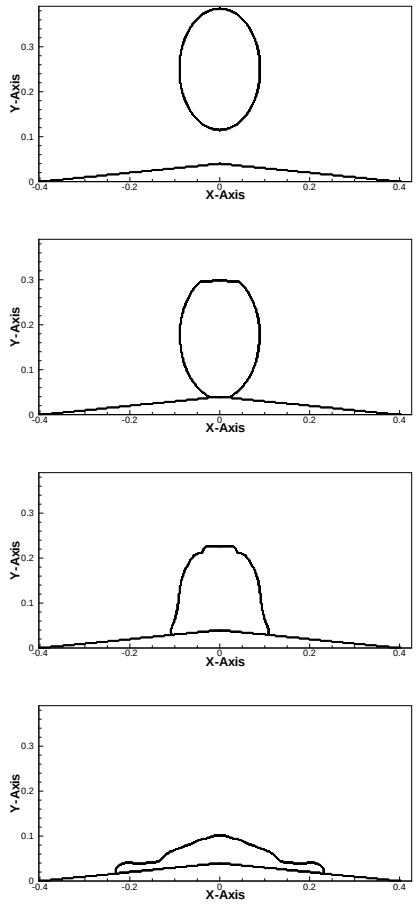


Figure 3.9: The numerical results of a droplet falling from a certain height and then spreading onto a slanted plate at four different computational iteration steps (0, 1000, 2000 and 3000).

tween the water within the jet and the ambient air is either a straight line(case-2) or an extended elliptic curve(case-3). In case-3, considering that the contact angle is difficult to specify at the corner, we expanded one grid-cell out and make an elliptical curve to form the water-air interface, see Figure 3.12.

In case-4, the same full-structured jet had been adapted but with a smaller distance between the jet nozzle and the slanted plate.

For all four computational cases, Neumann boundary is applied on the slanted

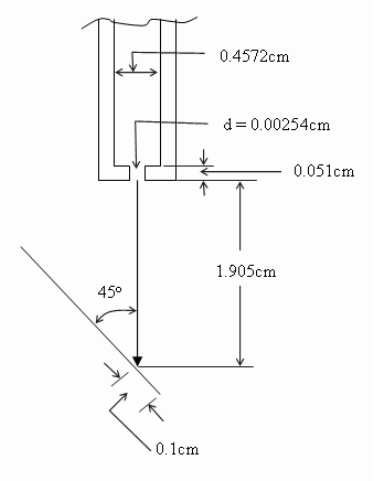


Figure 3.10: Geometric description of the thin film problem.

plate and the flow-through Dirichlet boundary condition is applied on all four computational boundary. For case-2,3,4, the Newmann boundary is set on the solid wall of the jet nozzle.

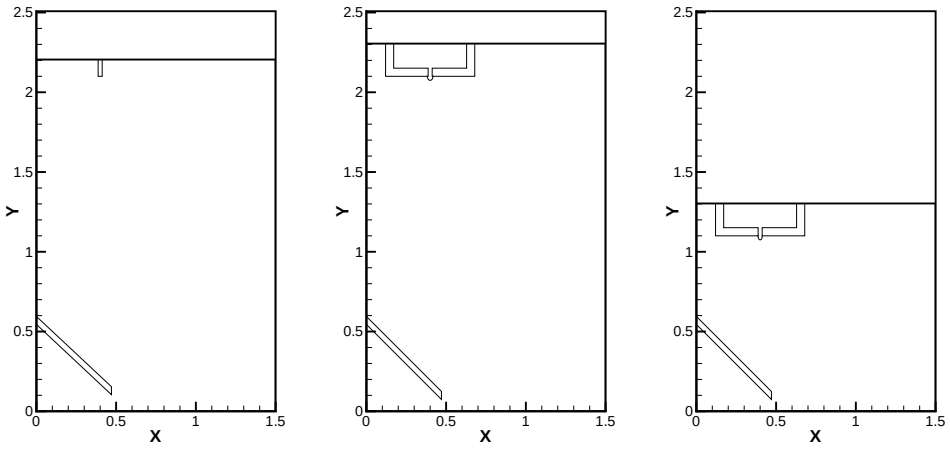
The computational grid applied in the simulations are 450×660 for case-1,2,3 and 300×460 for case-4.

The computed velocity vector and density contour for case-1 has been shown in Figure 3.13. Several vortex structures are well captured from the Figure 3.13 Right, and the density plot shows how the water interacts with the slanted plate.

When full-structured jet is applied as in case-2 and case-3, the results are quite different from that of the case-1. This may come from the B-node treatments on the jet nozzle which introduce the geometric bifurcations and the contact angle can not normally be handled, see Figure 3.12.

From Figure 3.14 and Figure 3.15, the density contours of case-2 and case-3 are much similar, both cases have more back-flow than those of case-1 and case-4.

When the distance between the jet nozzle and the slanted plate becomes smaller(case-



(a) case-1 (b) case-2 and case-3 (c) case-4

Figure 3.11: Initialization of the thin film problem. Left: the jet nozzle is treated as a flow-through Dirichlet boundary. Middle: full-structured jet with Newmann boundary on the jet nozzle. Right: full-structured jet with a smaller distance between the jet and the plate.

4), the back-flow at which the water hits the plate is reduced see Figure 3.16.

3.3 Summary

In our simulations, we applied some empirical data and simplified functions to simulate the propagation of the boundary nodes. Since the physical contact angles on the complex geometric structures are still not available, the current method is mainly developed as a model of computational interface which will take physical-based input to determine the specific contact angles in the later simulations.

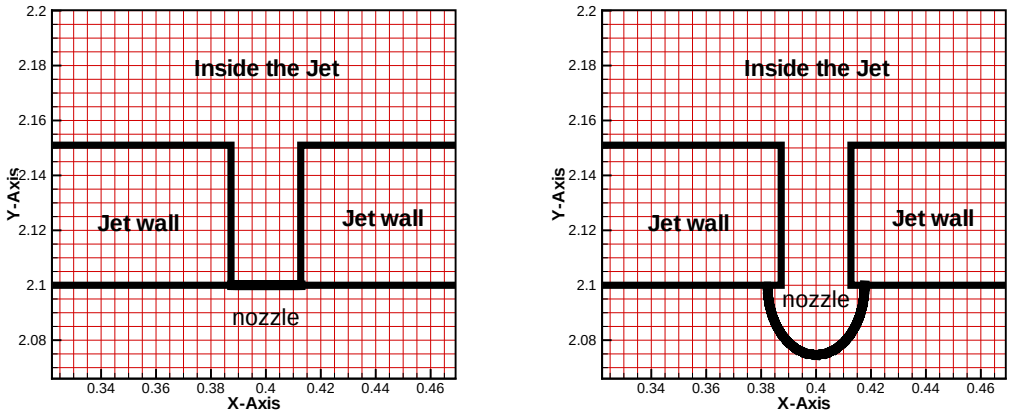


Figure 3.12: Different treatments of the interface at the jet nozzle. Left: case-2 where the initial interface at the jet nozzle is a straight line. Right: case-3 where we extended one more computational grid away from the jet nozzle wall and used an elliptic curve as the initial interface.

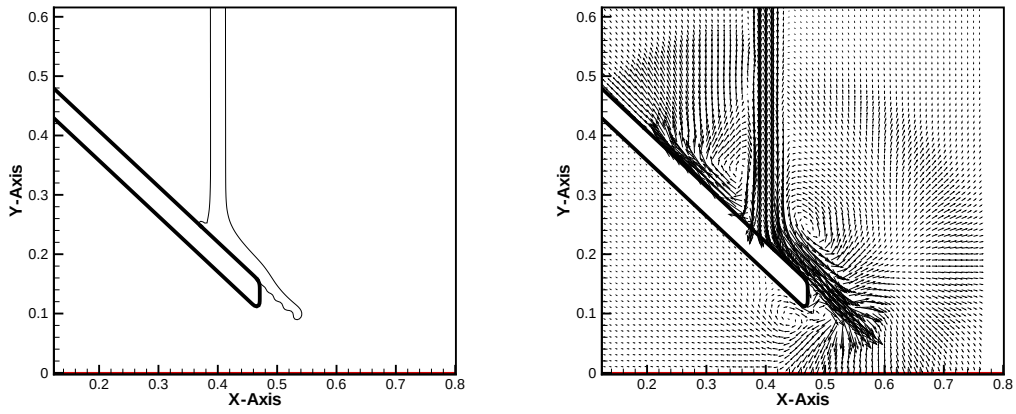


Figure 3.13: The material interface(Left) and the velocity vectors (Right) computed by the contact angle method(case-1). The water at the impact region has certain back-flow towards the upper part of the plate.

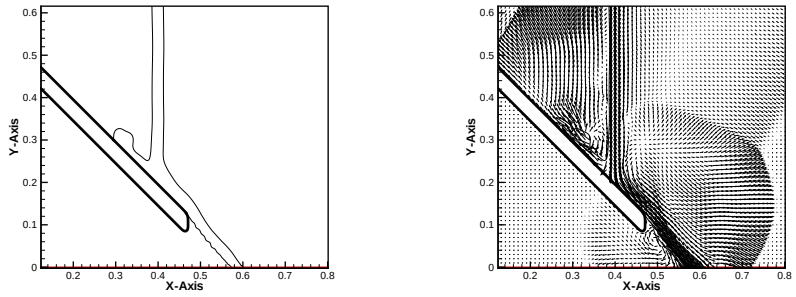


Figure 3.14: The density contour(Left) and the velocity vectors (Right) computed by the contact angle method(case-2).

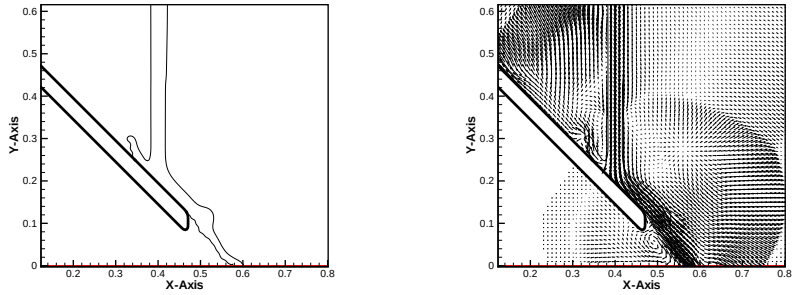


Figure 3.15: The density contour(Left) and the velocity vectors (Right) computed by the contact angle method(case-3).

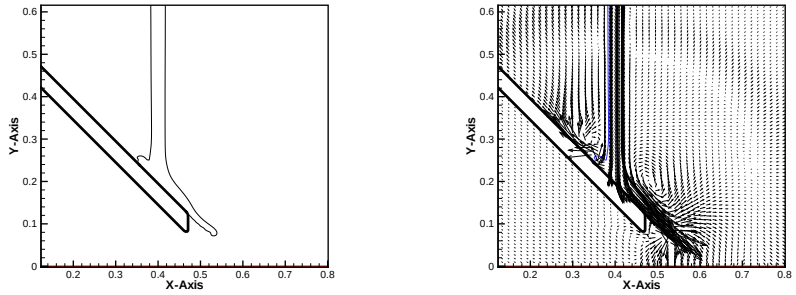


Figure 3.16: The density contour(Left) and the velocity vectors (Right) computed by the contact angle method(case-4).

Chapter 4

Cell Migration Problems

Here we developed some algorithms to simulate the behaviors of the moving cells with the *FronTier* code. Our algorithms are based on the mathematical and physical principles and combine with some observations from the biological experimental results.

By simulating different behaviors of cell migrations, we are hoping to characterize and quantify some typical processes of the cell migrations with Front Tracking method and help predict some medical activities in tumor progression and drug production et. al.

4.1 Introduction

Cell migration takes a central role in studying embryogenesis, immune response, tumor growth, and wound healing. Tissue formation and development during these processes all require the movement of some specific cells in a particular direction to a specific location. Errors during these processes may have serious consequences, including vascular disease, tumor formation and metastasis et al. Understanding the mechanisms of cell migrations is therefore of great importance for the development of some new therapeutic strategies, and the cellular transplantation.

Cell migration is a multidisciplinary research area which involves knowledge of biology, biophysics, biochemistry, immunology, neuroscience et al. Recent technical advances, particularly in computer hardware and computational biology, have extended the studies of the cell migrations into the world of numerical computation.

Currently, most of the research work on the cell migration is in the experiment and analysis of mechanism. For example, the polymerization of actin filaments has a well established role in driving the formation of protrusions. Meanwhile, most cells move and respond to chemo-attractants, chemical gradients, gradients of substrate stiffness, et al., and their direction of movement is determined by these gradients. Many studies also focused on the molecular mechanisms which regulate cytoskeletal dynamics and the adhesion, both take a key role in changing cell shape and motility.

The computational simulation of cell migration is very new compared with the experimental researches. Since the real cell migration is a very complex precess, in reality, most of the moving cells are fibroblast-like and change directions and shapes often and unpredictably, therefore to reproduce an exact cell migration is less possible. On the other hand, many cell migrations do follow certain rules. For example, some specific cells response to specific signals, and thus move towards them, both the strength and distance of these signals make a contribution to the moving velocities and directions.

Recent numerical simulation on the cell migrations involves the studies of the moving directions, trajectories, and the change of cell shape and size. Ray et al.(2002 [51]) applied an energy-based model to constrain the active cell shapes, sizes and positions in simulating a moving leukocyte. Agresar et al. (1998 [1]) applied an adaptively refined mesh and immersed boundary method to test the deformation and adhesion of the cells under different mechanic models.

In our study, we have tried to explore some typical behaviors of cell migrations,

and we care more about the overall moving velocity, the position and the direction of some specific types of cells. we also applied some algorithms to keep the volume roughly conservative, but the exact shape is not our concern at this time. The Front tracking method and the *FronTier* testfront code are applied to capture the interface of the cell membrane. The current study applies to geometric and velocity models.

4.2 Cell Migration Mechanisms

In order to migrate, cells have to attain energy from either intracellular or extracellular sources. Basically, a moving cell has its driven potential to move towards a specific direction, and is influenced by its living environments and obstacles in its path.

For example, a normal leukocyte moving in a blood vessel, will be influenced by the blood-pressure in the blood vessel and the interactions with white cells. When there is a wound somewhere on the body, leukocytes in the circulation migrate into the surrounding tissue to destroy invading microorgans and infected cells and to clear debris.

The mechanism of how leukocytes overcome the outside resistances and move intentionally towards the places where wounds occur is therefore the main topic we are trying to explore in our studies.

4.2.1 Thermodynamics

The basic processes of cell migrations satisfy all three laws of thermodynamics. We predict that a white cell with higher potential energy and living closer to the wounds can arrive the destination easier, while a white cell with low potential energy and farther to the wound may never be able to reach the destination.

We start our research from the mathematical and physical principles. If we neglect the influence of heat transfers in cell migration processes, and at this time, if we do not consider cell-cell and cell-boundary interactions which are considered as the outside-work in the first law of thermodynamics. All we need to consider is the change of internal energy which finally drives the cell migration.

The internal energy change can take the following forms:

- all possible chemical reactions and biologic processes(*i.e.* polymerization) inside the cell which may produce or consume energy.
- positional potentials.
- kinetic energy.
- potentials from the pressure gradients.

For the cell migration problem, the driving sources can be listed as the following:

- The polymerization and de-polymerization of actin filaments produce the absolute free energy which drives the cell motion.
- The motor proteins convert the chemical energy of nucleotide hydrolysis into mechanical works, which also produce energy to drive the cell.
- Polarized network of actin filaments exists, especially in the leading and trailing edge, which influences the local magnitudes of the driven forces, and further steer the cell motions.
- Elastic force on the membrane which helps the cell to keep its shape.
- The chemical gradients in the living environment.

4.2.2 Polymerization and Driving Horses

Polymers preform many tasks in living cells. Oster et al.(2003 [37]) discussed their rules in two general aspects:

1. Polymers can convert the binding free energy of their constituent monomers into mechanical force to generate a pushing axial load, while de-polymerizing microtubes can develop pulling forces.

2. Polymers can also store elastic energy during their polymerization that can be released later to generate the forces.

Polymerization can be considered to be the driven horses which produce the absolute free energy and finally moderate the intuition of the cell to move towards a certain direction. Certain kinds of cell crawling are apparently driven by actin polymerization. According to Mogilner(1996 [40]), this force can be approximated by a stochastic model. Theriot et al.(1992 [57]) suggest that the actin growth drives the bacterium forward.

Since the polymerization and de-polymerization are directly related with the working filaments in the cell actin network, and both the number of working filaments and the magnitude of the process are uncertainty in any specific location and at different time, we introduced some randomized terms in our velocity model to simulate these effects.

4.2.3 Polarization and Sense of the Directions

In order to move to a particular direction, cells have to sense extracellular chemo-attractants through specialized receptors on their cell membranes. These receptors may take active role in regulating the distribution of the actin filament network

and the local activities of the polymerization and de-polymerization processes. The polarized actin filament network in return triggers signaling cascades that lead to cell movement.

Watanabe et al.(2005 [58]) showed that a migrating cell is highly polarized and contains complex regulatory pathways downstream of various stimuli. Cell polarization includes the asymmetrical distribution of signaling molecules and the cytoskeleton, in addition to directed membrane trafficking.

To polarize the moving cell, we divided the cell into two parts: the leading edge and the trailing edge. The leading edge can be defined as the protrusion part of a cell. Even though the ratio of the leading edge varies according to the types of a cell, the protrusion part is typically $2 - 5\mu m$ which is $10 - 50\%$ of the cell boundary.

The polarization model in our method is based on a signal process which means for a given signal, there are a corresponding leading edge and a trailing edge. The cell responds to the signal according to the signal strength and its distance away from the signal, and thus different signals take different influential factors on the moving direction and velocity.

The division between the leading edge and the trailing edge is based on the distance between the signal and the cell membrane. Currently, we set 30% of the closest points on the boundary as the leading edge section and others as the trailing edge section. But we can change the portion of the protrusion part as much as we want.

4.2.4 Adhesion and Cell Crawling

Some cells migrate by first extending a protrusion at the leading edge. Once the protrusion is stabilized, *i.e.* by attaching to some surrounding substrate, the rear of the cell then releases its attachments to the substrate and contracts. This

kind of movement involves cell adhesion, and the cell must constantly switch between attaching and relinquishing their bonds with the basement membrane.

By applied a time-dependent velocity function on our model, we can simulate such kind of cell movement. The basic rules are:

1. Given a time interval or computational step interval.
2. At a certain time interval, move the leading edge while keep the trailing edge unmoved.
3. At the next time interval, move the trailing edge while keep the leading edge unmoved.
4. repeat step 2 and 3

The time intervals can be a variable which make the crawling more complex.

4.2.5 Shape and Size of Moving Cells

A moving cell, when treated a bulk of uniform materials enclosed by a closed surface, maintain a rough volume conservation in motion. The shape and the size thus need to change smoothly for the sake of the conservation of the overall volume.

As the uncertainty impacts of the intracellular polymerization on the cell membrane, the exact shapes of the moving cell are very difficult to track and therefore is not the focus of our current simulations.

4.3 Front Tracking and Cell Migration Models

Living cell, as a whole, can be simplified as a bulk of fluid wrapped by an envelope. The possible complex interface tangling has been handled properly by the *FronTier*

code. With the versatility of *FronTier* testfront code, we are trying to extend the application of Front Tracking method into the field of cell migration.

As the same with all the other numerical simulations, we phased out some nonessential factors to the cell migration, and made several assumptions:

- we neglected the internal complexities within the cell body, and treat the cell as an uniform component of a certain material.
- Since the chemical reactions and biologic processes has a big randomized factor, we applied some simplified stochastic models in our method to replace these complexities.
- Heat transfers between cell and its surrounding and inter-cellular interactions are also neglected at this stage.

we can further divide the force fields on a single moving cell into two categories:

- Inside the cell: polymerization and de-polymerization of actin filaments network which produce the energy to move, and elastic forces on the membrane which help to keep the cell shape.
- Outside the cell: chemical gradients and pressure gradients which drive or block the cell movement, and Chemo-attractants which emit signals to moderate cell polarization and influence the cell migration indirectly.

Our model is per-signal based, for multiple signals, we will apply a different strength and influential factor on each signal and then combine them accordingly.

4.3.1 Velocity Models

The *FronTier* code has a general purpose software library, which consists of the basic geometric functions of *FronTier*. The library, applied on a testfront code, can

simulate a moving object by a user-customized velocity function.

After we have established the force or energy conservation equations system, we need to map them into velocity and then implement in the testfront code.

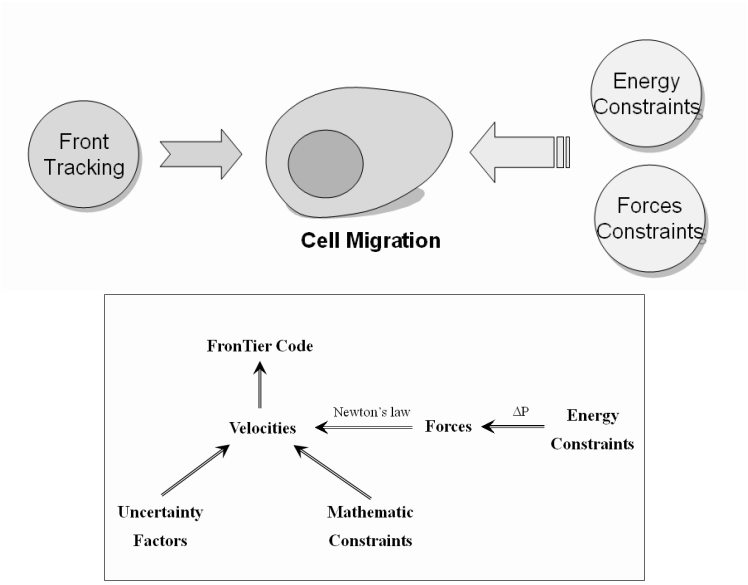


Figure 4.1: Using Front Tracking method to simulate Cell Migration Problems

Volume Conservation

Figure 4.2 show the change of volume/area of a moving cell with respect to the time step under different force fields. An increase of the cell volume/area can be found when the cell starts moving and the volume/area decreases when transferring into a changed force field.

From the mathematical point of view, to keep the volume conservative, the following relation should be satisfied:

$$\int_{\partial\Gamma} (\overrightarrow{v} \cdot \overrightarrow{n}) d\Gamma = \sum_{i=0}^{n} \left\{ v_{ix}^{n} \big(x_{i+1} - x_i\big) + v_{iy}^{n} \big(y_{i+1} - y_i\big) \right\} = 0 \tag{4.1}$$

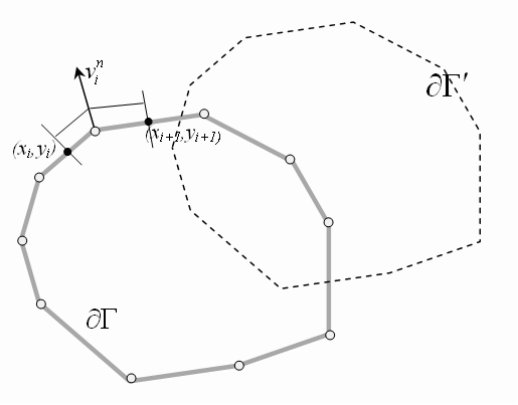


Figure 4.2: Mathematical Expression of the Volume Conservative

The summation of the multiplication between the normal component of the velocities and the length of local curve along the closed cell interface vanishes. Where (x_i, y_i) and (x_{i+1}, y_{i+1}) are the coordinates of the middle point of adjacent bonds to the given point, v_{ix}^n and v_{iy}^n are x and $y-$components of the normal velocity at the local point. n is the index of the point on the interface.

In our method, we adopted a simplified equation. Since we assumed that the density inside a cell is uniform the volume conservation can be expressed by the velocity along the boundary. By Stokes' theorem,

$$\int_{\Omega} dv_n = \int_{\partial\Omega} v_n \tag{4.2}$$

where Ω is the cell domain and $\partial\Omega$ is the cell boundary, v_n is a velocity component in the normal direction at a point on the boundary. To conserve volume we need the constraint

$$\int_{\partial\Omega} v_n = 0 \ . \tag{4.3}$$

Since we have discrete finite points on the cell boundary we have instead of

Equation 4.3

$$\Sigma_p v_n = 0 \tag{4.4}$$

where p's are points along the cell boundary.

In our method, we apply an velocity repository at each time step to contain the overall normal velocities which conserve the volume according to the computational points on the cell membrane.

1. assume the signal take a strength of 1.
2. before each time step, we generate a repository of velocities, and divide them into two part: head part(take positive number), and tail part(take negative number)
3. the velocity sum of head part and tail part are 1 and -1 respectively.
4. whether the interface points belong to head part or tail part are decided by their relative position from the signal.
5. at each interface point, we dish out a velocity value from the repository(step 2) as the absolute velocity of this interface point accordingly.

Figure 4.3 shows how a randomized number generator works out with some given number of random vectors. In our computational cases, we use this model to generate random velocities on the cell membrane.

We compute the random walk using our model for the center of mass of the cell. Because the mass is uniformly distributed, we do it by averaging the coordinate in each direction. At the boundary, each point propagates with the velocity calculated by:

$$v_n = v_n^{random} + v_n^{curvature} \tag{4.5}$$

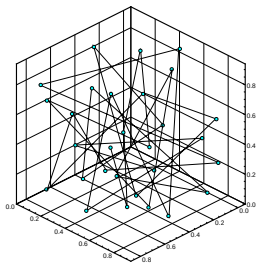


Figure 4.3: Randomized Number Generator in a Three-dimensional Matrix

between the point and the signal, the normal velocities with random magnitude in the leading edge are moving outwards the cell and those in the trailing edge are moving inwards the cell. Since it is hard to take both velocities into account at a time satisfying 4.4 we consider these two types of velocities(outwards and inwards) separately.

Mathematical Conservation

Certain constraints based on the displacements are applied to keep the size and shape of the moving cell. Mathematically, we can keep strict conservation of volumes by applying a coefficient α to each of the interface points on the cell membrane, see Equation 4.6:

$$\alpha = \sqrt{\frac{A_{new}}{A_{original}}} \Rightarrow \begin{cases} x_{re} = \alpha \cdot x_{new} \\ y_{re} = \alpha \cdot y_{new} \end{cases} \tag{4.6}$$

where A_{old}, A_{new} are the volume of the original interface and new interface respectively, x, y are the coordinates of the points on the interface. Subscript re means redistribution, new means the newly propagated point, and $original$ means the initial values.

4.3.2 Force Models

The velocity model is in fact a mathematical model which is established on a stochastic model and geometric conservation relations. To make our simulation physically reasonable, we need to add force and energy equations.

According to Mogilner and Oster's model(2003 [38]), the force and velocity relationship based on a single filament can be expressed in the following form:

$$V = V_{max} exp[-f_w l / k_B T] - V_{dep} \tag{4.7}$$

where $V_{max} = k_{on} \cdot l \cdot M$ is the free polymerization velocity, and $V_{dep} = k_{off} l$ is the de-polymerization velocity. Here k_{on} and k_{off} are the rate of monomer assembly and de-assembly respectively. $f_w l$ is the work done by a filament against the load during one act of assembly, k_B is the Boltzmann constant, and T is the absolute temperature.

This equation establishes a relation between the force $f_w l$ and the velocity of the interface V. The force produced by a filament can be obtained either by a force balance relation, or by the stochastic analysis.

According to our understanding, the direction of the cell migration is determined mainly by the surrounding gradients or signals from outside the cell. A cell heading to this direction has denser filaments network along with the heading than the trailing edge of this directions. When the polymerization and de-polymerization occur within the filament network, the growing filaments exert internal pressure on a cell boundary, which combined with the load force, makes the cell move and deform(*i.e.*Boal 2002 [4], Janetopoulos et al. 2004 [32]).

A pressure difference across the boundary membrane gives rise to a lateral tension whose magnitude and direction depends on the local curvature which obeys Young-

Laplace equation. In our case, we set the tangential force F_τ as a function of surface tensions τ , local mean curvature of the membrane r, and velocity V, thus we have: $F_\tau = f(\tau, r, V)$

For a single filament in the network, a force is generated on the cell membrane by the local filaments activities which is a function of the growth rates of the filament. The concentration of filaments inside the cell is not uniform, and we assume, is subject to the gradients and the chemo-attractants, thus the forces from the inside of the cell at any local membrane point can be written as a function of the chemical gradients and signals $F_f = k \cdot \Delta\Phi$, where $\Delta\Phi$ is the local force gradients of the chemicals around the cell. Here k is a time-dependent function which is influenced by the signal information, and its magnitude depends on the number of working filaments and the energy released during the polymerization processes.

To simplify our problem, we divided the main forces into two categories: random force and curvature-dependent force. The random force can be perceived as a microscopic force from the protrusion while curvature dependent force as a macroscopic force from the protrusion force and the traction force to keep the shape of the cell. The retraction force in the trailing edge and the curvature are automatically considered by the volume constraints.

4.4 Results and Discussions

Cell behaves differently in different environments and for different types of cells. Here we started from the simplest case: single cell in vacuo. We apply the methods mentioned in the previous sections and discussed several cell migration behaviors: extracellular gradients, signals and crawling, and three dimensional cells in multiple signals.

All the computation have the initial cell as a circle or sphere and the points on the cell boundary are uniformly distributed. The points are also redistributed at each computational time step.

Both Locally Grid-based and Grid-based tracking methods are used in the simulations, the Grid-based method has a smoother surface but worse in volume-conservation than those using the Locally Grid-based method.

4.4.1 Cell in Vacuo

Figure 4.4 shows a single cell in vacuo where no external force exists. the cell moves in a certain direction under the influence of the internal polymerization and depolymerization. The shapes are changed with time but the volumes are kept roughly constant.

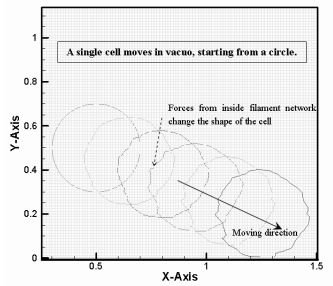


Figure 4.4: A single Cell in Vacuo

4.4.2 Cell under Gradient

Figure 4.5 shows a single cell moves in a simple velocity field (see Figure 4.5-b), where both moving direction and cell shapes and size are changed dramatically. The gradient become the main factor to influence the direction and shape of the cell migration.

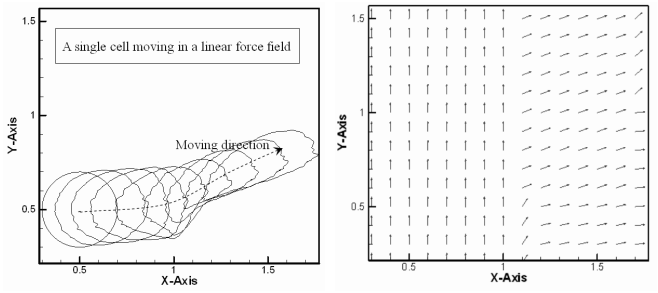


Figure 4.5: A single cell moving in a given velocity field

From this simulation, we found that wherever there exists a sharp change on any given factor(i.e. the external velocity field), the sizes of the cell may change dramatically and thus not conservative.

4.4.3 Volume Conservation

We use the method in §4.3.1 to keep the volume conserved for a moving cell. Table 4.1 shows the results on a two-dimensional moving cell. the computational grid is 300×300, the resulting different ratio of cell area is computed by the following function:

$$differentratio = (A_{computed} - A_{original})/A_{original} \tag{4.8}$$

where $A_{computed}$ is the computed cell area and $A_{original}$ is the original cell area. The different ratio is less than 10% in this computational case.

4.4.4 Mathematical Conservation

We also applied mathematical conservation algorithm in §4.3.1 to test the conservation of the cell volumes. No velocity conservation (see §4.3.1) constraint is

step	computed cell area	original cell area	different ratio
0	0.122576	0.122576	0.0
20	0.120706	0.122576	-1.5
40	0.129846	0.122576	5.9
60	0.116129	0.122576	-5.3
80	0.134256	0.122576	9.5
100	0.119784	0.122576	-2.3
120	0.126642	0.122576	3.3
140	0.122750	0.122576	0.1
160	0.124474	0.122576	1.5
180	0.134443	0.122576	9.7
200	0.133890	0.122576	9.2
220	0.126471	0.122576	3.2
240	0.133154	0.122576	8.6
260	0.133730	0.122576	9.1
280	0.123067	0.122576	0.4
300	0.125283	0.122576	2.2
320	0.124658	0.122576	1.7

Table 4.1: Volume of the Cell after applying Volume Conservation Algorithm

implemented.

A mean-value coefficients function is applied to do interface point-redistribution upon reaching a certain intervals(i.e. every ten computational steps).

Figure 4.6 shows the trajectories of a moving cell before and after we have applied a mean-factors re-distribution. By the constraint of the volume conservation, the volume is almost constant, and the shape of the cell keep approximately the same, the positions of the moving cell are changed.

Table 4.2 shows the computed values of volume and $\int_{\partial\Gamma}(\overrightarrow{v}\cdot\overrightarrow{n})d\Gamma$ for the computational case. The original volume of the cell is 0.125660.

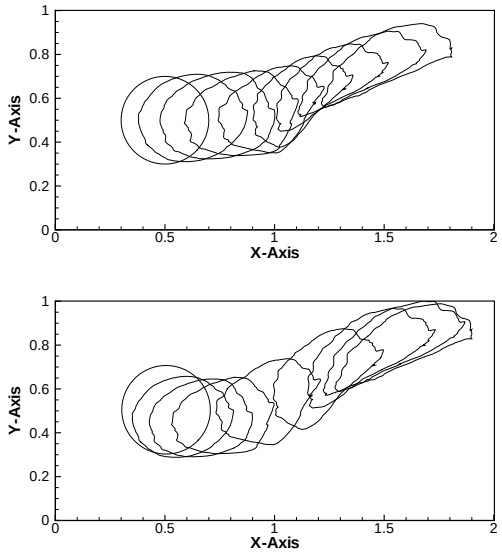


Figure 4.6: Volume Conservative

4.4.5 Cell Crawling

We implemented the method described in §4.2.4 to simulate the cell crawling. The time interval in our computation is a fixed value for each step. From the Figure 4.7, the leading or trailing edges of the cell are moved at a certain time-step while the other part keeps almost unchanged.

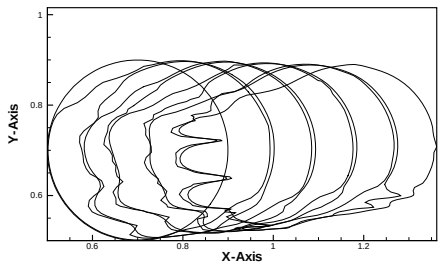


Figure 4.7: Cell Crawling

Time step	Non-redistribution		Mean-redistribution	
	Volume	$\int_{\partial\Gamma}(\overrightarrow{v}\cdot\overrightarrow{n})d\Gamma$	Volume	$\int_{\partial\Gamma}(\overrightarrow{v}\cdot\overrightarrow{n})d\Gamma$
1	0.123190	0.000254	0.125660	0.000256
20	0.146414	-0.000571	0.125660	-0.000366
40	0.155870	0.000226	0.125660	0.000378
60	0.155098	0.000825	0.125660	0.000357
80	0.129160	0.001254	0.125660	0.001360
100	0.103425	0.001886	0.125660	0.002139
120	0.096536	0.006780	0.125660	0.006352
140	0.102967	0.004383	0.125660	0.004948
160	0.113771	0.002534	0.125660	0.003005

Table 4.2: Volume of the cell before and after the redistribution

4.4.6 Three-Dimensional Cell Migrations

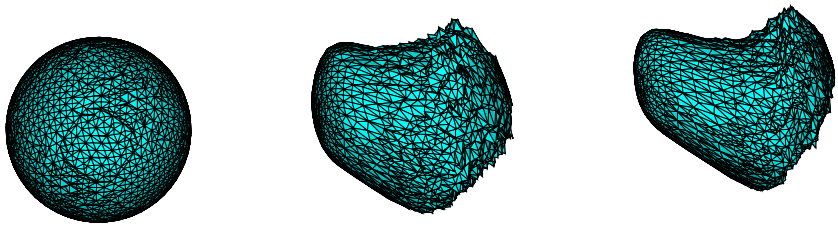


Figure 4.8: Cell attracted by a single signal

Using a similar algorithm, we can extend our velocity model to the simulation of three dimensional moving cells. Figure 4.8 shows a cell moves under the influence of a single extracellular signal.

Figure 4.9 and Figure 4.10 show a cell move under multiple extracellular signals. the deformations are varies according to the strength and distance of these signals.

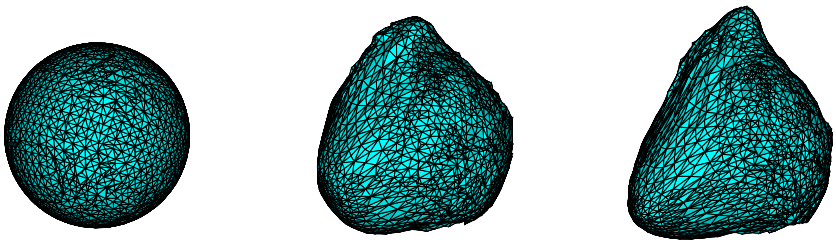


Figure 4.9: Cell Attracted by two signals

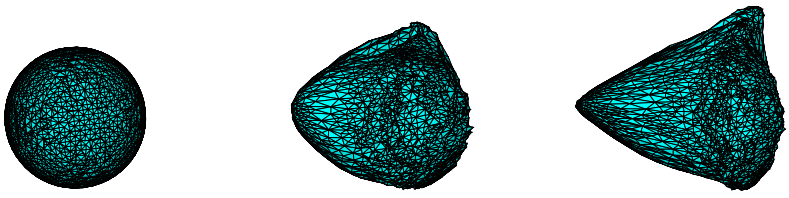


Figure 4.10: Cell Attracted by three signals

Chapter 5

Conclusions

Here we developed several algorithms and presented several applications of the Front Tracking method including: the construction of multiple component grid-based interface, the thin film formation problem, the cell migration problem et. al. From the simulations, we draw the following conclusions:

1. Front Tracking method is an effective way to analyze complex physical problems with moving discontinuities. In our computations, both grid-based and grid-free topological algorithms are applied for different problems.

2. FronTier can be developed into a comprehensive software package and provides a framework and interface to solve different physical and mathematical problems.

3. With several assumptions, the extended Marching Cube algorithm can be used to solve 3-D three-component problems. In some situations, special functions may be needed to handle tangled bonds and surfaces. Further improvement may also include dynamic optimization of the triangulation in constructing 57 isomorphically distinct patterns.

4. The analysis of the contact angle is the key to the simulations of physical problems containing multiple components. Applying the contact angle to the B-node

propagation solver is the first step in understanding the complex physics in the vicinity of the dynamic contact line. The further research can be a reference and extended into the propagation of the Curve Crossing Node which is the major challenge in simulating 3D jet problem.

5. Cell migration is a promising scientific front and the FronTier provides a flexible testfront package which can be used to simulate the cell migration. Understanding the mechanism and using the advantages of the high-resolution interface construction of Front tracking method can be expected to solve more complex biological physics.

Bibliography

[1] G. Agresar, J. Linderman, G. Tryggvason and K. Powell, "An Adaptive, Cartisian, Front-Tracking Method for the Motion, Deformation, and Adhension of Circulating Cells", *J. Comp. Phys.*, 142:346-380, 1998.

[2] D. Banks and S. Linton, "Counting Cases in Marching Cubes: Toward a Generic Algorithm for Producing Substitopes", *IEEE Transactions on Visualization and Computer Graphics*, 10:371-384, 2004.

[3] S. Benintendi and M. Smith, "The Spreading of a Nonisothermal Liquid Droplet", *Phys. Fluids*, 11:982989, 1999.

[4] D. Boal, "*Mechanics of the Cell*", Cambridge, Unite Kingdom at the University Press, 2002.

[5] S. Bohnet, R. Ananthakrishnan, A. Mogilner and J. Meister, "Weak Force Protrusion at the Leading Edge of the Lamellipodium", *Biophysical Journal*, 90:1810-1820, 2006.

[6] D. Bottino, A. Mogilner, T. Roberts, M. Stewart and G. Oster, "How nematode sperm crawl", *J. Cell Science*, 115:367-384, 2002.

[7] R. Braun, B. Murray, W. Boettinger and G. McFadden, "Lubrication Theory for Reactive Spreading of a Thin Drop" *Phys. Fluids*, 7:1797-1801, 1995.

[8] P. Bourke, "Polygonising a Scalar Field Using Tetrahedrons", http://local.wasp.uwa.edu.au/ pbourke/modelling/polygonise/.

[9] C. Bustamante and D. Keller and G. Oster, "The Physics of Molecular Motors", *Acc. Chem. Res.*, 34:412-420, 2001.

[10] Q. Chen, E. Rame and S. Garoff, "The Breakdown of Asymptotic Hydrodynamic Models of Liquid Spreading at Increasing Capillary Number", *Phys. Fluids*, 7:2631-2639, 1995.

[11] Q. Chen, E. Rame and S. Garoff, "The Velocity Field near Moving Contact Lines", *J. Fluid Mech*, 337:49-66, 1997.

[12] M. Chicurel, "Cell Migration Research is on the Move", *Science*, 295:606 - 609, 2002.

[13] P. Devreotes and C. Janetopoulos, "Eukaryotic Chemotaxis: Distinctions between Directional Sensing and Polarization", *J. BIOLOGICAL CHEMISTRY*, 278:20445-20448, 2003.

[14] J. Du, B. Fix, J. Glimm, X. Jia, X. Li, Y. Li and L. Wu, "A Simple Package for Front Tracking", *J. comp. phys.*, 213:613-628, 2006.

[15] E. Dussan, "On the Spreading of Liquids on Solid Surfaces: Static and Dynamic Contact Line", *Abbu. Rev. Fluid Mech.*, 11:371-400, 1979.

[16] J. Eggers and H. Stone, "Characteristic Lengths at Moving Contact Lines for a Perfectly Wetting Fluid: the Influence of Speed on the Dynamic Contact Angle", *J. Fluid Mechanics*, 505:309-321, 2004.

[17] M. Fermigier and P. Jenffer, "An Experimental Investigation of the Dynamic Contact Angle in Liquid-Liquid Systems", *J. Colloid and Interface Science*, 146:226-241, 1990.

[18] R. Finn, "*Equilibrium Capillary Surfaces*", Springer-Verlag press, 1986.

[19] E. Foxman and J. Campbell and E. Butcher, "Multistep Navigation and the Combinatorial Control of Leukocyte Chemotaxis", *Journal of Cell Biology*, 139:1349-1360, 1997.

[20] D. Fu, "*Numerical Simulation in Fluid Dynamics*", National Defense Industry press, 1993.

[21] J. Glimm, J. Grove, X. Li, K. Shyue, Q. Zhang and Y. Zeng, "Three Dimensional Front Tracking", *SIAM J. Sci. Comp.*, 19:703–727, 1998.

[22] J. Glimm, J. Grove, X. Li and D. Tan, "Robust Computational Algorithms for Dynamic Interface Tracking in Three Dimensions", *SIAM J. Sci. Comp.*, 21:2240–2256, 2000.

[23] M. Herant, W. Marganski and M. Dembo, "The Mechanics of Neutrophils: Synthetic Modeling of Three Experiments", *Biophysical Journal*, 84:3389-3413, 2003.

[24] L. Hocking, "A Moving Fluid Interface on a Rough Surface", *J. Fluid Mechanics Digital Archive*, 76:801-817, 1976.

[25] L. Hocking, "On Contact Angles in Evaporating Liquids", *J. Phys.Fluids*, 7:2950-2955, 1995.

[26] A. Horwitz and J. Parsons, "Cell Biology: Cell Migration Moving On", *Science*, 286:1102-1103, 1999.

[27] R. Kamm, "Cellular Fluid Mechanics", *Annu. Rev. Fluid Mech.* 34:211-232, 2002.

[28] D. Lauffenburger and A. Horwitz, "", Cell, 84:359-369, 1996.

[29] T. LEWINER, H. LOPES, A. VIEIRA and G. TAVARES, "Efficient Implementation of Marching Cubes Cases with Topological Guarantees", *J. Graphics Tools*, 8:1-15, 2003.

[30] L. Li and J. Glimm and X. Li, "All Isomorphic Distinct Cases for Multi-Component Interfaces in a Block", *J. Comp. Appl. Mathematics*, 152:263-276, 2003.

[31] W. Lorensen and H. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *Computer Graphics*, 21:163-169, 1987.

[32] C. Janetopoulos, L. Ma, P. Devreotes and P. Iglesias, "Chemoattractant-Induced Phosphatidylinositol 3,4,5-Trisphosphate Accumulation is Spatially Amplified and Adapts, Independent of the Actin Cytoskeleton", *Proc Natl Acad Sci.*, 101:8951-8956, 2004.

[33] J. Malkevitch, "*Cubes*", American MathematicalSociety, 2003.

[34] I. Maly, H. Wiley and D. Lauffenburger, "Self-Organization of Polarized Cell Signaling via Autocrine Circuits: Computational Model Analysis", *Biophysical Journal*, 86:10-22, 2004.

[35] A. Mogilner and L. Edelstein-Keshet, "REgulation of Actin Dynamics in Rapidly Moving Cells: A Quantitative Analysis", *Biophysical Journal*, 83:1237-1258, 2002.

[36] A. Mogilner and T. Elston and H. Wang and G. Oster, "*Computational Cell Biology*" Springer Press, 2002.

[37] A. Mogilner and G. Oster, "Force Generation by Actin Polymerization II: The Elastic Ratchet and Tethered Filaments", *Biophysical Journal*, 84:1591-1605, 2003.

[38] A. Mogilner and G. Oster, "Shrinking Gels Pull Cells", *SCIENCE*, 302:1340-1341, 2003.

[39] A. Mogilner and G. Oster, "Polymer Motors: Pushing out the Front and Pulling up the Back", *Curr. Biol.* 13:R721-R733, 2003.

[40] A. Mogilner and G. Oster, "Cell Motility Driven by Actin Polymerization", *Biophysical Journal*, 71:3030-3045, 1996.

[41] A. Mogilner and R. Wollman, "Quantitative Modeling in Cell Biology: What Is It Good for?", *Developmental Cell*, 11:279-287, 2006.

[42] G. Moretti and B. Grossman and F. Marconi, "*A Complete Numerical Technique for the Calculation of Three Dimensional Inviscid Supersonic Flow*", *AIAA*, 72-192, 1972.

[43] A. Neumman and J. Spelt, "*Applied Surface Thermodynamics*", Marcel Dekker press, 1996.

[44] I. Novak, B. Slepchenko, A. Mogilner and L. Loew, "Cooperativity between Cell Contractility and Adhension", *Physical Review Letters*, 93:268109-1-4, 2004.

[45] C. Parent and P. Devreotes, "A Cell's Sense of Direction", *Science Review*, 284:765-770, 1999.

[46] N. Patankar, "On the Modeling of Hydrophobic Contact Angles on Rough Surfaces", *Langmuir*, 19:1249 -1253, 2003.

[47] K. Painter and P. Maini and H. Othmer, "Development and Applications of a Model for Cellular Response to Multiple Chemotactic Cues", *J. Math. Biol.*, 41:285-314, 2000.

[48] T. Pollard and G. Borisy, "Cellular Motility Driven by Assembly and Disassembly of Actin Filaments", *Cell*, 112:453-465, 2003.

[49] M. Prass, K. Jacobson, A. Mogilner and M. Radmacher, "Direct Measurement of the Lamellipodial Protrusive Force in a Migrating Cell", J. Cell Biology, 174:767-772, 2006.

[50] D. Rajon and W. Bolch, "Marching Cube Algorithm: Review and Trilinear Interpolation Adaptation for Image-Based Dosimetric Models", *Comp. Med. Imaging and Graphuics*, 27:411-435, 2003.

[51] N. Ray and S. Acton, "Tracking Leukocytes In Vivo with Shape ans Size Constrained Active Contour", *IEEE Trans. on Medical Imaging*, 21:1222-1235, 2002.

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

[53] A. Ridley, M. Schwart, K. Burridge, R. Firtel, M. Ginsberg, G. Borisy, J. Parsons and A. Horwitz, "Cell Migration: Integrating Signals from Front to Back", *Sciences*, 302:1704-1709, 2003.

[54] L. Schwartz, R. Roy, R. Eley and H. Princen, "Surfactant-Driven Motion and Splitting of Droplets on a Substrate", J. Engr. Mathematics, 50:157175, 2004.

[55] A. Sklozowska, M. Wozniak and R. Matlakowska, "The Method of Contact Angle Measurements and Estimation of Work of Adhesion in Bioleaching of Metals", *Biol. Proc. Online*, 1(3), 114-121, 1999.

[56] J. Small, T. Stradal, E. Vignal and K. Rottner, "The Lamellipodium: Where Motility Begins", *Trends in Cell Biology* 12:112-120, 2002.

[57] J. Theriot, T. Mitchison, L. Tilney and D. Portnoy, "The rate of actin-based motility of Intracellular Listeria Monocytogenes Equals the Rate of Actin Polymerization", *Nature*, 375:257-260, 1992.

[58] T. Watanabe, J. Noritake and K. Kaibuchi, "Regulation of Microtubules in Cell Migrations", *Review: TRENDS in Cell Biology*, 15:76-83, 2005.

[59] M. Zaman, R. Kamm, P. Matsudaria and D. Lauffenburger, "Computational Model for Cell Migration in Three-Dimensional Matrices", *Biophysics Journal*, 89:1389-1397, 2005.