國立交通大學

應用數學系

碩 士 論 文

等位集與流體體積混合應用在二相流

Coupled Level Set and Volume-of-Fluid Method
for Two-Phase Flows

研 究 生：蔡修齊

指導教授：賴明治　教授

中 華 民 國 九 十 七 年 六 月

等位集與流體體積混合應用在二相流

Coupled Level Set and Volume-of-FluidMethod
for Two-Phase Flows

研 究 生：蔡修齊　　Student：Siou-Chi Tsai

指導教授：賴明治　　Advisor：Ming-Chih Lai

國 立 交 通 大 學
應 用 數 學 系

碩 士 論 文

A thesis
submitted to Department of Applied Mathematics
College of Science
National Chiao Tung University
in partial fulfillment of the requirements
for the Degree of Master
in

Applied Mathematics

June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

等位集與流體體積混合

應用在二相流

學生：蔡修齊　　指導教授：賴明治

國立交通大學應用數學系（研究所）碩士班

摘　　要

本論文介紹如何用等位集方法來處理介面上的熱方程問題，其中等位集方法只需要用到直角座標系即可。接著我們結合等位集方法和流體體積法來模擬二相流，而且這個混合的方法可以用來解關於介面的性質。最後我們加入不可溶解的介面活性劑在我們的介面上並解二相流問題，然後再觀察界面活性劑對於介面的影響。

Coupled Level Set and Volume-of-Fluid Method for Two-Phase Flows

Student : Siou-Chi Tsai Advisor:Ming-Chih Lai

Department (Institute) of Applied Mathematics

National Chiao Tung University

Abstract

In this paper we introduce level set method to solve heat equation on interface with Cartesian coordinate. Then we couple level set method and Volume-of-Fluid method to simulate two-phase flow for interface property and conserve the volume of inner area. Finally we add insoluble surfactant on the interface when simulating two-phase flows and observe the impact of surfactant on interface.

誌　　謝

本篇論文的完成，首先要感謝我的指導老師－賴明治教授。在老師的一再指點之下，讓我慢慢走入數值方法和科學計算的世界並且引導我學習與研究的方向，一步一步走入計算流體的專門領域。除了研究還有課業的指導，老師也告訴我做研究的態度還有方法，讓我很快的進入狀況，並且讓我在數值科學計算的領域中，找到我的研究興趣和研究的成就，最後還要感謝老師教我做人處世的道理以及如何面對人生的困境，這讓我覺得受益無窮。除了指導老師外，科學計算實驗室的夥伴們也給我不少幫助。首先我要感謝曾昱豪學長指點我關於流體和 Navier-Stokes 方程的理論和數值計算，並且提供解 Navier-Stokes 方程的程式讓我應用於二相流的模擬，讓我在研究的過程中能夠持續獲得進展。我也要感謝曾孝捷學長和陳冠羽學長指點我程式和課業，時時提點我該注意的事項，讓我在程式編寫時還是課業有問題時能有適時的引導。最後還要感謝其他碩二一起奮鬥的同學，在這段研究期間裡一起努力也一起歡笑，讓生活充實而且不孤單。

在論文口試期間，承蒙王偉仲教授、洪子倫教授與吳金典助理教授費心審閱並提供許多寶貴意見，使本論文得以更加完備，學生永銘在心。

最後，我要感謝我的父母還有我的哥哥，我的父母總是會鼓勵我在研究上能多花費心思，也告訴我做研究或做事情必須要嚴謹要小心，也要感謝我的哥哥，他提供我在研究所路程上的經驗，讓我能夠早點上軌道而且能夠早點適應環境，沒有他們的細心照顧和鼓勵，我沒有辦法如期完成這篇論文。最後，我想把這篇論文的喜悅和成果分享給我的家人以感謝他們對我的細心照顧和不斷的激勵。

目　　錄

1 Introduction

In this article, we introduce two methods which are *Level Set Methed* and *Volume-of-Fluid Method* to simulate two-phase flows. The Level set method is developed mainly by S. Osher[1] and J. A. Sethian[2] and the application of level set method is very wide like solving Stefan problem[9] or solving PDEs on surface[12, 15, 16, 18, 19]. In our article we also introduce how to solve heat equation on circle by Local Level Set Method.

To simulate two-phase flows problem, [6, 7] use level set method to simulate the interface motion under two-phase flows. If we just use level set to simulate, then the inner area of interface will not be conserved. The reason that inner area does not conserve is the scheme of level set method do not have this property. Although using level set to simulate the interface under two-phase flows is very easy and the method can be adapted to topology changing of interface, the method also has the critical drawback which is volume does not conserve.

To overcome the conservation of inner area, Hirt and Nichols[3] developed *Volume-of-Fluid Method* to handle the conservation of inner area. This method is mainly to handle the volume conservation issue and the method is also adapted to topology changing of interface as level set method.But the drawback of VOF method is how to describe the interface.If we just use VOF method, we will face on an issue which is how to reconstruct interface only using VOF method[10, 11] which is not as easy as level set method to describe interface.

So, [14, 17] coupled VOF and level set method to simulate two-phase flows for axisymmetric case and two-dimensional case respectively. In our article we mainly refer the method of [17] to simulate for the two-dimensional case.

The insoluble surfactant on the interface in two-phase flow is also an interesting topic and [21] use IIM and local level set method to handle two-phase flows with surfactant. We refer the local level set method of [21] and the coupled VOF and level set method to handle the two-phase flows with surfactant.

In this article we first introduce solving PDE on circle by level set method which relates the reinitialization of level function by ENO or WENO scheme[4, 5, 13] and then we use coupled VOF and level set method to simulate two-phase flows without surfactant for two-dimensional case. Finally we couple above two topics to simulate two-phase flow with surfactant.

2 Navier-Stokes Equations

2.1 Introduction

In order to simulate two-phase flows with the interface we need the model which simulates the motion of two-phase flows. This model is Navier-Stokes equations

$$\begin{aligned} \rho(\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u}) &= -\nabla p + \rho g - \sigma\kappa\mathbf{n} + \nabla\cdot\mu[\nabla\mathbf{u} + (\nabla\mathbf{u})^T] \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned} \tag{1}$$

where $\mathbf{u}$:velocity,ρ:density, p:pressure, μ:viscosity, g:gravity and $\sigma\kappa\mathbf{n}$ denotes the surface tension of the interface where σ:surface tension coefficient, κ:curvature on interface and $\mathbf{n}$:normal on interface.

2.2 Numerical Scheme

We solve Navier-Stokes equation by two steps as follows.

Step1 Prediction:
We solve first step by Crank-Nicholson method in time

$$\begin{aligned} (\nabla\cdot(\mu^{n+1}\nabla\mathbf{u}^*) - \frac{2\rho^{n+\frac{1}{2}}}{\Delta t}\mathbf{u}^*) =& 2\nabla p^{n-\frac{1}{2}} - \frac{2\rho^{n+\frac{1}{2}}}{\Delta t}\mathbf{u}^n - \nabla\cdot(\mu\nabla\mathbf{u})^n \\ &+ (3\mathbf{F}^n - \mathbf{F}^{n-1}) \end{aligned} \tag{2}$$

where

$$\mathbf{F} = \begin{bmatrix} \rho(uu_x + vu_y) - (\mu v_x)_y - (\mu u_x)_x - f_1 \\ \rho(uv_x + vv_y) - (\mu v_y)_x - (\mu u_y)_y - f_2 \end{bmatrix}$$

and $\mathbf{f} = (f_1, f_2)$ means the external force term $(\rho g - \sigma\kappa\mathbf{n})$

Note that μ^{n+1} and $\rho^{n+\frac{1}{2}}$ are computed by extrapolation as follows

$$\begin{aligned} \mu^{n+1} &= 2\mu^n - \mu^{n-1} \\ \rho^{n+\frac{1}{2}} &= \frac{3}{2}\rho^n - \frac{1}{2}\rho^{n-1} \end{aligned}$$

and Eq.(2) becomes Poisson type equation and can be solved by linear system for u^*.

Step2 Projection:

Since the new $\mathbf{u}^*$ is not diverge free, we apply this step to correct $\mathbf{u}^*$ to be diverge free which means $\nabla \cdot \mathbf{u} = 0$ and then compute new pressure term $\nabla p^{n+\frac{1}{2}}$.By Helmholtz-Hodge decomposition, we have

$$\begin{aligned} \mathbf{u}^* &= \mathbf{u}^{n+1} + \Delta t \nabla \psi^{n+1} \\ \nabla \cdot \mathbf{u}^{n+1} &= 0 \end{aligned}$$

So we have

$$\rho^{n+\frac{1}{2}} \frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\nabla \psi^{n+1}$$

and then take divergence we have

$$\nabla \cdot \left(\frac{\nabla \psi^{n+1}}{\rho^{n+\frac{1}{2}}}\right) = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}$$

After getting $\nabla \psi^{n+1}$ we have

$$\begin{aligned} \mathbf{u}^{n+1} &= \mathbf{u}^* - \frac{\Delta t}{\rho^{n+\frac{1}{2}}} \nabla \psi^{n+1} \\ \nabla p^{n+\frac{1}{2}} &= \nabla p^{n-\frac{1}{2}} + \nabla \psi + \nabla \cdot (\mu^{n+1}(\mathbf{u}^{n+1} - \mathbf{u}^*)) \end{aligned}$$

Acknowledgment

The whole scheme of solving Navier-Stokes equation as above is provided by Yu-Hau Tseng and again we thank Tseng for providing the code of solving Navier-Stokes equations.

3 Solving PDE on Interface by Level Set Method

3.1 Introduction

In the numerical simulation of partial differential equations, we usually encounter the problem of handling interface, for example:two-phase flows of Navier-Stokes equation or heat equation on the interface which is irregular or high dimensional. To describe the interface is the main issue when solving PDE with interface problem. We use level set function which we will have details later. There are many advantages that level set method provides, for example the interface can be easy presented and when normal vector on the interface are needed, it also can be calculated easily by level set method. Our goal in this chapter is to solve heat equation on the circle which means that the heat quantities just diffuse on the circle and do not diffuse out of circle.As we just use Cartesian grid computational domain, how do we handle this problem? We will give details step by step.

3.2 Level Set Function and Signed Distance Function

3.2.1 Level Set Function

The concept of level set function is very simple. We let the symbol ϕ to represent level set function.On the interface, ϕ is defined to be zero which is zero level set, outside the interface we define $\phi > 0$ and inside the interface we let $\phi < 0$. With the help of level set function, we can easily distinguish points which are inside or outside the interface.(See figure1)

3.2.2 Signed Distance Function

Next we will explain what the signed distance function is. Let a point $P(x_0, y_0)$ which is arbitrary and we have level function ϕ .If the value of $\phi(P)$ is equal to the distance from P to the interface,we call that ϕ is signed distance function. Maybe this description is not clear enough, let us take a example

Example.1 We take a level function of circle, (See figure1)

$$\phi(x, y) = \sqrt{x^2 + y^2} - 1$$

with center $O(0, 0)$ and radius $r = 1$ and we claim that ϕ is signed distance function. Note that O is inside interface and distance from O to interface is 1, then we find that $\phi(0, 0) = -1$ where $-$ represent inside interface

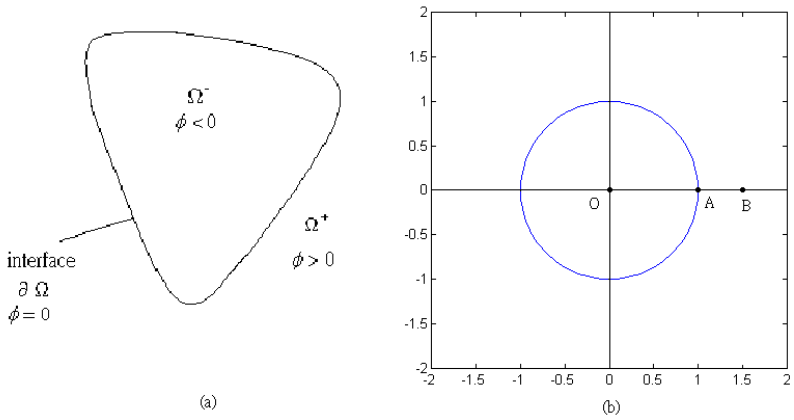

Figure 1: (a)Definition of level function ϕ (b)Figure of example.1

and 1 denote the distance. $A(1,0)$ is on the interface and we find that $\phi(1,0) = 0$ and $B(1.5,0)$ is outside interface with distance 0.5 to interface, so $\phi(1.5,0) = 0.5$.Since ϕ is signed distance function, we have such pretty property.

From the above example we can figure out the meaning of signed distance function. So for any point plugging into ϕ, we can know the point is outside,inside or on the interface and the distance from point to interface is also known when ϕ is signed distance function.

3.2.3 How To Distinguish Signed Distance Function?

In few cases we can just write down the signed distance function as mathematical formulation exactly like example.1. Since circle have very good properties, so the above example of circle is special case. Note that the level function of unit circle is not only one case. Consider that $\phi(x,y) = x^2+y^2-1$ is also a level function for unit circle, but unfortunately,it is no longer a signed distance function. Let us take a test,plug B into ϕ we get $\phi(1.5,0) = 1.25$ is not equal to 0.5, so this level function is not signed distance function.

Actually, signed distance function has a very good property which is $|\nabla\phi| = 1$ and it is also a condition to check whether a level function is a signed distance function or not. For computing normal $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$, if ϕ is signed distance function, we can calculate $\mathbf{n}$ more accurately and $\mathbf{n} = \nabla\phi$ where $\nabla\phi$ can be calculated by central difference and get second order for $\mathbf{n}$. If ϕ is not signed distance function then $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ will be first order accuracy and may not accurate enough, this is why using signed distance function to

calculate **n** will be better.

Most of level set functions are not signed distance function. For example, the level function of ellipse:$\phi(x,y) = \sqrt{(\frac{x}{a})^2 + (\frac{y}{b})^2} - 1$ is not a signed distance function.It is easy to check that $|\nabla\phi| \neq 1$. In order to let readers understand more about signed distance function and distinguish whether level function is signed distance function or not, we contour three level functions in the following :

$$\phi_1(x,y) = \sqrt{x^2+y^2} - 1 \tag{3}$$

$$\phi_2(x,y) = (x^2+y^2) - 1 \tag{4}$$

$$\phi_3(x,y) = \sqrt{(\frac{x}{0.6})^2 + (\frac{y}{0.3})^2} - 1 \tag{5}$$

for ϕ-values $[-20\Delta x : 5\Delta x : 20\Delta x]$ as in figure2.

Note that ϕ_1 and ϕ_2 are both level function of unit circle and ϕ_3 is level function for ellipse. In Eq.(1),the distance between any two adjacent level curves are the same and equal to $5\Delta x$, that is why we call Eq.(1) is signed distance function. Eq.(2) and Eq.(3) do not have such nice property and the distance between any two adjacent level curves are not the same. So when calculating normal on interface, signed distance function will be more accurate. Now we have introduced another method to distinguish signed distance function which is to observe the contour of ϕ.

Remark
The fifth level curve of each ϕ in figure2 is the interface, that is $\phi = 0$.We will find that the location of $\phi = 0$ of Eq.(1) and Eq.(2) are at the same location since they are both level functions for unit circle and $\phi = 0$ denotes the interface.

Since most level functions are not signed distance function and the exact mathematical form of signed distance function is hard to find, so how do we handle this problem? In the next section, we will introduce the Re-initialization process which can overcome the problem.

3.3 Re-initialization

In this section, we introduce how to modify a level function which is not a signed distance function into a signed distance function. The process is not complicated and it involves solving a PDE to steady state as follow:

$$\phi_t + S(\phi_0)(|\nabla\phi| - 1) = 0 \tag{6}$$

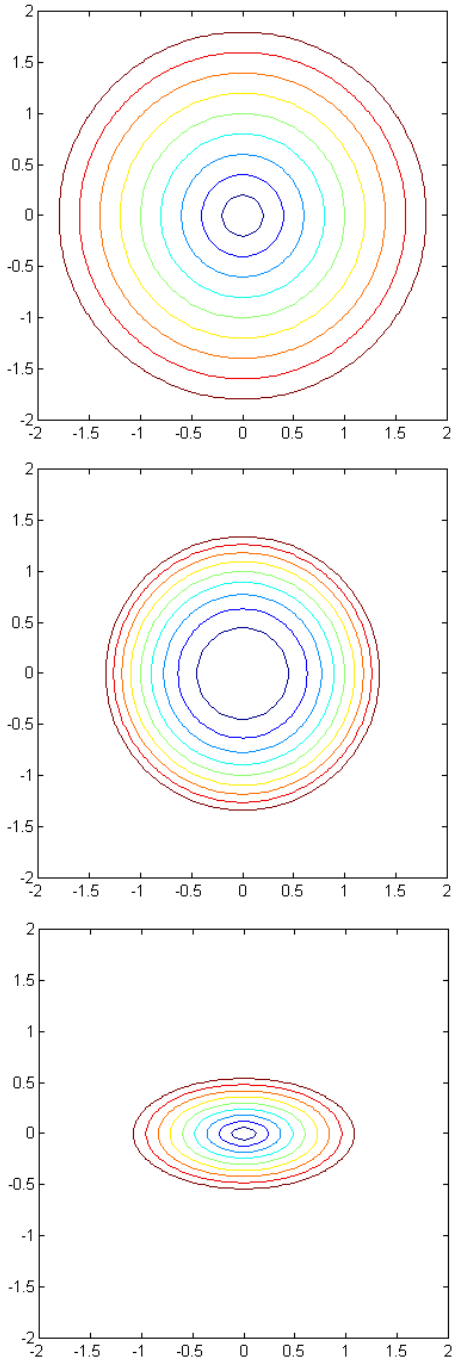

Figure 2: Contour of Eq.(3),(4) and (5)

where

$$S(\phi_0) = \begin{cases} 1 & \text{in } \Omega^+ \\ -1 & \text{in } \Omega^- \\ 0 & \text{on } \partial\Omega \end{cases}$$

Let us explain the meaning of Eq.(6) first. Consider a PDE which similar to Eq.(6) :

$$\phi_t + |\nabla\phi| = f(\mathbf{x}) \tag{7}$$

If we solve Eq.(7) to steady state, we will get that $|\nabla\phi| = f(\mathbf{x})$. Since signed distance function has a property $|\nabla\phi| = 1$, so if the level function ϕ is not signed distance function(i.e. $|\nabla\phi| \neq 1$), we use the concept of Eq.(7) to enforce $|\nabla\phi| = 1$, that is to solve :

$$\phi_t + |\nabla\phi| = 1 \tag{8}$$

From Eq.(7) we know that the information is propagated in the normal direction. So the information is carried from small ϕ-value to large ϕ-value. So we hope the information propagated from interface(i.e. $\phi = 0$). But if we solve Eq.(8) for whole domain Ω, since the level function is negative inside interface,so the information is propagated from the Ω^- and obviously it is wrong.

So if we solve Eq.(8) in $\Omega^+ \cup \partial\Omega$ then it means that the information is propagated form interface to Ω^+ and it make sense. Similar concept, we want the information propagated from interface to Ω^-, so we solve the equation :

$$\phi_t - |\nabla\phi| = -1 \tag{9}$$

in the domain $\Omega^- \cup \partial\Omega$.

Finally we couple Eq.(8) and Eq.(9) to get Eq.(6). Solving Eq.(6) to steady state, we have $|\nabla\phi| - 1 = 0$(i.e. $|\nabla\phi| = 1$) and that is the goal of reinitialization. Note that if $\phi = 0$ denotes the interface, then after reinitialization process,$\phi = 0$ is static and do not change it's value since $S(\phi_0) = 0$ on $\partial\Omega$. So the reinitialization process ensure that the location of interface will not be changed.

3.4 Spatial Discretization for Reinitialization Process

In this section, we will describe how to handle $|\nabla\phi|$ in Eq.(6). If we just use central difference to compute $|\nabla\phi|$, then we will get terrible result.Before handling this term, let us look at $1D$ example :

Example.2 (Burger's equation)

$$u_t + uu_x = 0 \tag{10}$$

We discretize u_x in Eq.(10) by upwind difference.We denote :

$$D_i^+ u = \frac{u_{i+1} - u_i}{\Delta x} \text{ and } D_i^- u = \frac{u_i - u_{i-1}}{\Delta x}$$

Note that the coefficient u of u_x decides the information from which direction. That is, if $u > 0$ we know that the characteristics come from left side so we use $D_i^- u$ to compute $(u_i)_x$. On the other hand, if $u < 0$,it means that the information and characteristics come from right side so we use $D_i^+ u$ to compute $(u_i)_x$.

The above equations explain what is upwind difference for $1D$ case.Note that Eq.(6) and Eq.(10) are similar for some sense,but Eq.(6) is a $2D$ PDE and is not as simple as Eq.(10) when handling first derivative term.

3.4.1 Godunov's Method

Since Eq.(6) is a $2D$ PDE which is not as simple as Eq.(10), we introduce Godunov's Method to handle $|\nabla\phi|$ in Eq.(6). Actually, Godunov's method uses the same concept which is characteristic coming from which direction. But in $2D$ domain you may doubt that the characteristic can come from infinite directions and how to decide? To overcome this problem, we just simplify and separate x-direction and y-direction to compute respectively that is just compute ϕ_x and ϕ_y respectively. Since we know the information propagated from interface, so when computing $(\phi_{i,j})_x$, we choose the point of $\phi_{i+1,j}$ and $\phi_{i-1,j}$ by the direction of information propagating and then compute $(\phi_{i,j})_x$. For example,if (i,j) is outside interface and if $\phi_{i-1,j} < \phi_{i,j} < \phi_{i+1,j}$ which means $(i-1,j)$ is nearest interface within these three points and the information is propagated from $\phi_{i-1,j}$ to $\phi_{i+1,j}$, so we use $D_{i,j}^-\phi = \frac{(\phi_{i,j}-\phi_{i-1,j})}{\Delta x}$. On the other hand, if $\phi_{i-1,j} > \phi_{i,j} > \phi_{i+1,j}$ which means information comes from $\phi_{i+1,j}$ to $\phi_{i-1,j}$, then we use $D_{i,j}^+\phi = \frac{(\phi_{i+1,j}-\phi_{i,j})}{\Delta x}$.

Having the basic concept of Godunov's mothod, we find that in some sense this method is as upwind method. Here we describe Godunov's method for solving Eq.(6) in the following :

Details of Godunov's Method

1. If $\phi_{i,j} > 0$ which means the point is outside interface, then

(a) If $D^+_{i,j}\phi > 0$ and $D^-_{i,j}\phi > 0$ then it means $\phi_{i-1,j} < \phi_{i,j} < \phi_{i+1,j}$ and $(\phi_{i,j})_x = D^-_{i,j}\phi$

(b) If $D^+_{i,j}\phi < 0$ and $D^-_{i,j}\phi < 0$ then it means $\phi_{i-1,j} > \phi_{i,j} > \phi_{i+1,j}$ and $(\phi_{i,j})_x = D^+_{i,j}\phi$

(c) If $D^+_{i,j}\phi \leq 0$ and $D^-_{i,j}\phi \geq 0$ then it means $\phi_{i-1,j} \leq \phi_{i,j}$ and $\phi_{i,j} \geq \phi_{i+1,j}$. In this case, the information comes form both two sides and we treat the shock by the larger influence between $D^-_{i,j}$ and $D^+_{i,j}$, that is $(\phi_{i,j})_x = \max(|D^-_{i,j}|, |D^+_{i,j}|)$

(d) If $D^+_{i,j}\phi \geq 0$ and $D^-_{i,j}\phi \leq 0$ then it means $\phi_{i-1,j} \geq \phi_{i,j}$ and $\phi_{i,j} \leq \phi_{i+1,j}$ and the information flow to both sides from (i,j). This does not make sense since interface do not pass grid (i,j). So in this case we just set $(\phi_{i,j})_x = 0$.

2. If $\phi_{i,j} < 0$ which means the point is inside interface.
(Note that inside interface the ϕ-value is negative and if ϕ is larger, it is much near the interface.)

(a) If $D^+_{i,j}\phi > 0$ and $D^-_{i,j}\phi > 0$ then it means $\phi_{i-1,j} < \phi_{i,j} < \phi_{i+1,j}$ and $(\phi_{i,j})_x = D^+_{i,j}\phi$.

(b) If $D^+_{i,j}\phi < 0$ and $D^-_{i,j}\phi < 0$ then it means $\phi_{i-1,j} > \phi_{i,j} > \phi_{i+1,j}$ and $(\phi_{i,j})_x = D^-_{i,j}\phi$

(c) If $D^+_{i,j}\phi \geq 0$ and $D^-_{i,j}\phi \leq 0$ then it means $\phi_{i-1,j} \geq \phi_{i,j}$ and $\phi_{i,j} \leq \phi_{i+1,j}$. In this case, the information comes form both two sides and we treat the shock by the larger influence between $D^-_{i,j}$ and $D^+_{i,j}$, that is $(\phi_{i,j})_x = \max(|D^-_{i,j}|, |D^+_{i,j}|)$

(d) If $D^+_{i,j}\phi \leq 0$ and $D^-_{i,j}\phi \geq 0$ then it means $\phi_{i-1,j} \leq \phi_{i,j}$ and $\phi_{i,j} \geq \phi_{i+1,j}$ and the information flow to both sides from (i,j). This does not make sense since interface do not pass grid (i,j). So in this case we just set $(\phi_{i,j})_x = 0$.

After simplifying the Godunov's method as above, the scheme is obvious and easy to implement by numerical computing. Here we give three remarks as follows:

Remark

1. In Godunov's method, we discetize ϕ_x by one-sided difference, so the scheme is first order accurate.

2. The cases $\phi > 0$ and $\phi < 0$ are opposite since the signed function $S(\phi_0)$ is 1 outside interface and -1 inside interface.

3. The y-direction case ϕ_y is as same as computing ϕ_x. So after solving ϕ_x and ϕ_y, the spatial discretization of $|\nabla\phi|$ in Eq.(6) is complete.

3.4.2 Second-Order Hamilton-Jacobi ENO Scheme

Since the Godunov's method is first order, if we want the scheme more precise, we introduce second-order ENO (essential non-oscillatory) scheme to discretize. The ENO scheme is based on Godunov's method for first-order term which described above when solving Eq.(6). We write the Taylor expansion of ϕ:

$$\phi(x) = \phi(\tilde{x}) + \phi'(\tilde{x})(x - \tilde{x}) + \frac{\phi''(\tilde{x})}{2!}(x - \tilde{x})^2 + \cdots \tag{11}$$

If we use $D^-\phi_i$, then it means we plug x_i and x_{i-1} into Eq.(11) and get $(\phi_i)_x$. Now the ENO scheme is to add a adjacent point plug into Eq.(11). The point is chosen to be much smooth between two sides. Let us take a example to explain :

Example.3 If Godunov's method tell us that we choose the point x_{i-1} when computing $(\phi_i)_x$ then we need to choose another point between x_{i+1} and x_{i-2} to plug into Eq.(11) and get second-order result. We want choose much smooth side point so we need some criterion for choosing point. Let us denote:

$$\begin{aligned}
&\Delta^+\phi(x_i) = \phi(x_{i+1}) - \phi(x_i)\\
&\Delta^-\phi(x_i) = \phi(x_i) - \phi(x_{i-1})\\
&\Delta^-\Delta^+\phi(x_i) = \phi(x_{i+1}) - 2\phi(x_i) + \phi(x_{i-1})\\
&\Delta^-\Delta^+\phi(x_{i-1}) = \phi(x_i) - 2\phi(x_{i-1}) + \phi(x_{i-2})
\end{aligned}$$

Then if $|\Delta^-\Delta^+\phi(x_{i-1})| \leqslant |\Delta^-\Delta^+\phi(x_i)|$ which means left side is much smooth and we choose the point x_{i-2} to plug into Eq.(11). Otherwise we choose x_{i+1} to plug into Eq.(11). No matter what point is chosen, the scheme is always second-order.

Now we describe the whole process for $1D$ case for ENO scheme based on Godunov's method :

1. If we use $D^-\phi$ for first-order discretization to compute $(\phi_i)_x$

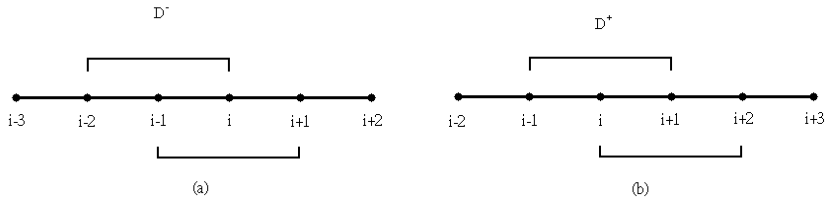

Figure 3: (a).left-biased (b).right-biased

(a) If $|\Delta^-\Delta^+\phi(x_{i-1})| \leqslant |\Delta^-\Delta^+\phi(x_i)|$,then

$$(\phi_i)_x = \frac{\phi(x_i) - \phi(x_{i-1})}{\Delta x} + \frac{\phi(x_i) - 2\phi(x_{i-1}) + \phi(x_{i-2})}{2\Delta x} \tag{12}$$

$$= \frac{3\phi(x_i) - 4\phi(x_{i-1}) + \phi(x_{i-2})}{2\Delta x} \tag{13}$$

(b) If $|\Delta^-\Delta^+\phi(x_{i-1})| > |\Delta^-\Delta^+\phi(x_i)|$,then

$$(\phi_i)_x = \frac{\phi(x_i) - \phi(x_{i-1})}{\Delta x} + \frac{\phi(x_{i+1}) - 2\phi(x_i) + \phi(x_{i-1})}{2\Delta x} \tag{14}$$

$$= \frac{\phi(x_{i+1}) - \phi(x_{i-1})}{2\Delta x} \tag{15}$$

2. If we use $D^+\phi$ for first-order discretization to compute $(\phi_i)_x$

(a) If $|\Delta^-\Delta^+\phi(x_i)| \leqslant |\Delta^-\Delta^+\phi(x_{i+1})|$,then

$$(\phi_i)_x = \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x} + \frac{\phi(x_{i+1}) - 2\phi(x_i) + \phi(x_{i-1})}{2\Delta x} \tag{16}$$

$$= \frac{3\phi(x_{i+1}) - 4\phi(x_i) + \phi(x_{i-1})}{2\Delta x} \tag{17}$$

(b) If $|\Delta^-\Delta^+\phi(x_i)| > |\Delta^-\Delta^+\phi(x_{i+1})|$,then

$$(\phi_i)_x = \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x} + \frac{\phi(x_{i+2}) - 2\phi(x_{i+1}) + \phi(x_i)}{2\Delta x} \tag{18}$$

$$= \frac{\phi(x_{i+2}) - \phi(x_i)}{2\Delta x} \tag{19}$$

To handle $2D$ case, it is as the same method and we split the domain into x-direction and y-direction to handle respectively. Here we have introduced the second-order ENO scheme for solving Hamilton-Jacobi equation like Eq.(6). Next we will introduce more higher order scheme.

3.4.3 Third-Order ENO Scheme

The third-order ENO scheme for spatial is based on second-order ENO scheme which is based on Godunov's method, so we realize that the concept of Godunov method is very important when solving Eq.(6). In second-order ENO, we use three points to calculate ϕ_x and get second-order result, the third-order is as the same concept as ENO-2. While deciding which three points are needed, we still demand one point to reach third order ENO(ENO-3) which is either left side or right side adjacent point. The fourth point is also chosen in smoother side and the criterion is similar to above ENO-2. Since the concept is similar, we do not take example again and we just give the details of algorithm as below:

1. If we use $D^-\phi$ for first-order discretization to compute $(\phi_i)_x$
   (a) If $|\Delta^-\Delta^+\phi(x_{i-1})| \leqslant |\Delta^-\Delta^+\phi(x_i)|$,then
      i. If $|\Delta^-\Delta^-\Delta^+\phi_{i-1}| < |\Delta^+\Delta^-\Delta^+\phi_{i-1}|$

$$(\phi_i)_x = \frac{\phi(x_i)-\phi(x_{i-1})}{\Delta x} + \frac{\phi(x_i)-2\phi(x_{i-1})+\phi(x_{i-2})}{2\Delta x} + \frac{\phi(x_i)-3\phi(x_{i-1})+3\phi(x_{i-2})-\phi(x_{i-3})}{3\Delta x}$$

      ii. If $|\Delta^-\Delta^-\Delta^+\phi_{i-1}| \geq |\Delta^+\Delta^-\Delta^+\phi_{i-1}|$

$$(\phi_i)_x = \frac{\phi(x_i)-\phi(x_{i-1})}{\Delta x} + \frac{\phi(x_i)-2\phi(x_{i-1})+\phi(x_{i-2})}{2\Delta x} + \frac{\phi(x_{i+1})-3\phi(x_i)+3\phi(x_{i-1})-\phi(x_{i-2})}{3\Delta x}$$

   (b) If $|\Delta^-\Delta^+\phi(x_{i-1})| > |\Delta^-\Delta^+\phi(x_i)|$,then
      i. If $|\Delta^-\Delta^-\Delta^+\phi_i| < |\Delta^+\Delta^-\Delta^+\phi_i|$

$$(\phi_i)_x = \frac{\phi(x_i)-\phi(x_{i-1})}{\Delta x} + \frac{\phi(x_i)-2\phi(x_{i-1})+\phi(x_{i-2})}{2\Delta x} - \frac{\phi(x_{i+1})-3\phi(x_i)+3\phi(x_{i-1})-\phi(x_{i-2})}{6\Delta x}$$

      ii. If $|\Delta^-\Delta^-\Delta^+\phi_i| \geq |\Delta^+\Delta^-\Delta^+\phi_i|$

$$(\phi_i)_x = \frac{\phi(x_i)-\phi(x_{i-1})}{\Delta x} + \frac{\phi(x_i)-2\phi(x_{i-1})+\phi(x_{i-2})}{2\Delta x} - \frac{\phi(x_{i+2})-3\phi(x_{i+1})+3\phi(x_i)-\phi(x_{i-1})}{6\Delta x}$$

2. If we use $D^+\phi$ for first-order discretization to compute $(\phi_i)_x$

   (a) If $|\Delta^-\Delta^+\phi(x_i)| \leqslant |\Delta^-\Delta^+\phi(x_{i+1})|$,then

      i. If $|\Delta^-\Delta^-\Delta^+\phi_i| < |\Delta^+\Delta^-\Delta^+\phi_i|$

$$(\phi_i)_x = \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x} + \frac{\phi(x_{i+1}) - 2\phi(x_i) + \phi(x_{i-1})}{2\Delta x} - \frac{\phi(x_{i+1}) - 3\phi(x_i) + 3\phi(x_{i-1}) - \phi(x_{i-2})}{6\Delta x}$$

      ii. If $|\Delta^-\Delta^-\Delta^+\phi_i| \geq |\Delta^+\Delta^-\Delta^+\phi_i|$

$$(\phi_i)_x = \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x} + \frac{\phi(x_{i+1}) - 2\phi(x_i) + \phi(x_{i-1})}{2\Delta x} - \frac{\phi(x_{i+2}) - 3\phi(x_{i+1}) + 3\phi(x_i) - \phi(x_{i-1})}{6\Delta x}$$

   (b) If $|\Delta^-\Delta^+\phi(x_i)| > |\Delta^-\Delta^+\phi(x_{i+1})|$,then

      i. If $|\Delta^-\Delta^-\Delta^+\phi_{i+1}| < |\Delta^+\Delta^-\Delta^+\phi_{i+1}|$

$$(\phi_i)_x = \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x} + \frac{\phi(x_{i+2}) - 2\phi(x_{i+1}) + \phi(x_i)}{2\Delta x} + \frac{\phi(x_{i+2}) - 3\phi(x_{i+1}) + 3\phi(x_i) - \phi(x_{i-1})}{3\Delta x}$$

      ii. If $|\Delta^-\Delta^-\Delta^+\phi_{i+1}| \geq |\Delta^+\Delta^-\Delta^+\phi_{i+1}|$

$$(\phi_i)_x = \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x} + \frac{\phi(x_{i+2}) - 2\phi(x_{i+1}) + \phi(x_i)}{2\Delta x} + \frac{\phi(x_{i+3}) - 3\phi(x_{i+2}) + 3\phi(x_{i+1}) - \phi(x_i)}{3\Delta x}$$

It seems that there are eight cases for ENO-3, but actually it can be reduced to just three cases. That is using $\{x_k|k = i-3, i-2, i-1, i\}$ or $\{x_k|k = i-2, i-1, i, i+1\}$ $\{x_k|k = i-1, i, i+1, i+2\}$ for D^- case to plug into Eq.(11). Similarly the right-biased stencil case for D^+ is $\{x_k|k = i, i+1, i+2, i+3\}$ or $\{x_k|k = i-1, i, i+1, i+2\}$ or $\{x_k|k = i-2, i-1, i, i+1\}$. Finally we get just three discretization forms.

3.4.4 Third-Order WENO

In this section we will discuss the third-order WENO(weighted essential non-oscillatory) scheme to discretize ϕ_x. Again WENO-3 is based on ENO-2.ENO-2 has two possibilities to compute ϕ_x under D^- (See Eq.(13) and

Eq.(15))and we combine this two by weights instead using only one to compute ϕ_x. That is why we call this method Weighted ENO.

The scheme of WENO-3 is more easy than ENO-3 since we only need to judge D^- or D^+. If we need D^- for first-order then we use left-biased points $\{x_k|k = i-2, i-1, i, i+1\}$ to compute ϕ_x. On the other case we use right-biased points $\{x_k|k = i-1, i, i+1, i+2\}$ to compute ϕ_x. The process of WENO is more simple since we do not need to judge which points are needed to add.

We write down the details of WENO-3 which we refer [12]:

1. If $D^-\phi$ is needed then

$$(\phi_x)_i = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} - \frac{w_-}{2}(\frac{\phi_{i+1} - 3\phi_i + 3\phi_{i-1} - \phi_{i-2}}{\Delta x})$$
$$w_- = \frac{1}{1+2r_-^2} \quad , r_- = \frac{\epsilon + (\phi_i - 2\phi_{i-1} + \phi_{i-2})^2}{\epsilon + (\phi_{i+1} - 2\phi_i + \phi_{i-1})^2}$$

2. If $D^+\phi$ is needed then

$$(\phi_x)_i = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} - \frac{w_+}{2}(\frac{\phi_{i+2} - 3\phi_{i+1} + 3\phi_i - \phi_{i-1}}{\Delta x})$$
$$w_+ = \frac{1}{1+2r_+^2} \quad , r_+ = \frac{\epsilon + (\phi_{i+2} - 2\phi_{i+1} + \phi_i)^2}{\epsilon + (\phi_{i+1} - 2\phi_i + \phi_{i-1})^2}$$

Note that the form is more simple and is easy to implement, so we recommend the WENO-3 scheme to discretize the ϕ_x and ϕ_y of reinitialization PDE.

Remark
Although the sign function of Eq.(6) is just simple 0,1 or -1, [12] suggested that

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla\phi|^2(\Delta x)^2}} \tag{20}$$

for a better result and reduce the time steps when solving Eq.(6) to steady state.

3.5 Temperal Discretization for Reinitialization Process

Since we prefer to use WENO-3 to discretize $|\nabla\phi|$ for spatial, if we want the scheme is higher order, we also need third-order method in time. Here we use TVD Runge-Kutta method which we refer [1].

3.5.1 Second-Order TVD Runge-Kutta Method

We let current time step be n and first we move time to $n+2$,then take average as below

$$\phi^{n+1} = \frac{1}{2}\phi^n + \frac{1}{2}\phi^{n+2}$$

3.5.2 Third-Order TVD Runge-Kutta Method

For third-order TVD-RK scheme we first need $\phi^{n+\frac{1}{2}}$ which is weighted by ϕ^n and ϕ^{n+2} and so that

$$\phi^{n+\frac{1}{2}} = \frac{3}{4}\phi^n + \frac{1}{4}\phi^{n+2}$$

Next we move $\phi^{n+\frac{1}{2}}$ to $\phi^{n+\frac{3}{2}}$ by iteration and finally we get ϕ^{n+1} as follows:

$$\phi^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\phi^{n+\frac{3}{2}}$$

Note that if ENO-2 is used for spatial, then TVD-RK2 have better been used for time and both ENO-3 and WENO-3 match TVD-RK3.

3.6 Numerical Test for Reinitialization Process

In this section, we will test some examples for reinitialization. We choose above example $\phi(x,y) = \sqrt{(\frac{x}{0.6})^2 + (\frac{y}{0.3})^2} - 1$ and a concave case which we refer [12] with polar coordinate $\phi(r,\theta) = (r - 0.5 + 0.1r\sin(7\theta))^3$.We contour $[-10\Delta x : 2\Delta x : 10\Delta x]$ for each case. Note that we use WENO-3 and TVD-RK3 method for these examples.

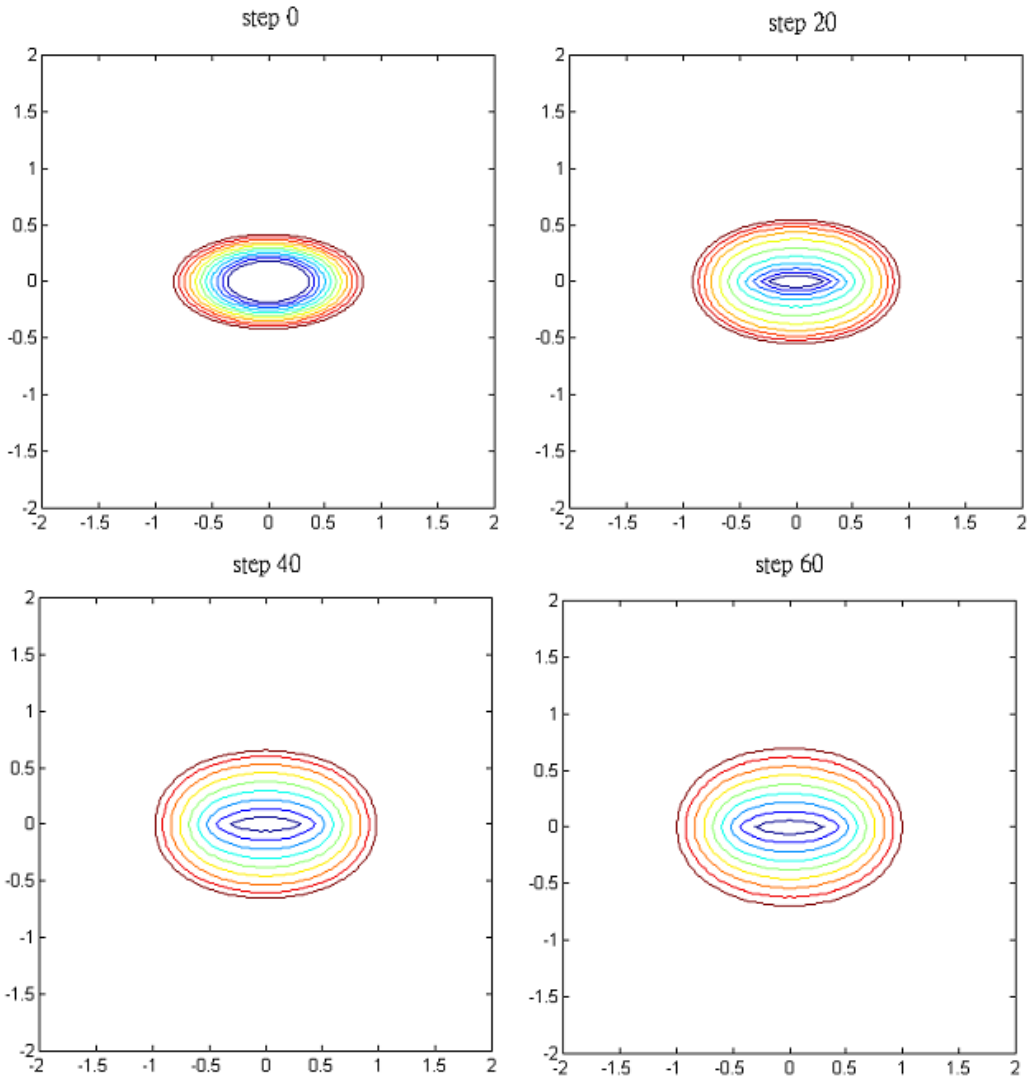

Figure 4: Reinitialization of ellipse

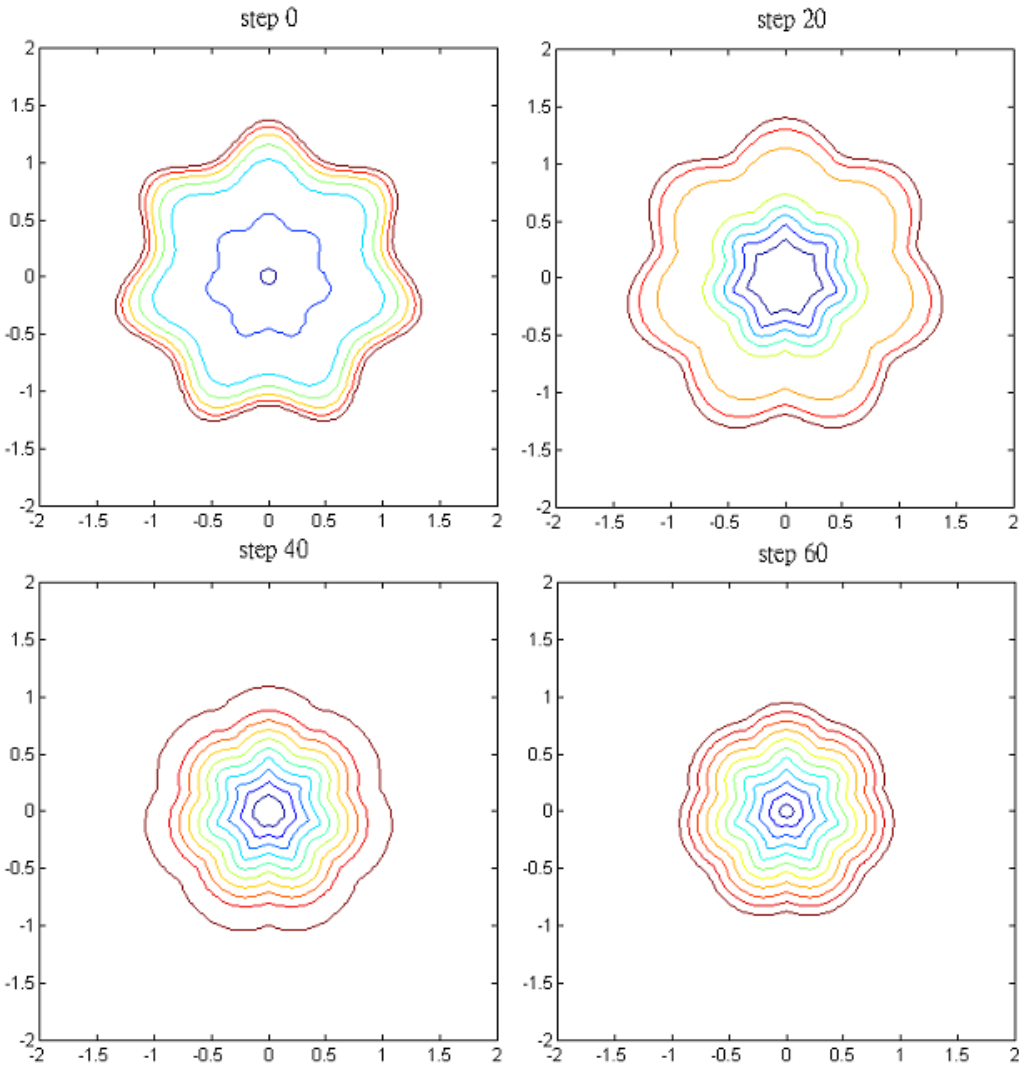

Figure 5: Reinitialization of plum blossom shape

Remark

1. From the figures of two examples, we find that the level function becomes signed distance function from interface to Ω^+ and Ω^- within a neighborhood of interface. As the time steps increase, the width of neighborhood also increase. The reason is that the information flows from interface to Ω^+ and Ω^-.

2. After reinitialization process, the level function becomes signed distance function in a neighborhood of interface and we can not write the level function by mathematical formulation exactly.

3. There is no clear condition of stopping criterion for reinitialization process. The iteration steps is determined by the width of signed distance function neighborhood of interface. If you want the width wider, the iteration steps needed more.

4. Eq.(6) is solved in whole domain and the above two examples are located at center region of computation domain, so if we just want a signed distance function of just a neighborhood of interface, we do not need the boundary condition.But if the interface contacts with the boundary, how do we set the boundary condition for Eq.(6)? We will give the idea and algorithm in next section.

3.7 Special Case and Boundary Condition

To handle Eq.(6) with interface contacts with boundary,we must be very carefully on boundary condition. Since the ENO and WENO scheme may need more than two points to compute, so on the boundary we have better not use ENO or WENO scheme, instead we use first-order Godunov's method to handle since this method just need two points to compute the first derivative term.Here we give the idea of how to set boundary condition for $1D$ case :

Concept of Boundary
If the left end point x_1 is in Ω^+ and we want compute ϕ_x. A ghost point x_0 on the left side of x_1 is needed.Since x_1 is in Ω^+ which means interface is on the right side of x_1 and x_0 is on the left side of x_1, so $\phi(x_0) \geq \phi(x_1)$. Similarly, if x_1 is in Ω^- then $\phi(x_0) \leq \phi(x_1)$

Although the point x_0 does not exist in the domain Ω, but the role of this point is important. We will give code of algorithm as below

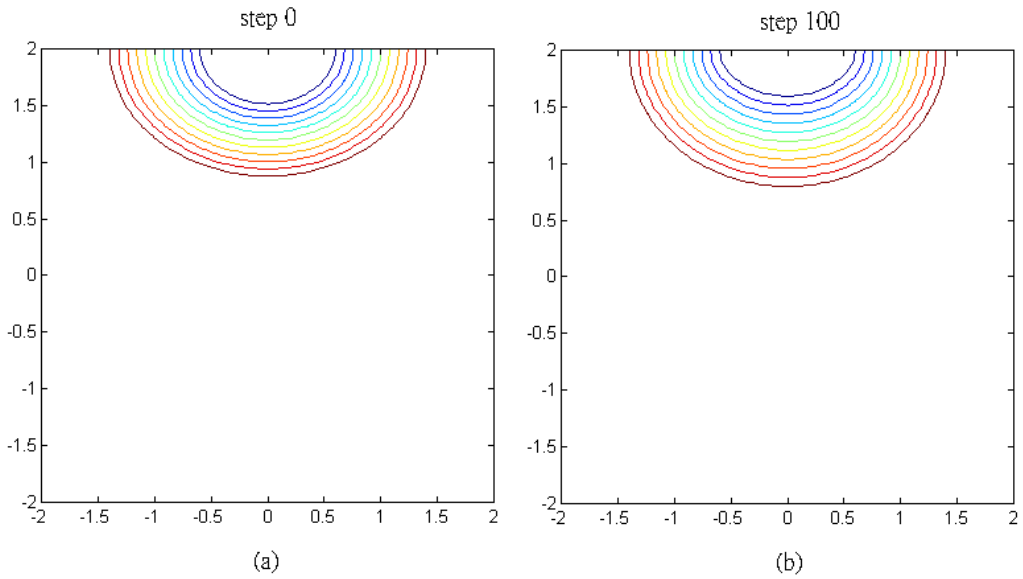

Figure 6: Contour of special case example

Algorithm

For x-direction and left side boundary

```
do j=1,n
   if (phi(1,j)>0d0) then
      if (phi(1,j)>=phi(2,j)) then
         DX(1,j)=(phi(2,j)-phi(1,j))/h
      else
         DX(1,j)=0d0
      end if
   else
      if (phi(1,j)<=phi(2,j)) then
         DX(1,j)=(phi(2,j)-phi(1,j))/h
      else
         DX(1,j)=0d0
      end if
   end if
end do
```

where DX denotes ϕ_x and the three other boundary are as same method to handle. We also suggest that the points adjacent boundary also use Go-

dunov's method, for $1D$ case we mean x_2. Here we contour a ellipse

$$\phi(x,y) = \sqrt{(\frac{x}{1})^2 + (\frac{y-2}{0.8})^2} - 1$$

for $[-10\Delta x : 2\Delta x : 10\Delta x]$ before and after reinitialization with boundary condition.(See figure6)

Until here, we have presented all process of reinitialization.No matter the interface is concave or convex, the reinitialization process all works that is if we can write down the level function of interface, we can reinitialize it to signed distance function even in the whole domain. The reinitialization process is very useful since it can calculate normal or curvature accurately, so it is really a very good and valuable scheme.

3.8 Normal Extension

In this section we will discuss the normal extension process. If u is some quantity which is defined on the interface, we can extend u in the normal direction.That is along the normal direction, the quantity u is constant. To achieve this purpose, we solve a Hamilton-Jacobi equation to steady state which is :

$$u_t + S(\phi)\frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla u = 0 \tag{21}$$

where $S(\phi)$ is signed function and we can discretize it by Eq.(20) which we have mentioned before.

If the condition $\frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla u = 0$ which means $\frac{\partial u}{\partial \mathbf{n}} = 0$ is satisfied, then we say that u is constant along the normal direction. Since $\frac{\partial u}{\partial \mathbf{n}} = 0$, another meaning is that u will not flow out of interface if u is heat quantity and we solve heat equation on the interface. So the purpose of the Eq.(21) is to achieve $\frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla u = 0$ by iteration.

To extend u off the interface, first we need to give quantity u on the interface or a small band of interface which says N_1, since we want u extended to a larger neighborhood N_2, so first we need to take reinitialization process in N_2 and then $|\nabla\phi| = 1$ in N_2 which reduces Eq.(21) to

$$u_t + S(\phi)\nabla\phi \cdot \nabla u = 0 \tag{22}$$

The concept of solving Eq.(22) is the same as solving Eq.(6). Upwind's concept is again used. For x-direction, if $S(\phi)\phi_x > 0$,then $u_x = D^- u$ and ϕ_x

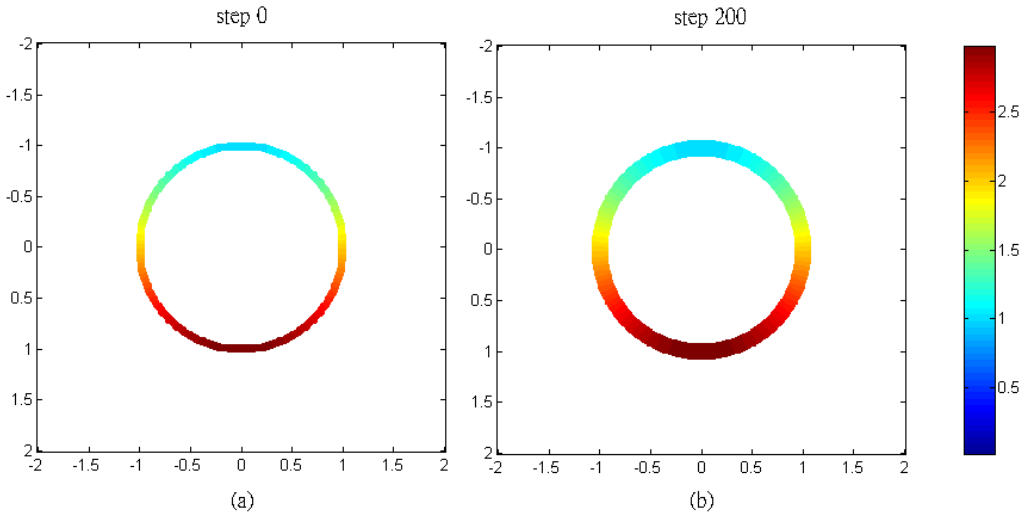

Figure 7: Normal extension of u for convex case

in this equation is discretized by central difference. Otherwise if $S(\phi)\phi_x < 0$ then $u_x = D^+u$. The ENO or WENO higher order schemes which have mentioned above can again be used and once we used higher order scheme, the neighborhood N_1 should be more wide since these methods need more points to calculate. Note that we use forward Euler method in time. Eq.(22) take the information off the interface to outside and inside with speed 1. We take some simple examples which we use WENO-3 and forward Euler in time.

Example.4 We let the level function be $\phi(x, y) = \sqrt{x^2 + y^2} - 1$ which is signed distance function so we do not need reinitialization process. We define $u = \sin(\theta) + 2$ on the grid node (i, j) which satisfied $|\phi(i, j)| \leq 4\Delta x$ and $u = 0$ on the others. Note that u is constant along normal direction in $|\phi(i, j)| \leq 4\Delta x$ and we want to extend u to $|\phi(i, j)| \leq 8\Delta x$. After 200 steps of iterations we will find that the value u have been extended to $|\phi(i, j)| \leq 8\Delta x$ and also constant along normal direction. (See figure7)

Example.5 We test another concave case $\phi(r, \theta) = (r - 0.5 + 0.1r\sin(7\theta))^3$ which need reinitialization process at first. We give initial value of u is as Example.4 and extend it to the same neighborhood.(See figure(8))

Remark
Note that if we want to extend the value off the interface to whole domain

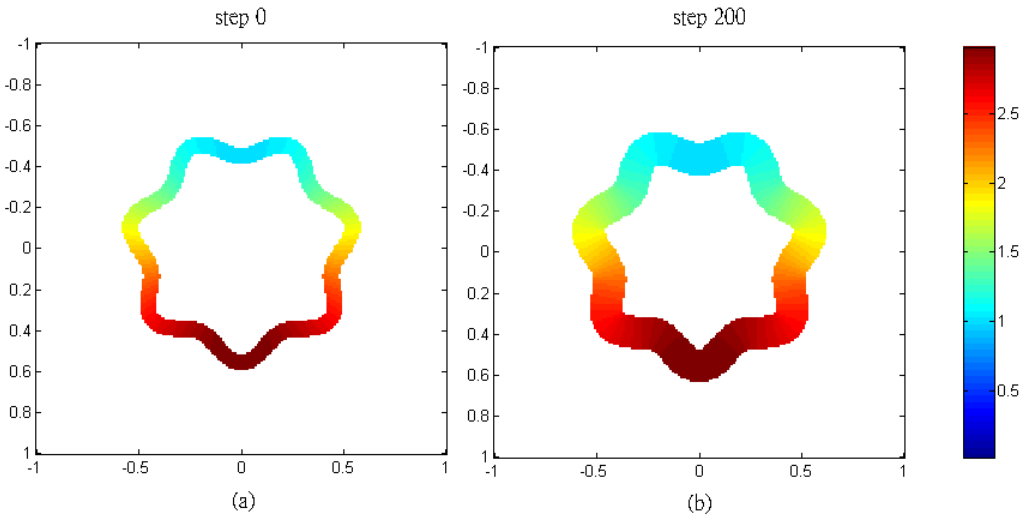

Figure 8: Normal extension of u for concave case

or more thick band, it is fine for convex case, but the concave case will not be allowed since the point far from interface will cause shock, that is there will be two normal lines intersect and cause terrible result. So the normal extension is meaningless for concave case if the extension band is too thick.

3.9 Solving Heat Equation on Interface

3.9.1 Surface Gradient and Surface Laplace

Recently, solving PDE on interface becomes more and more popular, such as solving heat equation on circle. Although circle has polar coordinate, if the interface is other shape which is irregular, then there will be no coordinate system to handle. So we use level set method to handle which is just Cartesian coordinate. To solve heat equation on interface, we introduce surface gradient ∇_s and surface laplace Δ_s. So the heat equation on interface is as below:

$$u_t = \Delta_s u = \nabla_s \cdot \nabla_s u \tag{23}$$

where ∇_s is defined as a operate $\nabla_s = P \cdot \nabla$ [15]. Note that P is a projection matrix which is defined as :

$$P := I - \frac{\mathbf{n} \otimes \mathbf{n}}{\|\mathbf{n}\|^2} \tag{24}$$

Again, if ϕ is signed distance function, we reduce Eq.(24) to

$$P := I - \mathbf{n} \otimes \mathbf{n} \tag{25}$$

and we refer [18] to simplify the surface laplace form as :

$$\Delta_s u = \nabla_s \cdot \nabla_s u = \Delta u - \frac{\partial^2 u}{\partial \mathbf{n}^2} - \kappa \frac{\partial u}{\partial \mathbf{n}} \tag{26}$$

where κ is curvature of interface.

The explanation of Eq.(26) is very easy.We know that Δ is diffusion term of whole domain, Δ_s denote diffusion on interface and $(\frac{\partial^2}{\partial \mathbf{n}^2} + \kappa \frac{\partial}{\partial \mathbf{n}})$ denotes the diffusion on normal direction of interface.So

$$\Delta = \Delta_s + \frac{\partial^2}{\partial \mathbf{n}^2} + \kappa \frac{\partial}{\partial \mathbf{n}}$$

and then we get the operate :

$$\Delta_s = \Delta - \frac{\partial^2}{\partial \mathbf{n}^2} - \kappa \frac{\partial}{\partial \mathbf{n}}$$

3.9.2 Outline of The Method

Since we solve heat equation on interface as Eq.(23) which means the heat just diffuse on interface and we have the condition that the derivative of u along normal direction is 0 which is $\frac{\partial u}{\partial \mathbf{n}} = 0$. Since we just define values on grid node, so we need to extend the u off interface within a band and solve Eq.(23) in this band. Since we know $\frac{\partial u}{\partial \mathbf{n}} = 0$ which coincide with $\nabla \phi \cdot \nabla u = 0$,so we solve Eq.(22) to extend u. The outline of whole procedure in one step is as follow :

Step1. Reinitialization the level function ϕ to signed distance function within a band by solving Eq.(6)

Step2. Give values on interface or in a small band of interface and then solve Eq.(22) to extend u to a thicker band which is thick enough to solve five-point laplace.

Step3. Solve Eq.(23) in the band which u have been extended and return to Step2.

Note that the new value we get will not be constant along normal direction any more, so we need to return to Step.2 to modify u to be constant along normal direction of interface.

3.9.3 Algorithm of Solving Heat Equation

The algorithm of Step1 and Step2 have introduced above, we will introduce the algorithm of Step3 which is how to solve Eq.(23). To solve Eq.(23), [18] used semi-implicit Crank-Nicholson method which is second-order in time and is stable when $O(\Delta t) = O(\Delta x)$. Note that we refer the scheme of [18] and use the symbol of [18]. We write the form of disretization of Eq.(23) as follow :

$$\begin{aligned}\frac{u^{n+1}-u^n}{\Delta t} =& \frac{\Delta u^{n+1}+\Delta u^n}{2} + \frac{3}{2}[-\kappa\frac{\partial u}{\partial \mathbf{n}} - \frac{\partial^2 u}{\partial \mathbf{n}^2}]^n \\ & - \frac{1}{2}[-\kappa\frac{\partial u}{\partial \mathbf{n}} - \frac{\partial^2 u}{\partial \mathbf{n}^2}]^{n-1} \end{aligned} \tag{27}$$

Note that the central difference is applied to $\Delta u, \mathbf{n}, \kappa, \frac{\partial u}{\partial \mathbf{n}} = \mathbf{n}\cdot\nabla u$ and $\frac{\partial^2 u}{\partial \mathbf{n}^2} = \mathbf{n}\cdot D^2 u\cdot\mathbf{n}$ where D^2 is the Hessian operator which is

$$D^2 u = \begin{bmatrix} u_{xx} & u_{xy} \\ u_{yx} & u_{yy} \end{bmatrix}$$

Since this semi-implicit method is two-step method, we need u^1 by u^0 at first which can be discretize as below :

$$\frac{u^1-u^0}{\Delta t} = \Delta u^1 + [\kappa\frac{\partial u}{\partial \mathbf{n}} + \frac{\partial^2 u}{\partial \mathbf{n}^2}]^0 \tag{28}$$

and then we can solve Eq.(27). Since we just solve Eq.(27) on a band, so we call this method as *Local Level Set Method* and we will have details as below

Local Level Set Method

The method which solving Eq.(27) on whole domain is called *Global Level Set Method* which need spending more time and computation and does not make sense when solving PDE on the interface. So we use *Local Level Set Method* which restricts the computational domain to be just a band(neighborhood) of interface and solve Eq.(27) on this band. Let us take three bands as above

$$\begin{aligned} N_1 &= \{(x_i, y_j) : |\phi(x_i, y_j)| < r_1\} \\ N_2 &= \{(x_i, y_j) : |\phi(x_i, y_j)| < r_2\} \\ N_3 &= \{(x_i, y_j) : |\phi(x_i, y_j)| < r_3\} \end{aligned}$$

where $0 < r_1 < r_2 < r_3$. [18] chooses $r_1 = 3\Delta x$,$r_2 = 6\Delta x$ and $r_3 = 9\Delta x$. Then we reinitialize ϕ within N_3 and take normal extension within N_2 and finally solve Eq.(27) in N_1.

Remark

1. Note that when solving Eq.(27) within the band N_1 by Crank-Nicholson method, we will encounter the data structure problem since we use five-point central difference to handle laplace term of Eq.(27) and the domain is just a band which is irregular. So we will need points which are not in N_1 but near N_1.

2. The linear system for u of Eq.(27) is symmetric positive definite. We have test some cases like concave case or irregular case and find that the linear systems for them are all symmetric positive definite and we solve the linear system by the package *DLSLSF* of *IMSL* in *FORTRAN 90*

3. To solve Eq.(27), we use the points not only in N_1 but also in $N_2 - N_1$ which near band N_1.So the points in $N_2 - N_1$ and near N_1 are the boundary points for solving Eq.(27) and we need boundary conditions for these points. [18] suggested the Dirichlet boundary conditions as extrapolation which is

$$u_{i,j}^{n+1} = 2u_{i,j}^{n} - u_{i,j}^{n-1}, \quad \forall (x_i, y_j) \in T_2 - T_1 \tag{29}$$

   and then these points are known and imposed to right hand side when solving the linear system for Eq.(27).

3.10 Numerical Results

Here we just test one example which the interface is static and the case of solving PDE on moving interface will be introduced in later chapter. We use the example in [18] and try to reproduce the result.

Example.6 Now we want to solve heat equation on a unit circle which means solving Eq.(23) on unit circle. On the circle, the surface laplace can be simplified to $\Delta_s = \frac{1}{r_0^2}\frac{\partial^2}{\partial\theta^2}$ where r_0 is radius and θ is central angle of x-axis. The reason we turn Eq.(23) to polar coordinate is to find the exact solution of Eq.(23) on the circle and we use Cartesian coordinate when handling numerical process.So we know that as $r_0 = 1$ then

$$\frac{\partial u(\theta, t)}{\partial t} = \frac{\partial^2 u(\theta, t)}{\partial \theta^2} \tag{30}$$

and the exact solution of Eq.(30) is

$$u(\theta, t) = e^{-t} \sin\theta + c \tag{31}$$

During our numerical process, we let $\phi(x, y) = \sqrt{x^2 + y^2} - 1$ denotes the level function of unit circle and solve

$$\begin{aligned}\frac{\partial u(x, y, t)}{\partial t} =& \Delta_s u(x, y, t) \\ =& \Delta u(x, y, t) - \frac{\partial^2 u(x, y, t)}{\partial \mathbf{n}^2(x, y)} - \kappa(x, y) \frac{\partial u(x, y, t)}{\partial \mathbf{n}(x, y)}\end{aligned}$$

numerically and the computational domain is $[-2, 2] \times [-2, 2]$. The initial we give is that $u_0(\theta) = \sin\theta + 2$ which is constant along the normal direction of interface and the exact solution is

$$u(x, y, t) = e^{-t} \sin\theta + 2 = e^{-t} \sin\left(\arcsin \frac{y}{\sqrt{x^2 + y^2}}\right) + 2.$$

Note that we set total time $T = 2$ and $\Delta t = \frac{1}{4}\Delta x$ and the boundary condition is as Eq.(29). We compare the error using L^∞-norm within the band N_1. Reinitialization process does not need and normal extension is needed every time step. The result is in Table1.

N	L^∞	Ratio	Order
40×40	4.63E-2	-	-
80×80	9.42E-3	4.919	2.298
160×160	2.99E-3	3.150	1.655
320×320	1.22E-3	2.439	1.286

Table 1: Error in maximum norm

Remark

1. Since we use the normal extension in each time step and we just take few iterations(we take five iterations) of Eq.(22) to avoid spending too much time, so u may not be constant along the normal direction and this is one of error contribution.

N	L^∞	Ratio
40×40	2.51E-2	-
80×80	1.96E-2	1.277
160×160	2.06E-2	0.953
320×320	1.99E-2	1.037

Table 2: Error in maximum norm

2. Since $\Delta_s u = \Delta u - \frac{\partial^2 u}{\partial \mathbf{n}^2} - \kappa \frac{\partial u}{\partial \mathbf{n}}$ and when solving heat equation on unit circle, we set $\frac{\partial u}{\partial \mathbf{n}} = 0$ which means $\Delta_s u = \Delta u$ in mathematical formulation. If we just let $\Delta_s u = \Delta u$ in our numerical process, we will get terrible result as in Table.2 and the reason is as the same as above remark. So we can not drop $\frac{\partial^2 u}{\partial \mathbf{n}^2}$ and $\kappa \frac{\partial u}{\partial \mathbf{n}}$ in our numerical process.

4 A Coupled Volume of Fluid and Level Set Method

4.1 Introduction

To simulate two-phase flows, we need interface to separate two fluids, say fluid1 and fluid2 are inside and outside interface respectively. It is convenient to describe the interface by level set function but if we only use level set method to handle the motion of interface, we will encounter a significant problem which is the inner area of fluid1 will not be conserved.

So we add Volume-of-Fluid method to achieve mass conservation of inner area volume. Our concept is very simple, level set method is easy to compute the properties about interface(e.g. curvature or normal). On the other hand, VOF method is applied to inner area volume conservation.

Note that since both level set method and Volume-of-Fluid method are adapted to topology changing of interface, so the coupled VOF and level set method is also adapted to topology changing of interface which is a nature phenomenon of our real life. We will test the coupled method in topology changing case.

Here we briefly introduce the concept of Volume of Fluid. We define fraction function F on the cell center as follows:(See figure.9)
If the cell is whole in the fluid1 that is the cell is inside interface and the interface do not pass this cell, then we define the fraction function of this cell is $F = 1$ and if the cell is whole in fluid2, in the same way we define the fraction of this cell to be $F = 0$. The final case is that interface pass the cell and separate two fluids in the cell and in this case, we define fraction of this cell by weight of fluid1, that is if weight of fluid1 in this cell is 30%, then the fraction of this cell is defined to be $F = 0.3$. Note that if the cell which interface passes, then the fraction F must between 0 and 1.

In this chapter we mainly refer [17] which have introduced the coupled VOF and level set method very clearly. The symbols and figures we used in this article are referred from [17] and we give more details of the coupled method in our article since this coupled method is complicate. We need not only the concept of VOF but also concept of level set method which we have introduced in the above chapter and we will give the outline of the coupled in our article.

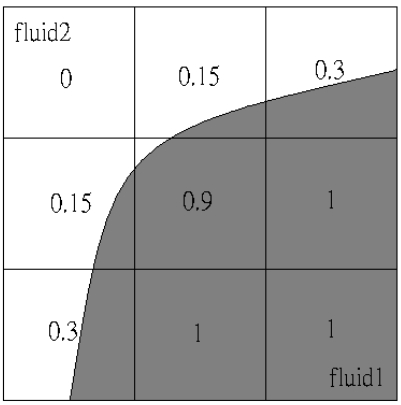

Figure 9: Concept of Volume of Fluid

4.2 The Governing Equation

In this section we will write down the governing equation which is Navier-Stokes equation. The interface properties is handled by level function ϕ which is negative inside interface and is positive outside interface. The Navier-Stokes equation is:

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}) = -\nabla p + \rho g + \nabla\cdot\mu[\nabla\mathbf{u} + (\nabla\mathbf{u})^T] - \sigma\kappa\nabla H \qquad (32)$$
$$\nabla\cdot\mathbf{u} = 0$$

$$\kappa = \nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}$$

$$H = \begin{cases} 1 & \text{if } \phi \le -1.5(\Delta x) \\ 0 & \text{if } 1.5\Delta x \le \phi \\ 0.5 + \frac{\phi}{3\Delta x} + \frac{\sin[\frac{2\pi\phi}{3\Delta x}]}{2\pi} & \text{if } |\phi| \le 1.5\Delta x \end{cases}$$

$$\rho = \rho_1 F + \rho_2(1-F)$$

$$\mu = \mu_1 F + \mu_2(1-F)$$

where H is a heaviside function smoothed over three grids and ρ_1, μ_1 are density and viscosity for fluid1 and ρ_2, μ_2 are density and viscosity for fluid2. κ is interface curvature which is evaluated by smooth level set function.

Note that $\mathbf{u}$ is defined at cell faces and ϕ, F, ρ, μ, p and g are all defined at cell center. (See figure10 and we especially thank [20] who provides this figure.)

4.3 The Advection Equations for Both Level Function and VOF

Since the interface will move by the velocity field $\mathbf{u}$ so we need to solve interface advection equations. Here we present two advection equations for both methods

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{33}$$

$$\frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0 \tag{34}$$

where F is fraction function and ϕ is level set function

Note that here we also solve the level set advection equation in order to compute normal of interface by smooth level function.

4.3.1 Algorithm of VOF Advection Equation

To solve advection equation of VOF, we rewrite Eq.(34) in a conservation form,

$$\frac{\partial F}{\partial t} + \nabla \cdot \mathbf{u} F = F \nabla \cdot \mathbf{u} \tag{35}$$

To solve Eq.(35) by VOF advection algorithm, we simply use split method like ADI method. So Eq(35) can be decomposed into two steps,

$$\frac{F^* - F^n}{\Delta t} + \frac{\partial u F^n}{\partial x} = F^* \frac{\partial u}{\partial x} \tag{36}$$

$$\frac{F^{n+1} - F^*}{\Delta t} + \frac{\partial v F^*}{\partial y} = F^* \frac{\partial v}{\partial y} \tag{37}$$

Before discretizing the two equations, we first note that the fraction function and level set function are defined at the cell center and the velocity field

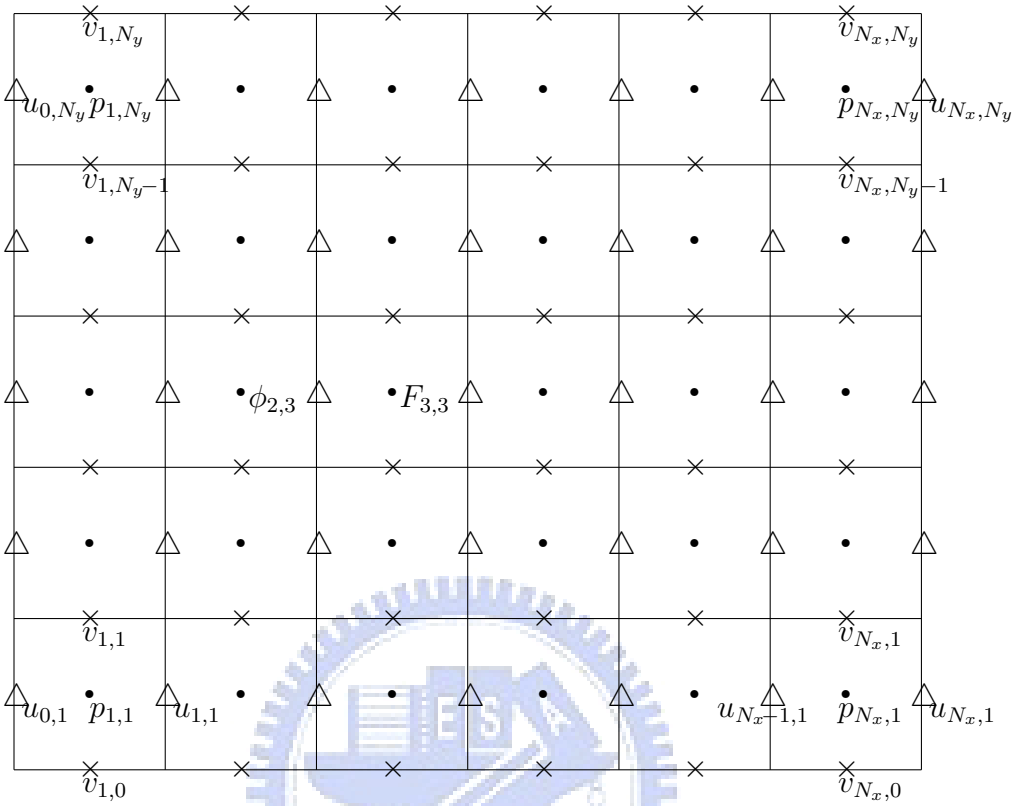

Figure 10: The computational domain using staggered grid

u is defined at cell faces. And now we begin to discretize the above two differential equations by finite volume method as below :

$$\begin{aligned} F^*_{i,j}\Delta x_i\Delta y_j =& F^n_{i,j}\Delta x_i\Delta y_j + (u\Delta t F^n\Delta y)_{i-1/2,j} - (u\Delta t F^n\Delta y)_{i+1/2,j} \\ & - F^*_{i,j}[(u\Delta t\Delta y)_{i-1/2,j} - (u\Delta t\Delta y)_{i+1/2,j}] \end{aligned} \tag{38}$$

$$\begin{aligned} F^{n+1}_{i,j}\Delta x_i\Delta y_j =& F^*_{i,j}\Delta x_i\Delta y_j + (v\Delta t F^*\Delta y)_{i,j-1/2} - (v\Delta t F^*\Delta y)_{i,j+1/2} \\ & - F^*_{i,j}[(u\Delta t\Delta y)_{i,j-1/2} - (u\Delta t\Delta y)_{i,j+1/2}] \end{aligned} \tag{39}$$

where $F_{i\pm1,j}$ and $F_{i,j\pm1}$ are unknowns.

Since F is defined on cell center, so $F_{i\pm1/2,j}$ and $F_{i,j\pm1/2}$ are define on the cell faces and we need to handle $F_{i\pm1/2,j}$ and $F_{i,j\pm1/2}$ when solving the VOF advection equation. If we just take average (eg.$(F_{i+1,j} + F_{i,j})/2 = F_{i+1/2,j}$), then the excessive computer error will be significant. The reason is that the fraction function F is not continuous, so we can not just take average to solve $F_{i\pm1/2,j}$ and $F_{i,j\pm1/2}$ for convenience.

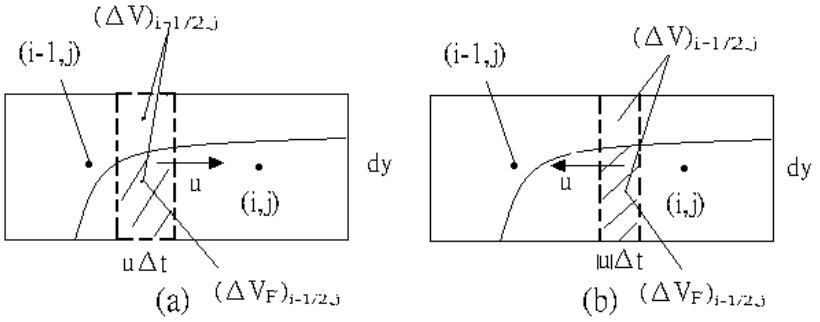

Figure 11: The evolution of volume flux for (a) $u > 0$ and (b) $u < 0$

Now, we realize that in the VOF advection process, the main problem is to calculate $F_{i\pm1/2,j}$ and $F_{i,j\pm1/2}$, so we apply geometry process to calculate.Suppose the cell say $F_{i,j}$, if we want to find $F_{i-1/2,j}$, we first need to know the velocity $u_{i-1/2,j}$ is negative or positive.If $u_{i-1/2,j}$ is positive which means that something flow into $F_{i,j}$ from $F_{i-1,j}$. On the other hand,if $u_{i-1/2,j}$ is negative, we know that something flow out of $F_{i,j}$ and then flow into $F_{i-1,j}$. Similary, $F_{i+1/2,j}$ and $F_{i,j\pm1/2}$ are in the same way to calculate.

The above is a very important concept, as we figure out the concept then [20] define $(\Delta V_F)_{i-1/2,j}$ and $(\Delta V)_{i-1/2,j}$ as the fluid1 volume and whole volume of the subregion $|u|\Delta t\Delta y$.(See figure.11)

$$(\Delta V_F)_{i-1/2,j} = \int F d(u_{i-1/2,j}\Delta t\Delta y_j) \tag{40}$$

$$(\Delta V)_{i-1/2,j} = \int d(u_{i-1/2,j}\Delta t\Delta y_j) \tag{41}$$

Then we get

$$F_{i-1/2,j} = \frac{(\Delta V_F)_{i-1/2,j}}{(\Delta V)_{i-1/2,j}} \tag{42}$$

Similarly, $F_{i+1/2,j}$ and $F_{i,j\pm1/2}$ are as the same concept.

Actually, we still do not seriously mention how to evaluate $(\Delta V_F)_{i-1/2,j}$ since it need to decide the interface shape which we will have details in the later section.

4.3.2 Algorithm of Level Set Advection Equation

When constructing the linear piecewise interface from fraction function, we need two elements, one is the fraction and another is the interface normal.

The interface normal can be evaluated by VOF fraction or level set function and we will compare these two cases to compute the interfacial normal in later section and find that on the same conditions the normal computed by the level set method will be better than the other.(See 4.10.3)

The reason is very clear, since the level set function is smooth and the fraction function is discontinuous, so level set function is better choice.

In order to calculate normal by level set function, we also need to move the level set function by this equation:

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

To solve above equation, we apply third-order WENO(weight essentially nonoscillatory) scheme to handle spatial and a third-order TVD Runge-Kutta scheme for the time evolution. Both schemes are mentioned in the above chapter. Here we briefly describe the scheme for the spatial:
For x-direction, if $u > 0$ then

$$\phi_{i,j,x} = \frac{1}{2}\left(\frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}\right) - \frac{\omega_-}{2}\left(\frac{\phi_{i+1,j} - 3\phi_{i,j} + 3\phi_{i-1,j} - \phi_{i-2,j}}{\Delta x}\right) \tag{43}$$

where

$$\omega_- = \frac{1}{1 + 2r_-^2}, r_- = \frac{\epsilon + (\phi_{i,j} - 2\phi_{i-1,j} + \phi_{i-2,j})^2}{\epsilon + (\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j})^2}$$

and if $u < 0$ then

$$\phi_{i,j,x} = \frac{1}{2}\left(\frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}\right) - \frac{\omega_+}{2}\left(\frac{\phi_{i+2,j} - 3\phi_{i+1,j} + 3\phi_{i,j} - \phi_{i-1,j}}{\Delta x}\right) \tag{44}$$

where

$$\omega_+ = \frac{1}{1 + 2r_+^2}, r_+ = \frac{\epsilon + (\phi_{i+2,j} - 2\phi_{i+1,j} + \phi_{i,j})^2}{\epsilon + (\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j})^2}$$

And for y-direction, if if $v > 0$ then

$$\phi_{i,j,y} = \frac{1}{2}\left(\frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y}\right) - \frac{\omega_-}{2}\left(\frac{\phi_{i,j+1} - 3\phi_{i,j} + 3\phi_{i,j-1} - \phi_{i,j-2}}{\Delta y}\right) \tag{45}$$

where

$$\omega_- = \frac{1}{1 + 2r_-^2}, r_- = \frac{\epsilon + (\phi_{i,j} - 2\phi_{i,j-1} + \phi_{i,j-2})^2}{\epsilon + (\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1})^2}$$

and if $v < 0$ then

$$\phi_{i,j,y} = \frac{1}{2}(\frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y}) - \frac{\omega_+}{2}(\frac{\phi_{i,j+2} - 3\phi_{i,j+1} + 3\phi_{i,j} - \phi_{i,j-1}}{\Delta y}) \tag{46}$$

where

$$\omega_+ = \frac{1}{1 + 2r_+^2}, r_+ = \frac{\epsilon + (\phi_{i,j+2} - 2\phi_{i,j+1} + \phi_{i,j})^2}{\epsilon + (\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1})^2}$$

Note that after solving Eq.(33), the interface move to new position and the level set function is no more signed distance function, so we need to reinitialize the new level set function to be signed distance function.We do not apply the level set reinitialization(i.e. solving Eq.6), instead we use another reinitialization process which related to the Volume-of-Fluid and we will have details in the later section.

4.4 Truncating Volume Fraction

After solving advection equation of VOF, there are some fraction $F_{i,j}$ which are less than 0 or more than 1. So we need to truncate these $F_{i,j}$ to be 0 or 1, that is

$$\begin{aligned} F &= 0 \quad \text{if } F < 0 \text{ or } \phi > \Delta x \\ F &= 1 \quad \text{if } F > 1 \text{ or } \phi < (-\Delta x) \end{aligned}$$

where ϕ is after reinitialized.

Another important adjustment must be careful which [3] has emphasized. In the *Bookkeeping Adjustments* of [3],it says that if $F_{i,j}^n$ is far from interface and $F_{i,j}^n = 0$ then after one time step $F_{i,j}^{n+1}$ still must be zero but it may not be zero in the numerical process. So we need to do some adjustment as follow:

$$\begin{aligned} F &= 0 \quad \text{if } F < \epsilon_F \\ F &= 1 \quad \text{if } F > 1 - \epsilon_F \end{aligned}$$

where $\epsilon_F = 10^{-6}$. If we do not do this process of adjustment, we will get very terrible result which will lead to the false of simulation of two-phase flows.

4.5 Outline of Coupled VOF and Level Set Method

Since the coupled VOF and level set method is complicate and have many steps to achieve which have priorities, so we give the outline of one time step of this coupled method :

Step1 Solve the Navier-Stokes equation which we have introduced in above chapter and then we can get velocity field $\mathbf{u}$ which can move interface and both fraction and level function.

Step2 Solve Eq.(34) to move fraction by $\mathbf{u}$ and get new fraction function.

Step3 Solve Eq.(33) to move interface and level function to new position by $\mathbf{u}$.

Step4 Reconstruct interface by PLIC which using F and $\mathbf{n}$ to decide the position of interface segment.

Step5 Calculate the advected volume in the interface cell that is to calculate $F_{i\pm1/2,j}$ and $F_{i,j\pm1/2}$ which needed in Step2. Note that this step must after interface reconstruction.

Step6 Determine two end points of each interface segment in each interface cell.

Step7 Reinitialize level set function within a band of interface by calculating the exact distance form cell center of neighborhood cell to interface segment. Finally, we return to Step1

We will give details of Step4 to Step7 in later sections.

Remark

Note that in Step4 we need $\mathbf{n}$ which we use level function to evaluate. Since Step3 just moves interface and ϕ have not reinitialized yet, so ϕ will not be signed distance function. Since we reinitialize ϕ within a band in each time step previous and we just move level function a little in one time step forward, so we still evaluate $\mathbf{n}$ in each interface cell in Step4 since the level function within a band of interface is not far from signed distance function(i.e. Similar to signed distance function). In this case $\mathbf{n}$ is evaluated by $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ and this time $|\nabla\phi|$ must be computed.

4.6 Interface Reconstruction

VOF method can be decomposed into two steps, one is fraction advection which we have mentioned in the above section and the other is interface reconstruction. To reconstruct the interface, we suppose that in each fraction cell which satisfying $0 < F < 1$, the interface is a segment which can be described as a linear equation $ax + by = c$, so after we connecting all interface segments, the interface will be piecewise linear. Thus we call this reconstruction method as PLIC (piecewise linear interface construction).

To reconstruct segments in each fraction cell which satisfies $0 < F < 1$, we need normal $\mathbf{n} = (n_x, n_y)$ which is equal to the pair (a, b) of $ax+by = c$ and it is easy to compute by level set function,$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ which we have mentioned in outline of the method.Once we have $\mathbf{n}$, the normal vector of segment can be determined, but the exact location of segment have not decided yet. For this reason,[17] introduced s, which means the distance from interface segment to the corner of an interface cell(i.e. s means the intercept c of $ax + by = c$). For generality, we choose the corner that is inside the interface and to be the farthest from the segment. As we get s and $\mathbf{n}$, we finally can have the exact location of the interface segment of an interface cell. So the interface reconstruction is to find s by F and $\mathbf{n}$. Finding the relation of F and s, we need to consider many cases of interface configurations, so we need to reduce the geometrical cases.

[17] suggested to use $|\mathbf{n}|$ instead of $\mathbf{n}$ and we can reduce the interface configurations and we can just consider only four cases as figure12. Since F is known, we can have the relation which we refer [17]:

$$2Fdxdy = dx_0dy_0 - \frac{dy_0}{dx_0}\langle dx_0 - dx\rangle^2 - \frac{dx_0}{dy_0}\langle dy_0 - dy\rangle^2 \tag{47}$$

where $\langle x\rangle = \max(x, 0)$ and dx_0 and dy_0 means the intercept of x-direction and y-direction of interface segment as shown in figure12. Note that dx_0 and dy_0 of Eq.(47) are not trivial and do not easy to compute, so we need substitute them by some simple forms.

From geometrical concept, we can easily find that $s = |n_x|dx_0 = |n_y|dy_0$ and then we can simplify Eq.(47) into a more simple form as:

$$2Fdx_1dy_1 = s^2 - \langle s - dx_1\rangle^2 - \langle s - dy_1\rangle^2 \tag{48}$$

where $dx_1 = |n_x|dx$ and $dy_1 = |n_y|dy$ .The vapor volume is

$$2(1 - F)dx_1dy_1 = (s_m - s)^2 - \langle s_m - s - dx_1\rangle^2 - \langle s_m - s - dy_1\rangle^2 \tag{49}$$

where $s_m = dx_1 + dy_1$.

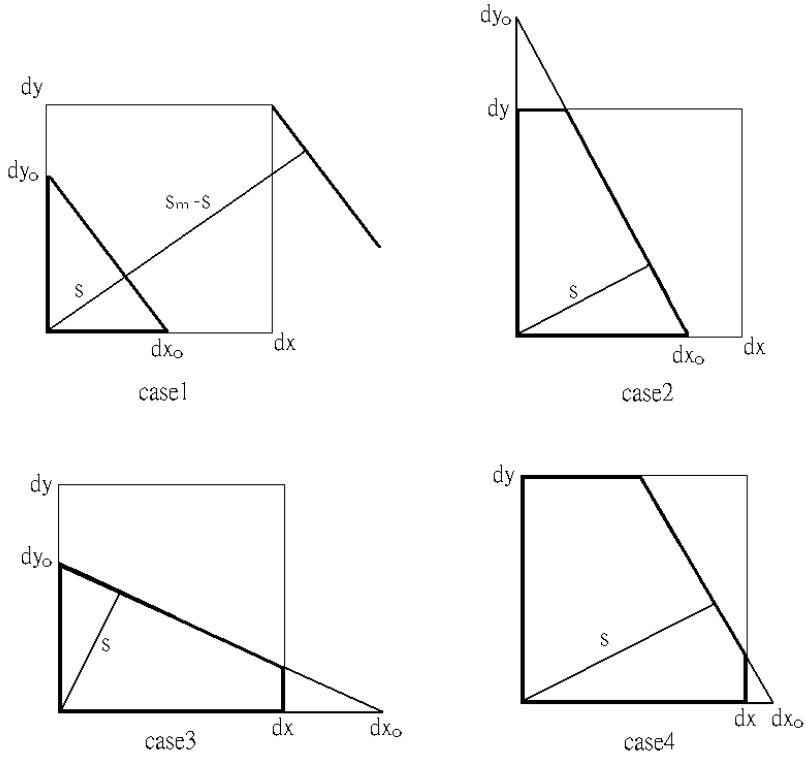

Figure 12: Four cases for interface configurations and fluid1 is inside thick line

Note that Eq.(48) and Eq.(49) are easy to compute and we want to combine two equations into one equation. [17] defined $F_c = \min(F, 1-F)$,$s_c = \min(s, s_m - s)$,$dx_c = \max(dx_1, dy_1)$ and $dy_c = \min(dx_1, dy_1)$. Now we can combine two equations into one as follow:

$$2F_c dx_c dy_c = s_c^2 - \langle s_c - dy_c \rangle^2 \tag{50}$$

From Eq.(50), we can clearly find the relation between s and F, after simplification we can get:

$$s_c = \sqrt{2F_c dx_c dy_c} \qquad \text{if } F_c < \frac{dy_c}{2dx_c} \tag{51}$$

$$= F_c dx_c + 0.5 dy_c \qquad \text{if } F_c \geq \frac{dy_c}{2dx_c} \tag{52}$$

and then we have

$$s = s_c \qquad \text{if } F \leq 0.5 \tag{53}$$

$$= s_m - s_c \qquad \text{if } F > 0.5 \tag{54}$$

After getting s finally, we have done all the interface reconstruction algorithm. In the next section, we will accomplish the fraction advection part with how to evaluate $F_{i\pm1/2,j}$ and $F_{i,j\pm1/2}$ accurately.

4.7 Evaluate Advecting Volume

In the previous section, we have introduced VOF advection algorithm and have mentioned $(\Delta V_F)_{i-1/2,j}$ and $(\Delta V)_{i-1/2,j}$ already. But we did not handle these because as handling $(\Delta V_F)_{i-1/2,j}$ and $(\Delta V)_{i-1/2,j}$ we need s which we have just got, so we did not calculate these two in the above section. Since we have s now, we can evaluate $F_{i\pm1/2,j}$ and $F_{i,j\pm1/2}$. A donor cell concept needed here and what is donor cell? Let us take a example here:For a cell $F_{i,j}$, and the left-side x-direction on the cell face is $u_{i-1/2,j}$, then if $u_{i-1/2,j}$ is positive then the donor cell of $F_{i-1/2,j}$ is $F_{i-1,j}$, otherwise if $u_{i-1/2,j}$ is negative, then the donor cell of $F_{i-1/2,j}$ is $F_(i,j)$. For the simple case first, if the donor cell is 0 or 1, then $(\Delta V_F)_{i-1/2,j} = (\Delta V)_{i-1/2,j}$ and $F_{i-1/2,j} = 0$ or 1. The other three cases are similar. Next [17] introduced $(\delta V_F)_0$ to denote the volume in the subregion $\delta x \times \delta y$ where $0 \leqslant \delta x \leqslant dx$ and $0 \leqslant \delta y \leqslant dy$. Note that the subregion contains the origin point of s.

Now we want to calculate the advected volume which equals to calculate $(\delta V_F)_0 = F\delta x\delta y$. We need the fraction of the subregion $\delta x \times \delta y$ say F_c and it can be calculated from the s_c(Here s_c will be recalculated) of the subregion since we know that s_c and F_c have a relationship.So first we compute s_c and then compute F_c and finally determine whether vapor or not. Note that n_x and n_y do not need recompute and just as the same. The procedure of finding F_c is same as Eq.(51)-(54). So from above equations of (51)-(54), we can regain F_c as the follows:

$$F_c = \frac{0.5s_c^2}{\delta x_c \delta y_c} \qquad \text{if } s_c < \delta y_c \tag{55}$$

$$= \frac{s_c - 0.5\delta y_c}{\delta x_c} \qquad \text{if } s_c \geq \delta y_c \tag{56}$$

and

$$(\delta V_F)_0 = \delta x\delta y F_c \qquad \text{if } s \leq 0.5s_m \tag{57}$$

$$= \delta x\delta y(1 - F_c) \qquad \text{if } s > 0.5s_m \tag{58}$$

where $\delta x_c = \max(|n_x|\delta x, |n_y|\delta y)$, $\delta y_c = \min(|n_x|\delta x, |n_y|\delta y)$, $s_m = \delta x_c + \delta y_c$ and $s_c = \min(s, \langle s_m - s\rangle)$. There are too many symbols to represent and for convenience we write $(\delta V_F)_0 = (\delta V_F)_0(n_x, n_y, \delta x, \delta y, s)$ which means that we calculate $(\delta V_F)_0$ by these five elements.

Since the interface configurations are not all the same so we need to split cases to discuss. Let us take a example as follows:For x-direction and left face of cell (i, j), if $n_x u \geq 0$,the subregion must take $|u|\Delta t \times dy$ and the subregion include the origin of s, then

$$\begin{aligned}(\Delta V_F)_{i-1/2,j} &= (\delta V_F)_0(n_x, n_y, |u|\Delta t, dy, s)\\ (\Delta V)_{i-1/2,j} &= |u|\Delta t\Delta y\\ F_{i-1/2,j} &= \frac{(\Delta V_F)_{i-1/2,j}}{(\Delta V)_{i-1/2,j}}\end{aligned}$$

Note that n_x, n_y and s are given by donor cell which we have mentioned.

On the other hand, if $n_x u < 0$, it means that the subregion of advected $|u|\Delta t \times dy$ do not include the origin of s, but the vapor subregion $(dx - |u|\Delta t) \times dy$ contains the origin of s.So in this case, if we want to calculate $(\Delta V_F)_{i-1/2,j}$, we need calculate vapor volume $(\delta V_F)_0(n_x, n_y, dx - |u|\Delta t, dy, s)$ first and then use total volume of donor cell $Fdxdy$ to minus the vapor volume which is:

$$\begin{aligned}(\Delta V_F)_{i-1/2,j} &= Fdxdy - (\delta V_F)_0(n_x, n_y, dx - |u|\Delta t, dy, s)\\ (\Delta V)_{i-1/2,j} &= |u|\Delta t\Delta y\\ F_{i-1/2,j} &= \frac{(\Delta V_F)_{i-1/2,j}}{(\Delta V)_{i-1/2,j}}\end{aligned}$$

Similarly, for the y-direction:

$$\begin{aligned}(\Delta V_F)_{i,j-1/2} &= (\delta V_F)_0(n_x, n_y, dx, |v|\Delta t, s) && \text{if } n_y v \geq 0\\ &= Fdxdy - (\delta V_F)_0(n_x, n_y, dx, dy - |v|\Delta t, s) && \text{if } n_y v < 0\\ (\Delta V)_{i,j-1/2} &= |v|\Delta t\Delta x\\ F_{i,j-1/2} &= \frac{(\Delta V_F)_{i,j-1/2}}{(\Delta V)_{i,j-1/2}}\end{aligned}$$

4.8 Determining the End Points of Interface Segment

Since the interface of each interface cell is segment, so we need to decide the two end points of the interface segment. We use $\mathbf{dx}_{s1} = (dx_{s1}, dy_{s1})$ and $\mathbf{dx}_{s2} = (dx_{s2}, dy_{s2})$ as in [17] to represent two end points relative to the left-down corner of a interface cell.For convenience, we let $dx_1 \geq dy_1$ where $dx_1 = |n_x|dx$ and $dy_1 = |n_y|dy$ as above. If $dx_1 < dy_1$ then we rotate the cell to achieve $dx_1 \geq dy_1$ and let $s_c = \min(s, dx_1 + dy_1 - s)$ as in above section.

Now we only look case1 and case2 of figure.12 and we have:

$$\begin{aligned}
\mathbf{dx}_{s1} &= (\frac{s_c}{|n_x|}, 0) \\
\mathbf{dx}_{s2} &= (0, \frac{s_c}{|n_y|}) && \text{if } s_c < dy_1 \\
&= (\frac{s_c - dy_1}{|n_x|}, dy) && \text{if } s_c \geq dy_1 \\
dx_{s1,2} &= dx - dx_{s1,2} && \text{if } n_x(s - 0.5s_m) < 0 \\
dy_{s1,2} &= dy - dy_{s1,2} && \text{if } n_y(s - 0.5s_m) < 0
\end{aligned}$$

Since some cells have been rotated before, so we need to rotate back these cells which have been rotated. That is if $|n_x|dx < |n_y|dy$, the components for $\mathbf{dx}_{s1}$ interchange(i.e. dx_{s1} and dy_{s1} interchange for $\mathbf{dx}_{s1}$) and similar to $\mathbf{dx}_{s2}$. Finally, we can determine the end points exactly,

$$\mathbf{x}_{s1,2} = \mathbf{dx}_{s1,2} + \mathbf{x}_{i-1/2,j-1/2}$$

After determining end points of each interface cell, we connect each interface segment as piecewise linear and this is the reason why the interface reconstruct algorithm is called PLIC.

4.9 Reinitialization for Level Function

After Eq.(33) to move the interface to the new position say ϕ^{n+1}, the level function is no more signed distance function, so we need to reinitialize it to be signed distance function. Note that here the procedure of reinitialization is not to solve the equation $\phi_t + S(\phi)(|\nabla\phi| - 1) = 0$ to steady state, instead we apply VOF properties to calculate the distance between center of neighborhood cell to the interface segment.First step we need to record which cells are inside interface and which are outside and we use

$$S^{\phi} = \text{sign}(F - 0.5)$$

Then we compute ϕ within a band of interface:

1. Let $|\phi|$ to be large number
2. For each interface cell $0 < F < 1$
   (a) We first compute the distance from the cell center to interface by $d(\mathbf{x}) = \mathbf{n} \cdot (\mathbf{x} - \mathbf{x}_{s1})$ where $\mathbf{x}$ denotes cell center and $\mathbf{x}_{s1}$ is a end point which we have computed.

(b) Then for each neighborhood cell $F_{k,l}$ of $F_{i,j}$ where

$$|k-i| \leq 4 \text{ and } |l-j| \leq 4,$$

$|\phi_{k,l}|$ is the shortest distance of two choices:One is cell center $\mathbf{x}_{k,l}$ to interface segment of $F(i,j)$ and the other is $\mathbf{x}_{k,l}$ to $\mathbf{x}_{i+m/2,j+n/2}$ where

$$m = \max(-1, \min(1, k-i)) \text{ and } n = \max(-1, \min(1, l-j))$$

There are three cases in the procedure:

i. If $S^{\phi}_{k,l} \neq$sign$[d(\mathbf{x}_{i+m/2,j+n/2})]$ (As point A of figure13(a)), it means that the projection of $\mathbf{x}_{k,l}$ to interface segment is not in the cell (i,j), so the shortest distance is evaluated between $\mathbf{x}_{k,l}$ to cell face of (i,j) and we have

$$|\phi_{k,l}| = min(|\phi_{k,l}|, |\mathbf{x}_{k,l} - \mathbf{x}_{i+m/2,j+n/2}|)$$

ii. Else if $\mathbf{x}_{k,l}$ have been projected to interface cell (i,j), that is the point $\mathbf{x}_{k,l} - \mathbf{n}d(\mathbf{x}_{k,l})$ is in the cell (i,j), then we just calculate the distance from $\mathbf{x}_{k,l}$ to interface segment of cell (i,j), (As point B in figure.13(a))

$$|\phi_{k,l}| = min(|\phi_{k,l}|, |d(\mathbf{x}_{k,l})|)$$

iii. Else just take smaller distance from $\mathbf{x}_{k,l}$ to two end points, (As point C in figure13(a))

$$|\phi_{k,l}| = min(|\phi_{k,l}|, |\mathbf{x}_{k,l} - \mathbf{x}_{s1}|, |\mathbf{x}_{k,l} - \mathbf{x}_{s2}|)$$

3. In few cases the interface segment may lay on the cell face and the fraction is 1 or 0. The process is as follow:(See figure.13(b))
If the cell is 1 or 0 and has a neighborhood cell (i',j') where $|i'-i| \leq 1$ and $|j'-j| \leq 1$ satisfying $S^{\phi}_{i,j} \neq S^{\phi}_{i',j'}$, then we tag neighborhood cell (k,l) satisfying $|k-i| \leq 4$ and $|l-j| \leq 4$.Then $|\phi_{k,l}|$ is the shortest distance between $\mathbf{x}_{k,l}$ and $\mathbf{x}_{i+m/2,j+n/2}$,

$$|\phi_{k,l}| = min(|\phi_{k,l}|, |\mathbf{x}_{k,l} - \mathbf{x}_{i+m/2,j+n/2}|)$$

where $m = \max(-1, \min(1, k-i))$ and $n = \max(-1, \min(1, l-j))$

4. Finally,we determine the ϕ-value to be positive or negative by S^{ϕ}

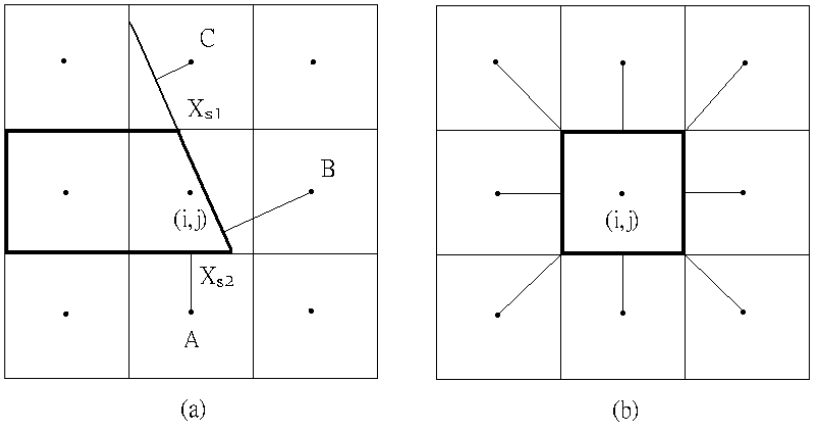

Figure 13: Reinitialization of level set

Remark
Note that every time we calculate ϕ of neighborhood cell must compare with the calculated value of itself and take the smaller value. For example:Suppose (i, j) and $(i+1, j)$ are both interface cell and (k, l) is a neighborhood cell of both interface cells.At first we let $\phi_{k,l} = 10$ and tag cell (i, j) and then we can get distance between two cells (k, l) and (i, j) by above procedure which must smaller than 10. So we take the smaller and get value of $\phi_{k,l}$. Next we tag cell $(i+1, j)$ and again compute the distance between two cells of (k, l) and $(i+1, j)$ and then compare with above computed value and take smaller. After handling each interface cell's reinitialization process, we get level set function ϕ which is signed distance function within a band of neighborhood.

4.10 Some Simple Test

Here we test our above schemes for a simple example. We set the interface to be an ellipse with level function:

$$\phi(x, y) = \sqrt{(\frac{x}{0.6})^2 + (\frac{y}{0.3})^2} - 1 \quad \text{with domain:}[-2, 2] \times [-2, 2] \tag{59}$$

Since the level function of the ellipse is not a signed distance function, we apply Eq.(6) to ϕ at first and then ϕ becomes a signed distance function.Note that ϕ is defined at cell center.

Next we need to calculate the original fraction F of interface cell. Since F and s have relation so we can first find s and then get F using the relation between s and F. Note that s is the distance from farthest corner in fluid1 to interface segment of a interface cell, so we define another ϕ say ϕ' :

$$\phi'(x,y) = \sqrt{(\frac{x}{0.6})^2 + (\frac{y}{0.3})^2} - 1 \tag{60}$$

This time ϕ' is defined at grid node. Tag interface cell (i,j) and we will have four values(See figure14) of ϕ' in each corner and they mean the distance from corner to interface of this cell. If the value of ϕ' is positive, then it is outside interface. Otherwise it is inside interface. Now we focus on the smallest value of negative corner and take it's absolute value to be s of this interface cell. Finally we apply Eq.(48) and then we have F in each interface cell. So we get the initial fraction function by above manner.

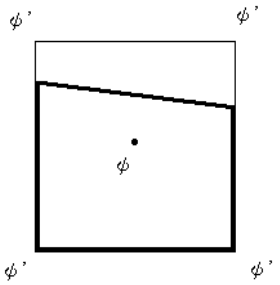

Figure 14: Locations of ϕ and ϕ'

Now we let the velocity field fixed with $\mathbf{u} = (-0.5x, 0.5y)$ which satisfying $\nabla \cdot \mathbf{u} = 0$ and apply coupled VOF and level set method with $\Delta t = \frac{1}{4}\Delta x$ as below:

4.10.1 Test of VOF Advection

We test our VOF advection equation by above simple example and we have the result as above:

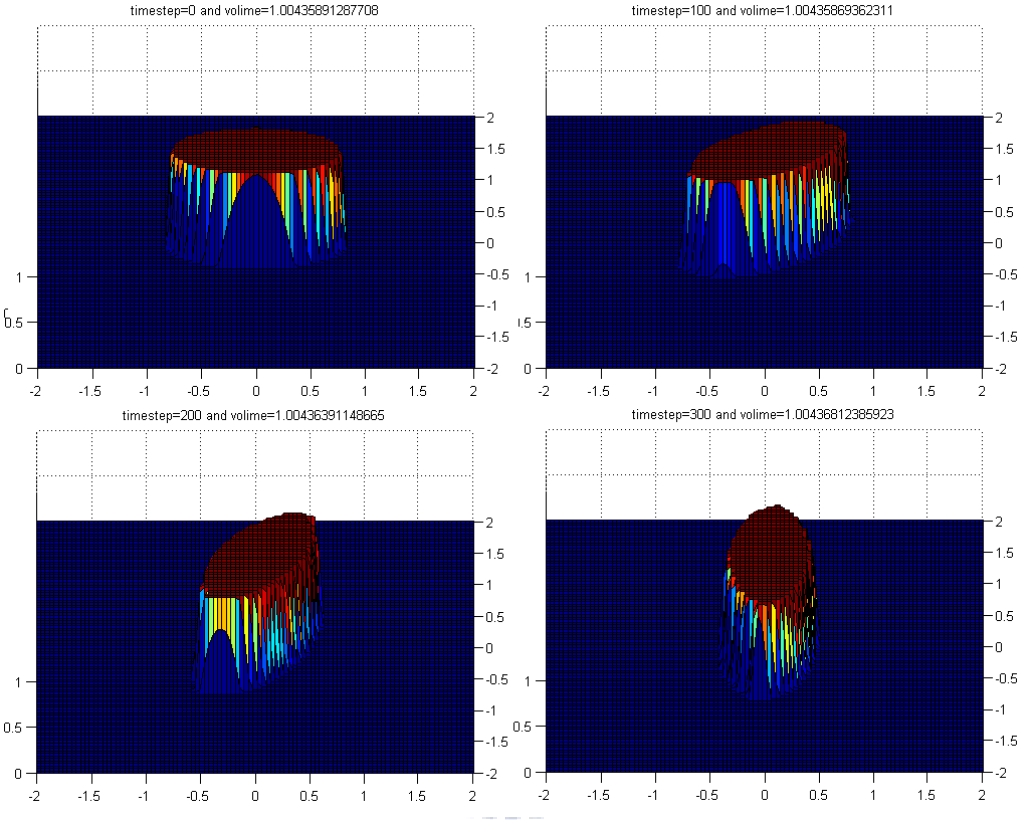

Figure 15: Test of fraction advection

Note that the inner volume is almost conserved and the volume variation is less than 0.1%. The ellipse rotated since the velocity field $\mathbf{u}$.

4.10.2 Test of Reinitialization

This time we test the reinitialization of level set which uses VOF method to handle and we have the above result

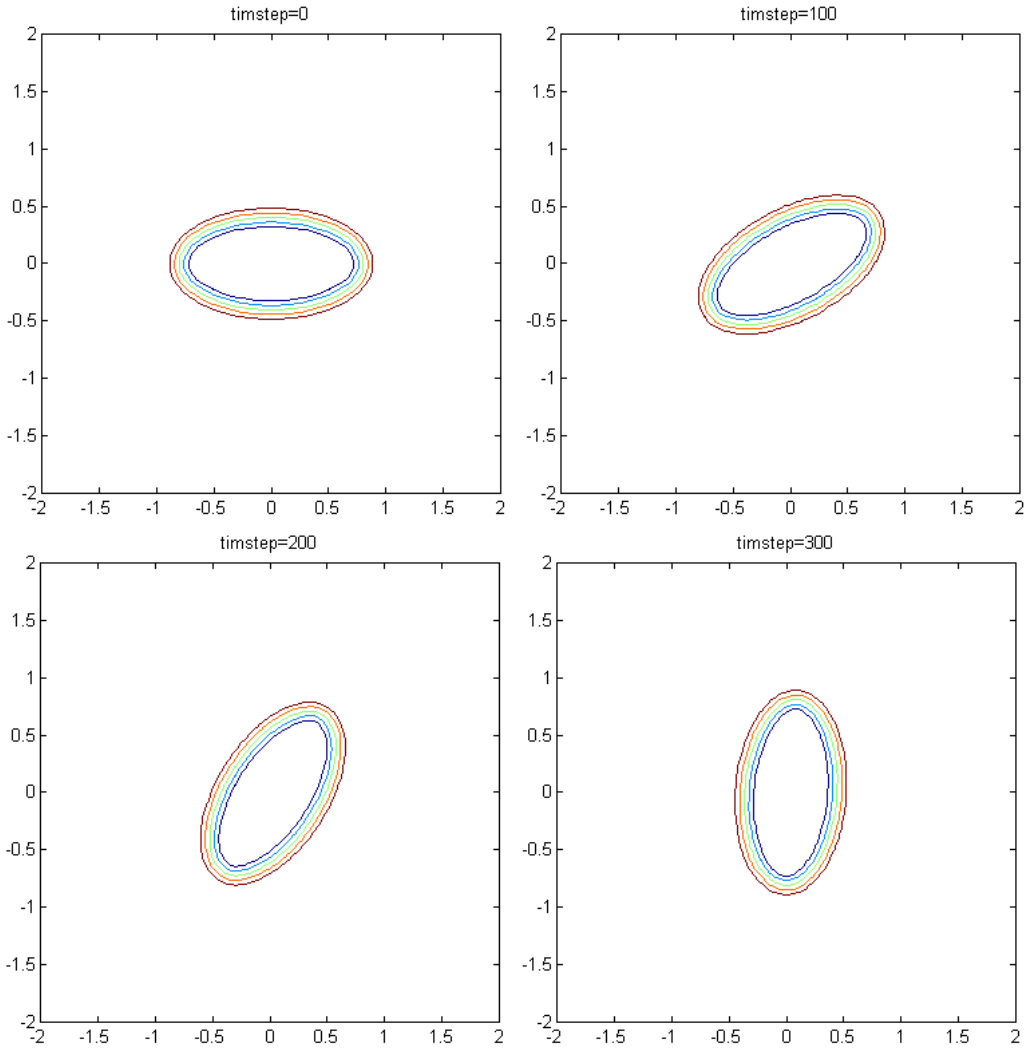

Figure 16: Test of Reinitialization

Here we contour five ϕ-values of $[-0.08, -0.04, 0, 0.04, 0.08]$ after reinitialization process in time steps $0, 100, 200$ and 300.

Remark
The above reinitialization process just need to do *1* time. If we use

$$\phi_t + S(\phi)(|\nabla\phi| - 1) = 0$$

to reinitialize ϕ, then we need some iterations to reach steady state in each time step which will spend much time than the VOF reinitialization which just do *1* time in one time step.

4.10.3 Test of Interface Reconstruction

Interface reconstruction is a important step of VOF method.Since we use segments to approximate interface, so the interface will be piecewise linear. Here we decide end points of each interface cell and then connect two end points of each interface cell to finish the piecewise linear interface. We have the result in figure17.

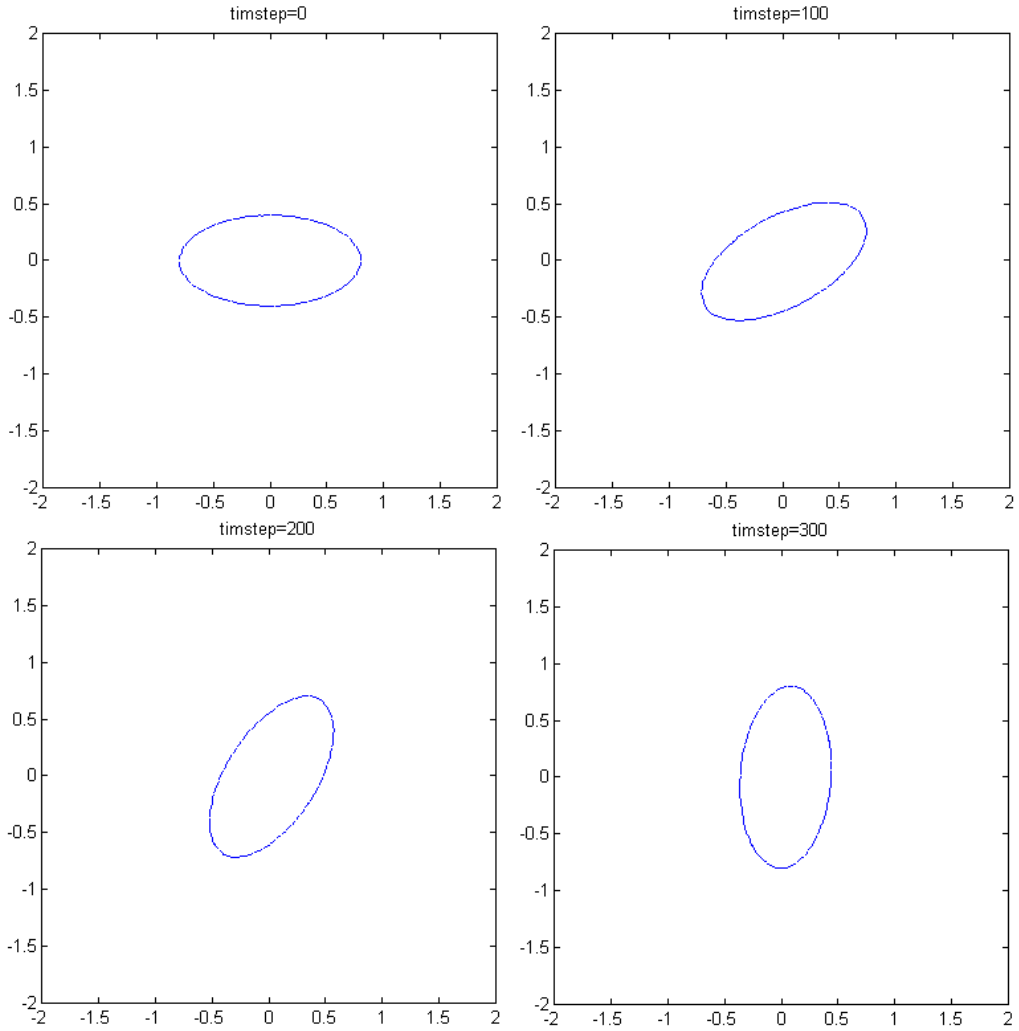

Figure 17: Test of Interface Reconstruction using coupled method

Remark

The above interface reconstruction is that we use coupled VOF and level set method. If we just use pure VOF method, will it be better or worse? The difference of pure VOF method and coupled method is how to calculate $\mathbf{n}$. If we use pure VOF method to calculate $\mathbf{n}$ that is using fractions to calculate normal, then we first calculate the normal of grid node which we refer [10],for example: normal of $(i+1/2, j+1/2)$

$$n_{x,i+1/2,j+1/2} = \frac{F_{i+1,j} - F_{i,j} + F_{i+1,j+1} - F_{i,j+1}}{2\Delta x}$$
$$n_{y,i+1/2,j+1/2} = \frac{F_{i,j+1} - F_{i,j} + F_{i+1,j+1} - F_{i+1,j}}{2\Delta x}$$

then take average to get cell center normal:

$$\mathbf{n}_{i,j} = \frac{\mathbf{n}_{i+1/2,j-1/2} + \mathbf{n}_{i-1/2,j-1/2} + \mathbf{n}_{i+1/2,j+1/2} + \mathbf{n}_{i-1/2,j+1/2}}{4}$$

We present the interface construction of just using VOF method in figure18. We find that using coupled method is better than using VOF only since when calculating normal, level set is smooth and fraction is discontinuous, so using level set to calculate normal is better.

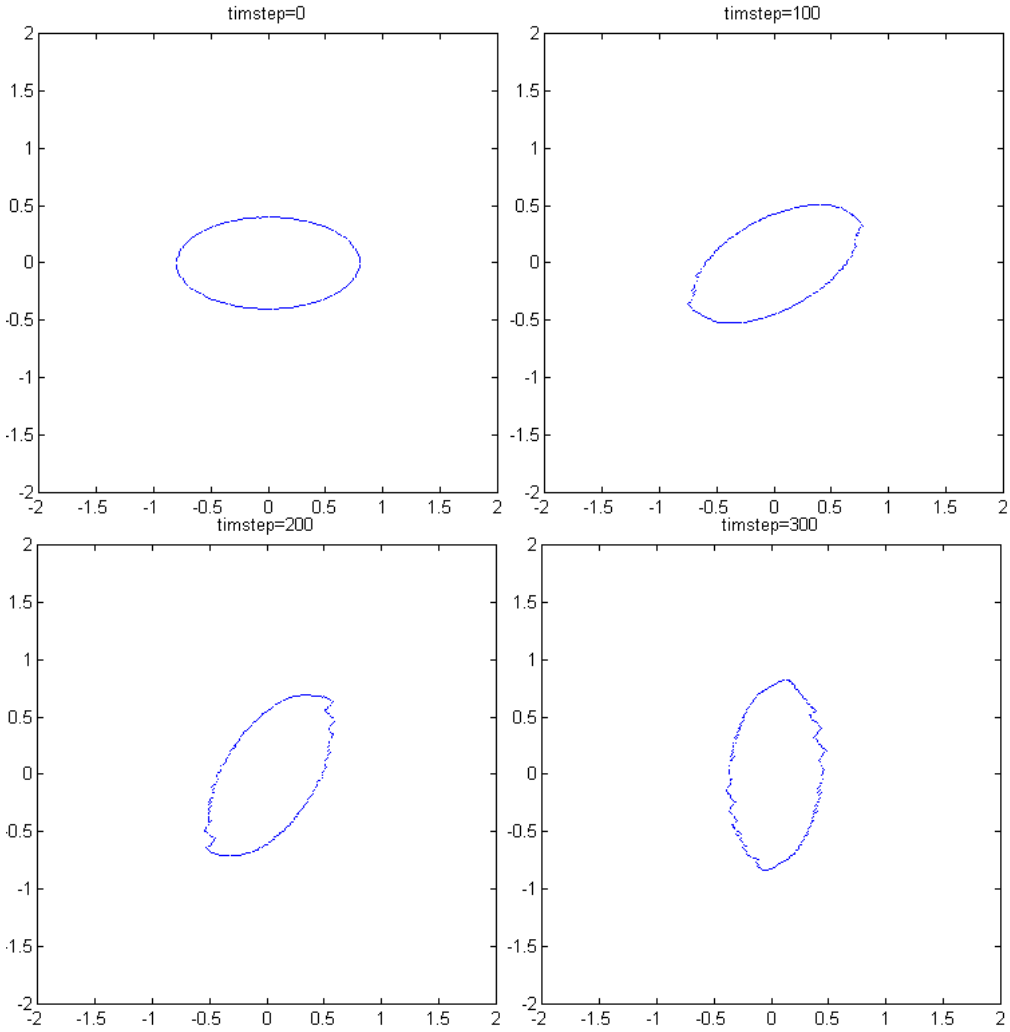

Figure 18: Test of Interface Reconstruction using pure VOF method

4.11 Numerical Results

In this section we will show our numerical results of simulating two-phase flows. Here we thank [22] who provided fluid code of solving Navier-Stokes equation with different densities ρ_1 and ρ_2 and viscosities μ_1 and μ_2.

Example.1
Our first example is to test single bubble simulation. Here we give our setting. Inside interface:$\rho_1 = 1$,$\mu_1 = 0.02$ and outside interface:$\rho_2 = 40$, $\mu_2 = 0.005$ and let $\sigma = 0.01$. We let initial interface to be a circle which center at $(0, -0.2)$ with radius $r = 0.15$. We take $\Delta t = \frac{1}{4}\Delta x$.

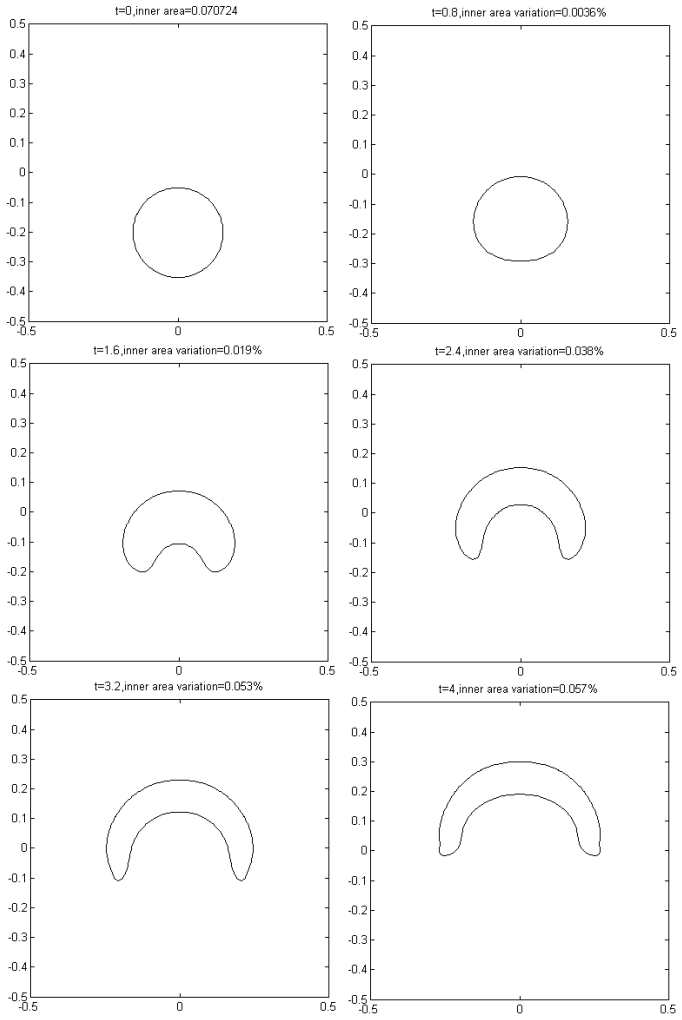

Figure 19: Simulation of example.1

Note that the inner area variation is less than 0.1% and if area variation is positive then it means area increases. On the other hand, if area variation is negative then area decreases. We contour $\phi = 0$ in this and later examples.

Example.2
In this example, we simulate the motion of two bubbles in two phases. We use two circles to represent interface of two bubbles. If the initial distance between two bubbles is far enough, then the two bubbles will not merge each other. Note that [8] had details about how far that two bubbles will not merge.

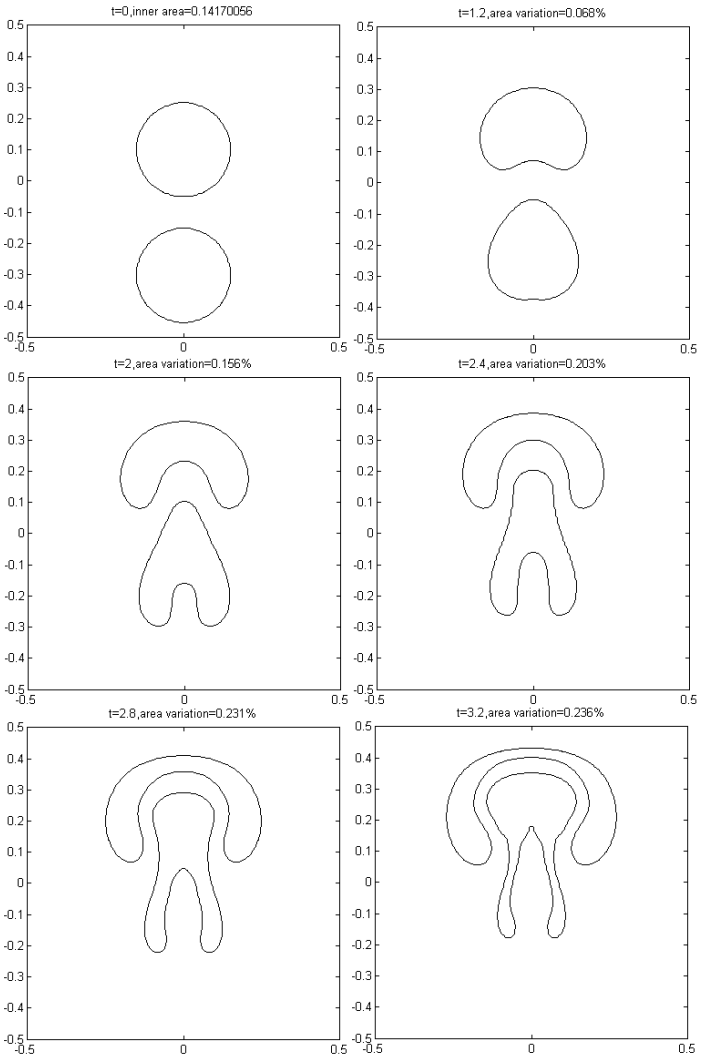

Figure 20: Simulation of example.2

We use level function to represent the interface of two circles and we first choose the case which do not merge

$$\phi_1(x,y) = (\sqrt{x^2+(y-0.1)^2} - 0.15) \times (\sqrt{x^2+(y+0.3)^2} - 0.15)$$

Note that ϕ_1 is not signed distance function and we need apply reinitialization process at the beginning.

Here we give our setting. Inside interface:$\rho_1 = 1$, $\mu_1 = 0.02$ and outside interface:$\rho_2 = 40$, $\mu_2 = 0.00025$, $\sigma = 0.005$ and $\Delta t = \frac{1}{4}\Delta x$. Note that since the deformation of bubble shape is much heavier and so the area variation increase more. But the area variation still less than 1% which is also a good result.

Example.3

In this example, our setting is just as example.1 except the level function and we let the distance between two bubbles to be smaller. So the level function is

$$\phi_2(x,y) = (\sqrt{x^2+y^2} - 0.15) \times (\sqrt{x^2+(y+0.3)^2} - 0.14)$$

We find that two bubbles merge with each other and then break up into two parts. This example tells us a very important information which is that this coupled VOF and level set method also can be applied to the changing of interface topology like merging and breaking and the inner volume variation is still less than 1%.

Example.4

In this example we apply steady shear flow to simulation one bubble in two-phase flow. The circle is centered at $(0,0)$ with radius 0.1 in the computational domain $[-0.5,0.5]\times[-0.5,0.5]$ and we set the boundary condition of velocity field as $\mathbf{u}_b = (2y, 0)$. We set the gravity $g = 0$, $\sigma = 0.1$, $\rho_1 = \rho_2 = 1$ and $\mu_1 = 0.1, \mu_2 = 1$.Note that the area variation is less than 10^{-3}% since we let the time step $\Delta t = \frac{1}{40}\Delta x$.

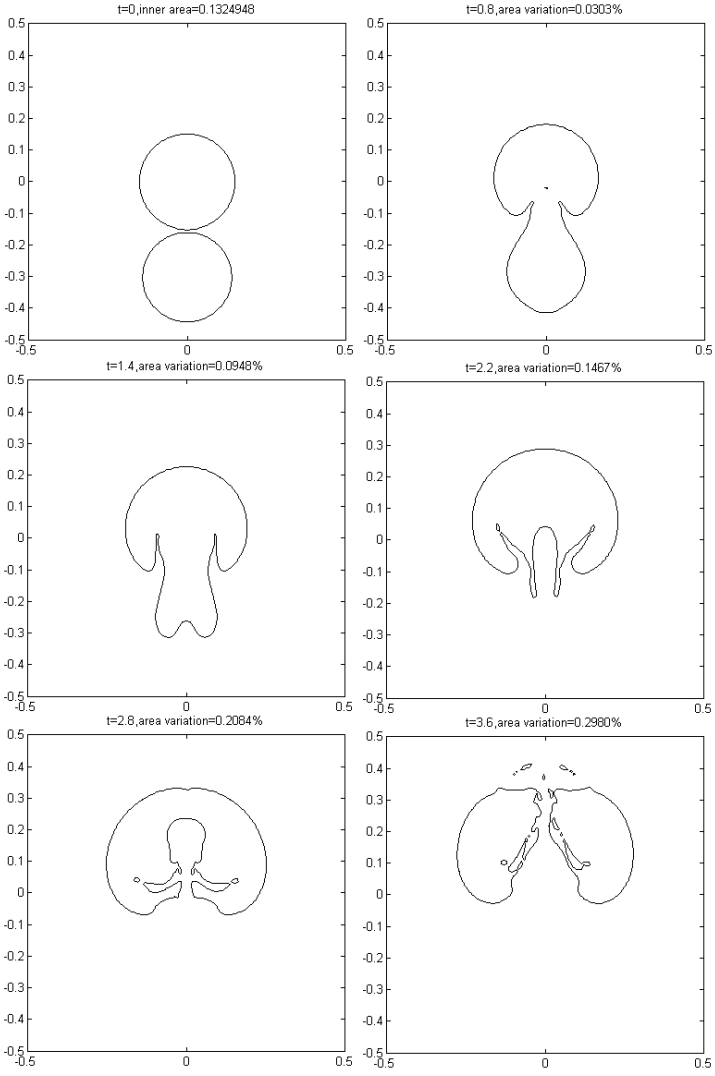

Figure 21: Simulation of example.3

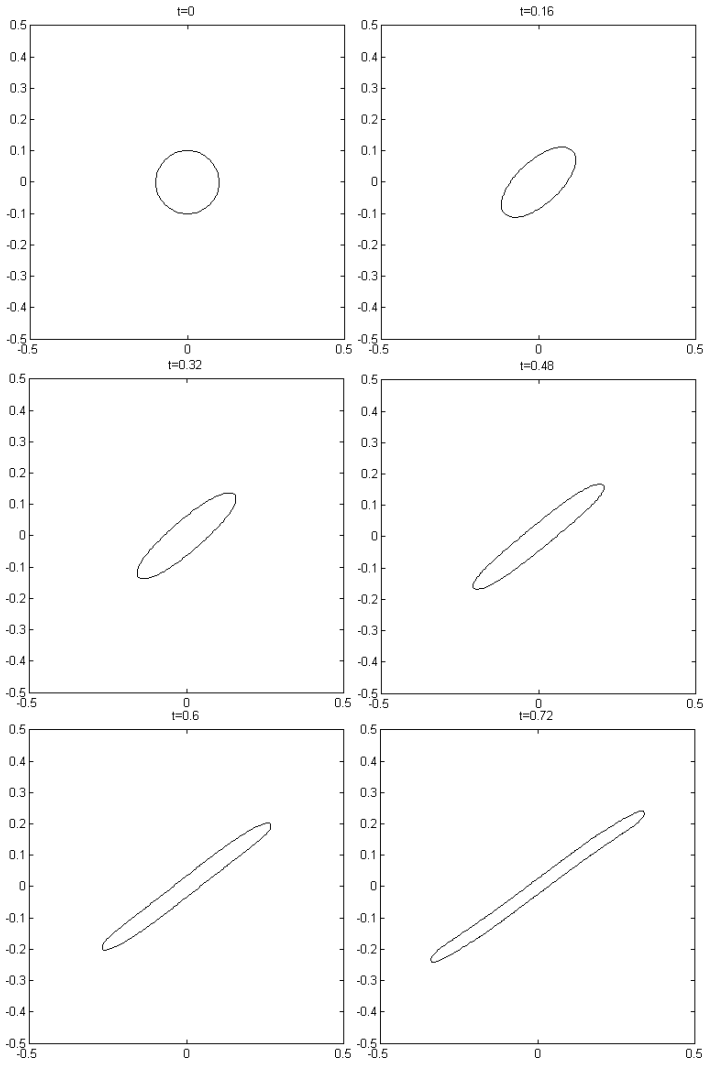

Figure 22: Simulation of example.4

5 Insoluble Surfactant on Interface

5.1 Introduction

Since we have introduced the VOF method and Local Level Set Method in above chapter, we now couple these two methods to solve insoluble surfactant on interface. Note that we have introduced how to solve heat equation on circle which is static(i.e. interface do not move) and in this chapter interface will be moved by velocity field. So the surfactant problem is more complicate than heat equation on circle since the interface is no more static.

We refer to [21] which uses level set method to handle the interface properties and solves surfactant equation by local level set method. We also couple VOF into level set method to conserve inner area of bubble. Since we use local level set method as [21], so we can expect that the mass of surfactant will not be conserved in our numerical scheme and this is the drawback of local level set method. However the local level set is still powerful since it just defined on Cartesian grid and the computational domain is just a band of interface.

5.2 Governing Equations with Insoluble Surfactant

We want to solve Navier-Stokes equation with two-phase flow and surfactant on the interface, so the Navier-Stokes equation becomes

$$\begin{aligned} \rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}) &= -\nabla p + \rho g - \nabla\cdot\mu[\nabla\mathbf{u} + (\nabla\mathbf{u})^T] + \sigma\kappa\mathbf{n} - \nabla_s\sigma \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned} \tag{61}$$

where $(\sigma\kappa\mathbf{n} - \nabla_s\sigma)$ is called Laplace-Young equation and since the effect of surface tension is just on the interface, we approximate $\sigma\kappa\mathbf{n}$ by $\sigma\kappa\nabla H$ where H is a heaviside function which is defined as above chapter. Note that $\sigma\kappa\mathbf{n}$ is the capillary force and $\nabla_s\sigma$ is Marangoni force where ∇_s is surface gradient which is $\nabla_s = (I - \mathbf{n}\otimes\mathbf{n})\nabla$ that we have mentioned above.

Since the surfactant will depress the surface tension coefficient σ, so the surfactant and σ have a relation.We refer [21] to write down the relation :

$$\sigma(\Gamma) = \sigma_0 - RT\Gamma \tag{62}$$

where Γ is surfactant concentration, σ_0 is surface tension coefficient for clean interface which means without surfactant, R is the ideal gas constant and T

is absolute temperature. Then we refer [22] to non-dimensionalize Eq.(62) as the form :

$$\sigma(\Gamma) = \sigma_0(1 - \beta\Gamma) \tag{63}$$

where β satisfies $0 \leq \beta \leq 1$.We can know that from Eq.(63), if surfactant concentration Γ is lager, then the surface tension coefficient σ is smaller.

Since the surfactant will diffuse and convect along the interface, so we can have the diffusion-convection equation of surfactant as follow:

$$\Gamma_t + \mathbf{u} \cdot \nabla\Gamma - \mathbf{n} \cdot (\nabla\mathbf{u}\mathbf{n})\Gamma = \frac{1}{Pe}\Delta_s\Gamma \tag{64}$$

where Pe is Peclet number.

Next we describe how to calculate the mass of surfactant. Since the surfactant is just defined on interface which we use ϕ to describe. The surface integral of Γ on the interface $\partial\Omega$ is

$$\int_{\partial\Omega} \Gamma(\mathbf{x})ds = \int \Gamma(\mathbf{x})\delta(\phi)|\nabla\phi|d\mathbf{x} \tag{65}$$

where δ is a $1D\delta$-function which we refer [18] as

$$\delta(x) = \begin{cases} 0, & \text{if } |x| > \omega, \\ -\frac{1}{6\omega}(1 + \cos(\frac{\pi x}{\omega})) & \\ +\frac{4}{3\omega}(1 + \cos(\frac{2\pi x}{\omega})), & \text{if } |x| < 0.5\omega \\ -\frac{1}{6\omega}(1 + \cos(\frac{\pi x}{\omega})). & \text{if } 0.5\omega \leq |x| < \omega \end{cases}$$

[18] chooses $\omega = 1.5\Delta x$ and it is easy to check that

$$\int_{\Omega} \delta(x)dx = 1$$

So the total mass of surfactant M is

$$M = \int_{\partial\Omega} \Gamma(\mathbf{x})ds = \int \Gamma(\mathbf{x})\delta(\phi)|\nabla\phi|d\mathbf{x}$$

5.3 Outline of Method

When solving Eq.(64) on moving interface relative to two-phase flow, the process is complicate since there are many equations needed to solve and we couple VOF and level set to handle the properties of interface, so we describe the outline of whole method for one time step as below :

Step1 Solve the Navier-Stokes equation to get the velocity $\mathbf{u}$ and note that our velocity $\mathbf{u}$ is defined on the cell face.(e.g.solve Eq.(61))

Step2 Solve the advection equation of VOF by finite volume method and reconstruct the interface by PLIC method.

Step3 Move the interface of level function by the velocity $\mathbf{u}$.

Step4 Reinitialize ϕ to be signed distance function within a band by the VOF method.

Step5 Extend the surfactant Γ off interface by normal extension method.

Step6 Evaluate surfactant equation within the band of interface,that is to solve Eq.(64) using local level set method.

Step7 Evolve Eq.(63) and solve $\nabla_s\sigma$. Note that we also define σ in a band of interface and solve $\nabla_s\sigma$ by Cartesian coordinate. Finally we return to Step1.

The process of whole method is as above and there are seven steps to handle, so the priority of solving which equation must be decided.

5.4 Algorithm of The Method

We have given the details from Step1 to Step6 in above chapters and we just write the algorithm of Step7 since the surfactant equation is a little different as comparing with heat equation.

The surfactant equation can be rewritten as

$$\Gamma_t + \mathbf{u}\cdot\nabla\Gamma - \mathbf{n}\cdot(\nabla\mathbf{u}\mathbf{n})\Gamma = \frac{1}{Pe}(\Delta\Gamma - \frac{\partial^2\Gamma}{\partial\mathbf{n}^2} - \kappa\frac{\partial\Gamma}{\partial\mathbf{n}}) \tag{66}$$

and again we use semi-implicit Crank-Nicholson method to solve Eq.(64) which we have mentioned above. Note that Crank-Nicholson is just used in diffusion part which is $\Delta_s\Gamma$ and we treat the convection part with explicit discretization, then the discretization form is :

$$\begin{aligned}\frac{\Gamma^{n+1}-\Gamma^n}{\Delta t} =& \frac{1}{Pe}\frac{\Delta\Gamma^{n+1}+\Delta\Gamma^n}{2} + \frac{3}{2}[-\frac{1}{Pe}(\kappa\frac{\partial\Gamma}{\partial\mathbf{n}} + \frac{\partial^2\Gamma}{\partial\mathbf{n}^2}) - \mathbf{u}\cdot\nabla\Gamma + \mathbf{n}\cdot(\nabla\mathbf{u}\mathbf{n})\Gamma]^n \\ & - \frac{1}{2}[-\frac{1}{Pe}(\kappa\frac{\partial\Gamma}{\partial\mathbf{n}} + \frac{\partial^2\Gamma}{\partial\mathbf{n}^2}) - \mathbf{u}\cdot\nabla\Gamma + \mathbf{n}\cdot(\nabla\mathbf{u}\mathbf{n})\Gamma]^{n-1}\end{aligned} \tag{67}$$

We refer [21] to write the form of Eq.(66) and [21] suggested that in the discretization for spatial of Eq.(66), central difference is used in all term except this term $\mathbf{u}\cdot\nabla\Gamma$ which is discretized by upwind-WENO-3 scheme which we have introduced. Note that we encounter the data structure problem again as in above chapter.

Next we handle $\nabla_s\sigma$, which we can rewrite as

$$\begin{aligned}\nabla_s\sigma &= (I-\mathbf{n}\otimes\mathbf{n})\nabla\sigma \\ &= \begin{bmatrix} n_y^2 & -n_xn_y \\ -n_xn_y & n_x^2 \end{bmatrix}\begin{bmatrix} \sigma_x \\ \sigma_y \end{bmatrix} \\ &= \begin{bmatrix} \sigma_x n_y^2 - \sigma_y n_x n_y \\ \sigma_y n_x^2 - \sigma_x n_x n_y \end{bmatrix}\end{aligned}$$

Note that σ_x and σ_y are discretized by central difference.

5.5 Numerical Results

The effect of surfactant is to decrease the surface tension coefficient and so the interface will deform much heavier than clean interface.

In this section we refer the example of [22] and we apply the steady shear flow with boundary condition $\mathbf{u}_b = (2y, 0)$ in the computational domain $[-0.5, 0.5] \times [-0.5, 0.5]$.The initial bubble is centered at $(0, 0)$ with radius $r = 0.15$.

Example.1

We give our setting as follows.Initial surface tension coefficient $\sigma_0 = 0.5$, gravity $g = 0$, $\rho_1 = \rho_2 = 1$, $\mu_1 = 0.1$, $\mu_2 = 1$ and Peclet number $Pe = 1$. Note that our time step is as the example of above chapter, $\Delta t = \frac{1}{40}\Delta x$ and we consider three cases that are $\beta = 0$(clean), $\beta = 0.25$ and $\beta = 0.5$(See figure23).

As our expectation, if β is larger then the interface will deforms heavier since we know that

$$\sigma(\Gamma) = \sigma_0(1 - \beta\Gamma)$$

which means that concentration affects the surface tension coefficient.

Note that our inner area is conserved within 0.01% but the surfactant concentration which computed by Eq.(64) will not be conserved as we predict before.(See figure24)

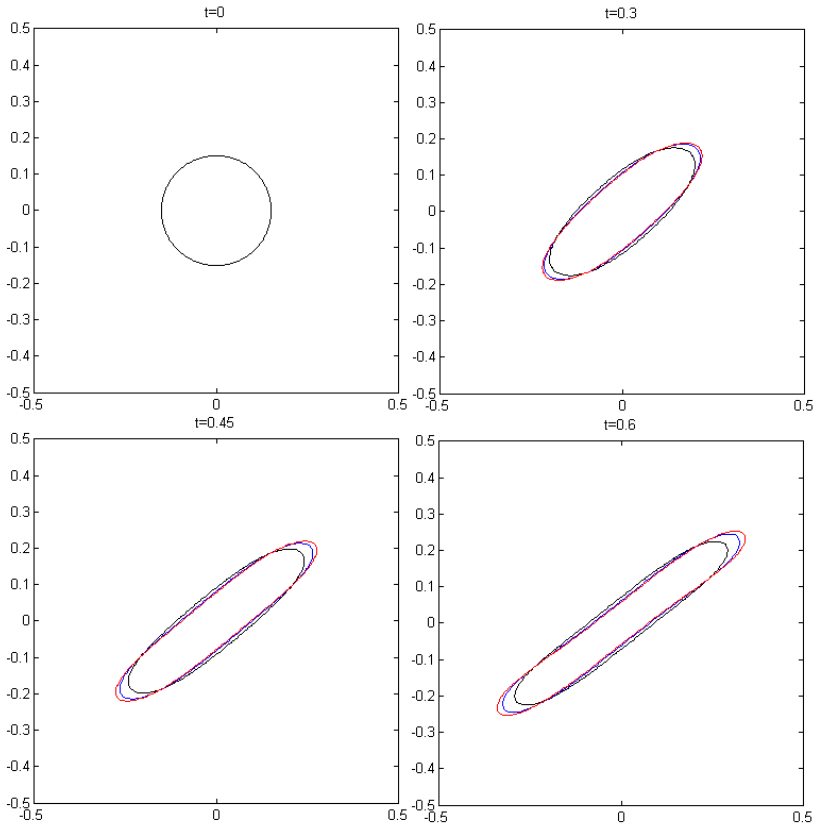

Figure 23: Time evolution of a bubble in shear flow with $\beta = 0$(black), $\beta = 0.25$(blue) and $\beta = 0.5$(red)

6 Conclusion and Future Work

In this article we introduce some methods to simulate two-phase flow with or without surfactant. We also introduce level set method to solve heat equation on a circle and we can extend the level set method to $3D$ directly; that is to solve heat equation on sphere $\phi(x, y, z) = \sqrt{x^2 + y^2 + z^2} - 1$. We can also couple level set method and IIM(immerse interface method) to solve heat equation on the $2D$ domain with jump condition on the interface. The application of level set is very widely and it is really a valuable method.

The VOF method used in this article is to conserve the inner area and we get nice result in our numerical simulation. By coupling VOF and level set method, we are not only able to reconstruct the interface accurately but also to conserve the volume of inner area. Moreover, the method is also adapted to changing of interface topology such as two bubbles merging or one bubble braking into two parts.

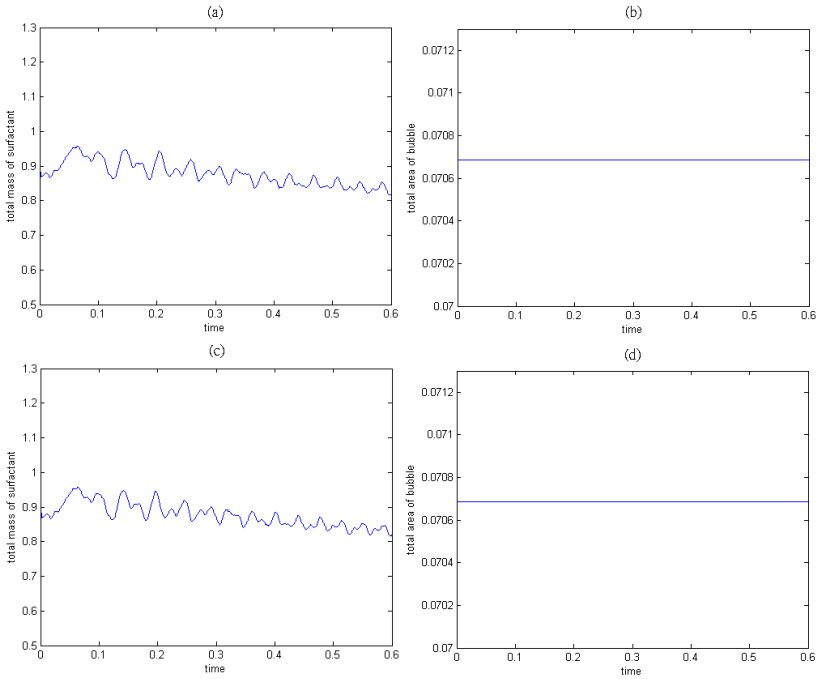

Figure 24: (a) Total surfactant in each time step for $\beta = 0.25$. (b) Total area of drop in each time step for $\beta = 0.25$. (c) Total surfactant in each time step for $\beta = 0.5$. (d) Total area of drop in each time step for $\beta = 0.5$.

In our simulation of two-phase flow with surfactant on interfaces, the inner area is still conserved as we expect but the mass of surfactant is not conserved which does not make sense in our real life. So our future work is to develop another method to let the mass of surfactant to be conserved, that is we use other method to handle Eq.(64) and get feasible result in mass of surfactant.

References

[1] S. Osher, R. Fedkiw (2003). Level Set Methods and Dynamic Implicit Surfaces, Springer

[2] J. A. Sethian (1996). Level Set Methods and Fast Marching Methods, Cambrige

[3] C. W. Hirt, B. D. Nichols (1981). Volume of Fluid Method for the Dynamics of Free Boundaries, J. Comput. Phys 39, 201-225

[4] C.-W. Shu (1989). Efficient Implementation of Essentially Non-oscillatory Shoch-Capturing Schemes II, J. Comput. Phys 83, 32-78

[5] S. Osher, C.-W. Shu (1991). High-Order Essentially Nonoscillatory Schemes for Hamilton-Jacobi Equations, SIAM J. on Num. Analysis, Vol.28,No.4 907-922

[6] M. Sussman, P. Smereka, S. Osher (1994). A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow, J. Comput. Phys 114, 146-159

[7] H. K. Zhao, T. Chan, B. Merriman, S. Osher (1996). A Variation Level Set Approach to Multiphase Motion, J. Comput. Phys 127, 179-195

[8] L. Chen, S. V. Garimella, J. A. Reizes, E. Leonardi (1997). Motion of Interacting Gas Bubbles in A Viscous Liquid Including Wall Effects and Evaporation, Numer. Heat Trans,Part A, 31, 629-654

[9] S. Chen, B. Merriman, S. Osher, P. Smereka (1997). A Simple Level Set Method for Solving Stefan Problems, J. Comput. Phys 135, 8-29

[10] R. Scardovelli,S. Zaleski (1999). Direct Numerical Simulation of Free-Surface and Interfacial Flow, Annu. Rev. Fluid Mech. 31, 567-603

[11] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski (1998). Volume-of-Fluid Interface Tracking with Smoothed Surface Stress Methods for Three-Dimensional Flows, J. Comput. Phys 152, 423-456

[12] D. Peng, B. Merriman, S. Osher, H. K. Zhao, M. Kang (1999). A PDE-Based Fast Local Level Set Method, J. Comput. Phys 155, 410-438

[13] G.-S. Jiang, D. Peng (2000). Weighted ENO Schemes for Hamilton-Jacobi Equations, SIAM J. Sci. Comput. Vol.21, No.6 2126-2143

[14] M. Sussman, E. G. Puckett (2000) A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows, J. Comput. Phys 162, 301-337

[15] M. Beralmio, L.-T. Cheng, S. Osher (2001) Variational Problems and Partial Differential Equations on Implicit Surfaces, J. Comput. Phys 174, 759-780

[16] G. Son (2001) A Numerical Method for Bubble Motion with Phase Change, Numer. Heat Trans,Part B, 39, 509-523

[17] G. Son, N. Hur (2002) A Coupled Level Set and Volume-of-Fluid Method for the Buoyancy-Driven Motion of Fluid Particles, Numer. Heat Trans,Part B, 42, 523-542

[18] J.-J. Xu, H.-K. Zhao (2002) An Eulerian Formulation for Solving Partial Differential Equations Along A Moving Interface, J. of Sci. Comp. Vol.19, 573-594

[19] J. B. Greer (2006) An Improvement of A Recent Eulerian Method for Solving PDEs on General Geometries, J. of Sci. Comp. Vol.29, 321-352

[20] H. Tseng (2006) Numerical methods and applications for immersed interface problems, Master Thesis.

[21] J.-J. Xu, Z. Li, J. Lowengrub, H.-K. Zhao (2006) A level-set method for interfacial flows with surfactant, J. Comput. Phys 212,590-616

[22] M.-C. Lai, Y.-H. Tseng, H. Huang , An immersed boundary method for interfacial flows with insoluble surfactant, J.Comput.Phys., in press 2008