

ABSTRACT

Title of dissertation: INTERFACE ADVECTION
AND JUMP CONDITION
CAPTURING METHODS
FOR MULTIPHASE INCOMPRESSIBLE FLOW

Zhipeng Qin, Doctor of Philosophy, 2014

Proposal directed by: Professor Amir Riaz
Department of Mechanical Engineering

In this work, new numerical methods are proposed to efficiently resolve interfaces occurring in multiphase incompressible flows. Multiphase flow problems consist of a large class of physical phenomenon from bubbles to bow waves in ships. Over the recent decades, numerical methods are becoming an important tool in addition to pure analytical and experimental methods. However, there is still large room for improvement in existing numerical methods.

Contributions are made in the field of interface advection and the jump conditions for pressure. In the case of advection, a method is developed specifically for implicit interfaces that evolve with the Eulerian advection of a scalar field. The new method is validated by comparison with the interfaces that evolve with Lagrangian advection using a connected set of marker particles.

To accurately capture the jump conditions, a second order accurate method is proposed for solving the variable coefficient Poisson's equation in the discretized Navier-Stokes formulation. This new method assumes both phases exist in the

interface cell and that their collective effect can be expressed by a volume fraction weighted average value.

The new capabilities have been integrated to build a dynamic Navier-Stokes equation solver. The new advection scheme is also associated to track the interface. The new solver is tested by applications in several two phase flow problems.

INTERFACE ADVECTION AND JUMP CONDITION
CAPTURING METHODS FOR MULTIPHASE
INCOMPRESSIBLE FLOW

by

Zhipeng Qin

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:

Dr. Amir Riaz, Chair/Advisor

Dr. Peter S. Bernard

Dr. James D. Baeder

Dr. Jelena Srebric,

Dr. Ricardo H. Nochetto, Dean's Representative

© Copyright by
Zhipeng Qin
2015

Dedication

I dedicate my dissertation work to my family and many friends. A special gratitude is to my loving parents, parents in law, and my wife.

Acknowledgments

I would like to express the deepest appreciation to my advisor, Professor Amir Riaz, who introduced me to Computational Fluid Dynamics area. In my four years PH. D. program, he has been a tremendous mentor for me. His attitude and the substance of a genius guide me to grow as a research scientist. Professor Amir Riaz continually and persuasively conveyed a spirit of adventure in regard to research and scholarship. With his priceless encouragement and advisement, I feel confident to face challenge from research work as well as life.

I would also like to thank my committee members, Professor Peter S. Bernard, Professor James D. Baeder, Professor Jelena Srebric and Professor Ricardo H. Nochetto, for serving as my committee members even at hardship. I also want to thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you. A thank you is also to Professor Elias Balaras, who extended my numerical algorithm to three spatial dimension and collaborated in all my three publications.

In addition, I would like to specially thank my beloved wife, Xiao Yu, for her love, kindness and support she has shown since we first met with each other nine years ago. Without her endless love and understanding, I can not realize my dream and finish this dissertation. Last but not least, I would like to thank my parents and all of my friends who incented me to strive towards my goal.

Table of Contents

List of Figures	vi
List of Tables	viii
1 Introduction	1
1.1 Motivation	1
1.2 Advection of the interface	2
1.3 Boundary Capturing Method	9
1.4 Goals	15
1.5 Contribution	15
2 Topology Preserving Advection of Implicit Interfaces on Cartesian Grids	17
2.1 Overview	17
2.2 Geometric projection of interface topology	21
2.2.1 Projection of particle based interface	28
2.2.2 Projection of implicit interface	34
2.3 Two phase incompressible flow	47
2.3.1 Numerical method	47
2.3.2 Validation	49
2.4 Conclusion	57
3 Second Order Accurate Boundary Capture Method	59
3.1 Overview	59
3.2 Two-phase Poisson equation	60
3.3 Numerical Method	61
3.3.1 One Dimension	61
3.3.2 Two Dimensions	65
3.3.3 Three Dimensions	70
3.4 Numerical results	72
3.4.1 1-D examples	72
3.4.2 2-D examples	74
3.4.3 3-D examples	78
3.5 Conclusion	78

4	High Accurate Solution of Two Phase Flow Problems	81
4.1	Overview	81
4.2	Governing Equation	82
4.3	Proposed Numerical Method	83
4.3.1	Advection Scheme	83
4.3.2	Projection Method	84
4.3.2.1	Projection Method	84
4.3.2.2	Boundary condition	85
4.4	Poisson's Equation	85
4.4.1	Convection Terms	86
4.4.2	Viscous Terms	86
4.4.3	Jump condition capturing method	87
4.4.3.1	Conservative form of jump condition	88
4.4.3.2	Value of α	90
4.5	Temporal Updating Method	92
4.6	Numerical Solution	92
4.6.1	Steady cycle drop	92
4.6.2	Oscillation drop	92
4.6.3	Rising bubble	94
4.7	Conclusion	96
A	Geometric projection in 3D	101

List of Figures

2.1	Sketch illustrating the principle of geometric projection	20
2.2	Implementation of the stepwise refinement of geometric projection . .	22
2.3	Typical arrangement of four interfacial points	24
2.4	Contours of $\phi(\mathbf{x}, t)$, obtained with geometric projection	31
2.5	Normal angle distribution and its convergence	32
2.6	Contours with different schemes on a 65×65 grid	36
2.7	Contours with different schemes on a 129×129 grid	37
2.8	Contours with different schemes on a 129×129 grid for longer time .	39
2.9	Convergence analysis for different advection methods	40
2.10	Interface profiles on coarse and fine grids after one full period of rotation for $T = 2$	44
2.11	Pressure and velocity solutions with exact curvature	49
2.12	Oscillation of the interface with geometric projection	50
2.13	Rayleigh-Taylor problem with different Re number	52
2.14	Accuracy of numerical simulation of Rayleigh-Taylor instability . . .	53
2.15	The evolution of Rayleigh-Taylor instability	54
2.16	Rising bubble contours at different time	55
3.1	Jump condition in one dimension. Volume fraction ϕ_1 and ϕ_2 are used to weight discontinuous variable P	62
3.2	The Area fraction method used to obtain volume fraction for Two- Dimension case. A is the area inside of the interface and $\phi_{1,i,j} = \frac{A}{dxdy}$. .	65
3.3	(left) Exact and computed solutions. P_1 (green diamond), P_2 (blue cross), and \bar{P} (red cycle). (right) Error vs. grid spacing h . L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line) . .	69
3.4	(left) Exact and computed solutions. P_1 (green diamond), P_2 (blue cross), and \bar{P} (red cycle). (right) Error vs. grid spacing h . L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line) . .	71
3.5	(left) Numerical solution for the 2-D case specified by Equations (3.75) for $[P] \neq 0$, $[\lambda P_n] \neq 0$. (right) Error vs. grid spacing h . Colors indicate, methods GFM (green), CSF (blue) and 2nd order scheme (red). L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line).	73

3.6	(left) Numerical solution for two spatial dimensional case, $[P] = 0$, $[\lambda P_n] \neq 0$. (right) Error vs. grid spacing h . Colors indicate, methods GFM (green), CSF (blue) and 2nd order scheme (red). Compared with GFM and CSF, the slope of new 2nd order scheme is indicated as $O(h^2)$ or larger for L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line) of the error.	74
3.7	(left) Numerical solution for two spatial dimensional case, $[P] \neq 0$, $[\lambda P_n] \neq 0$. (right) Error vs. grid spacing h . Colors indicate, methods GFM (green), CSF (blue) and 2nd order scheme (red). Compared with 1st order scheme and CSF method, the slope of new 2nd order scheme is indicated as $O(h^2)$ or larger for L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line) of the error.	76
3.8	(a) Numerical solution for two spatial dimensional case, $[P] \neq 0$, $[\lambda P_n] \neq 0$. (b) Error vs. grid spacing h . Colors indicate, methods GFM (green), CSF (blue) and 2nd order scheme (red). Compared with 1st order scheme and CSF method, the slope of new 2nd order scheme is indicated as $O(h^2)$ or larger for L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line) of the error. (c) L_∞ norm of the error of new jump condition capturing method (red) and "immersed interface" method (pink). The value of new jump condition capturing method is five times lower.	77
3.9	(left) $P(x, y, z = 0.4)$ cross section of numerical solution for three spatial dimensional case, $[P] \neq 0$, $[\lambda P_n] \neq 0$. (right) Error vs. grid spacing h . Colors indicate, methods GFM (green), CSF (blue) and 2nd order scheme (red). Compared with 1st order scheme and CSF method, the slope of new 2nd order scheme is indicated as more than $O(h^3)$ or larger for L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line) of the error.	79
4.1	(a) Computational domain separated into four quadrants by the sign of \hat{n} . (b) "upwind" method builds conservative jump condition for each quadrants, e.g., the jump condition form is Equation (4.34) in the yellow domain, first quadrant.	87
4.2	Implementation of the α . (a) Localization of grids where the interface passes through. Curvature is calculated by second order discretization on the cell centers (black symbols). The value of α of the grids is the curvature of the interface segments (red symbols) corresponding to the cell centers. The level set function is negative on the left and positive on the right hand side. (b) A cell center x and its corresponding interface segment, x_1 . The character ratio at x_1 is $r_x + \Delta r$, then $r_x - \phi$. (c) Same localization of grids as (a) with opposite level set function. (d) With the new condition, the character ratio at x_1 is $r_x - \Delta r$, then $r_x - \phi$	91

4.3	(a) Error of pressure (E_P) and (b) error of velocity ($E_{\mathbf{u}}$) for steady drop case and with density and viscosity ratio of 10^2 and $We = 1$. Solution by CSF method, 1st order scheme and our 2nd order scheme are indicated by red line, green line and blue line.	93
4.4	(a) Amplitude oscillation of the interface and (b) the kinetic energy for density and viscosity ratio of Inf , $We = 1$ and $Re = 100$. Solid blue line in (right) indicates the analytical solution.	95
4.5	Interface profiles for the rising bubble problem on a 200×200 grid for $Re = 100$, $We = 200$ and $Fr = 1$. at (a) $t = 3.0$; (b) $t = 4.2$; (c) $t = 5.4$; (d) $t = 6.0$	97
4.6	Accuracy of numerical simulation of rising bubble for $Re = 100$ and $We = 200$. (a) Volume conservation error as a function of time on a 200×200 grid, (b) error on differet sized gids at $t = 4$,(c) error on differet sized gids at $t = 5$,(d) error on differet sized gids at $t = 6$. Lines indicate new jump condition capturing method (red) and the 1st order one (green). The error is smaller in the case of our new method and decays with 1.5 nd order accuracy even after bubble broken.	98
4.7	Interface profiles for the rising bubble problem on a 200×200 grid for $Re = 100$, $We = 2$ and $Fr = 1$. at (a) $t = 3.0$; (b) $t = 4.8$; (c) $t = 5.4$; (d) $t = 6.0$	99
4.8	Accuracy of numerical simulation of rising bubble for $Re = 100$ and $We = 2$. (a) Volume conservation error as a function of time on a 200×200 grid, (b) error on differet sized gids at $t = 6$	100
A.1	(a) P_1 projection in 3-D. The distance \mathbf{d}_1 is the shortest distance from the grid point \mathbf{x} to the interface marker \mathbf{x}_p . (b) P_2 projection. The scalar field resides on cell vertices.	102
A.2	P_1 and P_2 projections for a different interface configuration. The interface marker \mathbf{x}_p is always connected to four panels.	102
A.3	Four possible configurations of interface panels based on 3,4,5 and 6 interface markers within a cell.	103
A.4	P_3 projection. (a) Interface markers connected to P_2 panel. (b) Projection to the smooth interface.	103

List of Tables

2.1	Number of particles needed for high accuracy	27
-----	--	----

2.2	Topology convergence by geometric projection based on marked particle	30
2.3	Topology convergence by geometric projection based on implicit interface	33
2.4	Topologic convergence of stepwise projection scheme	45
2.5	Run time of different advection schemes	46
2.6	Error in the velocity field measured with the L^2 norm on various grids	50
4.1	Numerical period by 1st order and 2nd order jump condition capturing method and the error from the analytical solution. Our new jump condition method can obtain almost exact solution on high resolution	95

Chapter 1: Introduction

1.1 Motivation

Multiphase flow problems involve numerous phenomenon, such as drops, bubbles, solid particles, capillary waves, porous media flows. These phenomenon determine the behaviors of chemical reactors, energy production systems, oil extraction process, and the global climate at various temporal and spatial scales. Traditionally, research problems related to these areas were investigated by pure analytical methods or laboratory experiments. However, the complexity of the multiphase flows problems resulted in limited understanding by traditional methods. Consequently, numerical simulation is becoming an essential tool to for studying the multiphase flow phenomenon and is poised to have a major impact in industry and research. [5]

For numerical simulation of multiphase flows, there are two fundamental aspects that determine overall effectiveness of the solvers. These are the advection of the interface and the implementation of jump conditions across the interface. These two aspects are necessary to obtain a numerical solution of the Navier-Stokes equations in each phase. Several methods have been developed for each of the two aspects. For example, the volume-of-fluid method (VOF), front tracking scheme

(FTS), and the level set method, exist for the purpose of advecting the interface. In addition, Ghost Fluid Method and the "immersed interface" method are developed to implement the jump conditions for multiphase incompressible flow.

While these existing numerical methods provided reasonable results they are limited by issues such as the lack of mass conservation and the distortion of interface topology during advection. Moreover, existing implementation of interfacial jump conditions are only 1st order accurate. Hence, there is a clear need for improvement. In order to facilitate the development of improved methods, a standard evaluation system, containing essential features such as the volume conservation, topology maintenance, computational cost and robustness is also needed.

1.2 Advection of the interface

The first successful simulation for the motion of the fluid interface was demonstrated with the Marker-and-cell (MAC) method, first proposed by Harlow and Welch [27] and developed by Daly [17, 18], which used the marker particles to define different fluids on uniform structured grids. Following this pioneering approach, numerous methods have emerged for improving the quality of interface advection. Among these, volume-of-fluid method (VOF), front tracking scheme (FTS) and level set methods are the most popular. [5]

These methods use a scalar field based indicator function to represent the interface separating the different phases. The advection of the interface is realized by updating the value of the indicator function. VOF focuses on finding the fluxes

into and out of the numerical cells and generally consists of three computation steps, reconstruction of the interface, fluxes calculation and the marker function update. In these three steps, reconstruction of the interface is the basic issue. In the past years, many reconstruction techniques were developed.

The earliest approach used to reconstruct the interface was a simple line interface calculation (SLIC) scheme, which was proposed by W. E. Noh et al. [76], following the idea of local surface approximation. With the help of SLIC, straight lines, either perpendicular or parallel to each coordinate direction, were used to compose a one dimensional component, the composition of which defined the fluid surface. Similar to the idea in Debar's paper[20], a local surface approximation was defined for each mixed-fluid zone. At every time step, various fluids were tested in each one-dimensional coordinate direction regarding whether they were present or absent in just three cells, i.e., the mixed cell, the right one and the left one. Based on trial and error, the surface approximation was determined for the center mixed cells. Because each mixed-fluid zone was decomposed into several one-dimensional components, the SLIC was able to advance the fluid surfaces in time.

In a manner similar to the SLIC method, Hirt and Nichols (H-N) [9] approximated the interface by a straight line that divided the cell in two parts. However, the H-N method defined the interface approximation by target directions. Single valued functions, e.g., $Y(x)$ and $X(y)$, were built to determine the shape of the interface and the position of fluid. The comparison between the derivatives of $Y(x)$ and $X(y)$ was used to determine whether the interface should be horizontal or vertical. On the other hand, the sign of the single valued functions was used to decide which side

was occupied by which fluid and neighboring cell were chosen as the interpolation neighbors for surface cell.

Considering the accuracy, there was no substantial difference between SLIC and H-N schemes. Piecewise-linear interface calculation (PLIC) scheme was subsequently developed for improving the reconstruction. Rudman [65] used a combination proposed by Zalesak [80] to eliminate both the diffusiveness and instability, which were two main challenges for previous reconstruction techniques, and improve the accuracy of PLIC. This method involved the following steps. An intermediate value of a marker function was determined by a monotonic advection scheme at first. Then the possible numerical diffusion was corrected by an anti-diffusive flux. In order to ensure no new extrema were introduced into this calculation, a correction factor, whose value was decided by the marker function and its intermediate value in under consideration cell and its neighbors, was defined. Finally, both the anti-diffusive flux and correction factor were used to obtain the value of marker function in new time step.

VOF were popular and performed successfully for relatively simple cases and smooth interface flows. But, for the more complex cases, FTS (Front Tracking Scheme) provided high accuracy because the interface itself was described by additional computational elements. Instead of updating the value of the marker function, FTS tracked the interface by moving marker particles. Earlier, the direct FTS method was used with the immersed interface is represented by connected marker points. W. F. Noh [49] proposed coupled-Eulerian-Lagrangian method. It coupled particle tracking and the surface approximation together by lagrangian polygonal

lines. Similarly, the particle-in-cell (PIC) method was developed by A. A. Amsden [4]. By PIC method, Mass particles were used to tag and keep track of various fluids. In order to capture more details, Richtmyer et al. [62] and Glimm et al. [37] used irregular grid and finite difference stencil in the vicinity of the interface, which was also formed by a connected set of particles.

A somewhat different idea was developed by Peskin [53, 54]. The connected marker points were also used to represent the moving immersed interface, which moved at the local fluid velocity and exerted forces locally on the fluid, either imposed externally or adjusted in the same direction of the expected velocity. However, compared with previous FTS, Peskin's scheme used the same grid in the whole domain. For the cells around interface, the force was distributed on the fixed grid.

The major drawback of direct FTS was complexity. In order to simplify the tracking process, Unverdi et al. [75] proposed a coupling method. It combined some features from direct FTS and VOF but without reconstructing the interface. It derived from Peskin's method and a well-known vortex-in-cell (VIC) method reformulating the governing equation as an integral equation over the interface. Unlike the traditional FTS method that treated each phase separately, Tryggvason and collaborators (see e.g. [28, 32, 74]) treated all phases together by solving a single set of Navier-Stokes equation in the whole domain. Therefore, it was able to represent interfaccial interaction in a more natural way, which was the toughest challenge for FTS.

Compared with VOF and FTS, level set method, introduced by Osher et al. [67] and Shu et al. [10], is significantly simpler to implement, particularly for 3-D

application. Because the level set method acts indirectly on the implicit interface associated with the field, it leads to a smoother transition of both incompressible and compressible flow [77] across the interface. This provides accurate topological quantities, such as curvature and normals. The level set method has therefore become the main alternative to VOF and FTS for propagating sharp interfaces on Cartesian grids. [5]

In order to maintain the accuracy of the level set method, the scalar field is needed to be reinitialized to a signed distance function and the reinitialization is needed to be carried out frequently during the advection. Classical reinitialization method (CR) was the oldest method and also most well-known numerical scheme. It was proposed first by E. Rouy [26], where a region Ω^+ was given with the level set function $\phi \geq 0$ on Ω^+ and $\phi = 0$ on $\partial\Omega^+$. At the reinitialization step, it evolved an equation $\phi_t = 1 - |\nabla\phi|$ until ϕ became close enough to a distance function to reach a steady state. Sussman et al. [73] pointed out E. Rouy's method to still be explicit scheme, because its interface condition was prescribed on $\partial\Omega^+$. To provide a pure implicit scheme, they replaced the reinitialization equation by $\phi_t = S(\phi_0)(1 - \sqrt{\phi_x^2 + \phi_y^2})$, where $\phi_0(x) = \phi(x, 0)$ and the zero level set of ϕ_0 was set as air-liquid interface. At every computational time step, ϕ was reconstructed to have the same zero level set as ϕ_0 by solving the redistance equation. By Sussman's CR method, one iteration per time step was usually enough to converge ϕ to be distance function.

Sussman and his collaborators continued to improve CR method and applied it for many 2-D and 3-D two-phase cases. In [72], a "constraint" was implemented

along with the higher order differences schemes. At the same time, the effect from the thickness of interface for the numerical results had been noticed in two-phase flow cases. Sussman et al. [47] gave the interface a time independent width of only a few grid points wide to treat steep density ratio (1000/1) cases. Sussman et al. [71] extended his original CR method to more two phase incompressible flow cases by coupling it to the adaptive projection method. CR method was also used to solve 3D and axisymmetric incompressible Two-Phase flows. However, its drawback in mass conservation was amplified in more complex cases. In order to solve this issue, some coupled methods had been developed. In [70] and [30], combination schemes of level set method and either VOF or FTS provided superior results to the classical redistance iteration scheme alone.

Although the conservation of volume improved with the use of such techniques, the accuracy of representing interface topology on the Cartesian grid had not been investigated. The implicit interface could also be defined by an indicator function that transitioned smoothly from one fluid to the other over a transition region of finite width [19, 50]. Here too, the numerical distortion of the indicator function during advection perturbed the interface. A reinitialization procedure was applied to correct the deviation by subjecting the indicator function to "recompression" in directions normal to the interface [51]. With this approach, Olsson et al. [51] and Sheu [68, 69] reported a significant improvement in volume conservation compared with reinitialization based on the Hamilton-Jacobi formulations.

However, similar to the development in other areas, every new method may pose new potential problems. The existing level set methods led to a degree of

improvement, but it was far from enough. For example, Desjardins et al.[23] noted that, when the "recompression method", proposed by [50, 51], was utilized, errors related the spurious drift and deformation of the interface would still pose a problem.

To facilitate the decoupling of advection and reinitialization, we propose a general approach for constructing a signed distance function by minimum distance calculation. The implicit interface would not be distorted by this scheme and the accurate topological quantities can be obtained. Moreover, the volume conservation can be maintained because the reinitialization step is needed only for advection to provide the necessary interface topology.

The principle of forming the signed distance function on the basis of the minimum distance between the interface and the grid has been employed in the past [19]. Adalsteinsson et al. [2] considered the minimum distance from the grid to a set of interface markers for initializing the fast marching procedure. Russo et al. [66] used this approach to obtain a reference solution to gauge the accuracy of their reinitialization. More recently, Desjardins [24] define the minimum distance with respect to piecewise linear interface segments to initialize the fast marching method [2]. A 1st order convergence of the curvature has been reported by Desjardins [24] as a result. The authors however do not report on the accuracy of the signed distance function and the normals.

In our publication [57], we provided a comprehensive report about the accuracy of geometric projection method and also compared it with classic reinitialization scheme and recompression method. From the report, our stepwise geometric projection method not only reduced the absolute error value, but also increased the

convergence rate significantly for topological properties, e.g., curvature and normal.

1.3 Boundary Capturing Method

In addition to the advection of the interface, an accurate solver for Navier-Stokes equation is also a key requirement for multiphase flow computation. In the past 50 years, many techniques have been developed. Early literature about numerical solution of complicated problems in fluid dynamics has been reviewed by Balder [3]. The early methods focused on single phase incompressible flow, such as the study of shear layers [15], and attempted to develop the discrete form of the incompressible constraint. Harlow and Welch [27] proposed a technique, for the numerical investigation of the time-dependent flow of incompressible fluids, enforcing the incompressible constraint by deriving a Poisson's equation for the pressure. It coupled FTS to Navier-Stokes equation solver, but failed to define the physical boundary condition for pressure. This drawback affected the accuracy of simulation. Krywicki and Ladyzhenskaya [43] tried to improve Harlow's work to avoid of artificial boundary condition for pressure. And other types of boundary condition for pressure was also imposed, in [11] and [1].

A well-known "projection method" was developed in a series of Chorin's papers. It is still widely used as an efficient means of solving incompressible Navier-Stocks equation. At first, Chorin [12] concentrated on the search for steady solution of the Navier-Stocks equation and introduce an artificial compressibility which is the principle of his method. After that, he [13] continued to apply this method for time-

dependent problems. It rewrote the dimensionless governing equation decomposed it into the sum of a vector with zero divergence and a vector with zero curl. The component with zero divergence was used to obtain velocity at the next time step and the component with zero curl was used to update pressure. This decomposition had also been extensively used in existence and uniqueness proofs for the solution of the Navier-Stokes equation in [31]. This method could be seen to project the intermediate vector field onto divergence-free field to recover the velocity, therefore, it was called as "projection method".

Then Chorin also established a convergence analysis for his projection method in [14]. A proof in both the maximum and L2 norms, with a suitable error estimate, has been obtained for special problems with periodicity boundary conditions. It showed that the projection method could reach first-order accuracy in time and second-order accuracy in space.

Chorin's method provided a successful model for later researchers, after then, large amount of work had been done to improve its computation efficiency from different aspects. J. Kim and P. Moin [42] focused on mass conservation, which was significant for numerical stability [36]. In order to maintain the mass conservation, the Chorin's method was coupled with an approximate-factorization technique, proposed by R. Beam [61] and W. R. Briley [78], on a staggered grid. J. Kim and P. Moin also noticed that the concurring boundary condition used in original Chorin's method led to inconsistent and erroneous results. Therefore, an appropriate boundary condition was derived for intermediate velocity by the similar technique in [64].

While some researchers concentrated on mass conservation and boundary condition, others tried to increase the accuracy of projection method. Braza [6] first hinted the possibility of improving the accuracy of Chorin’s method in the time to $O(\Delta t^2)$. Van Kan realized this possibility for some special cases by pressure correction. [40] He showed that his proposed pressure correction method in a system of ”constrained” ODE’s similar to the Navier-Stokes problem under consideration under reasonably weak assumptions lead to a solution with $O(\Delta t^2)$ accuracy. P. Colella [52], Van Leer [44] and J. B. Bell [34] worked on the efficient discretization for the nonlinear convection terms included in the Navier- Stokes equation, they proposed and developed similar explicit second-order Godunov methods. All above methods were summarized by J. B. Bell [35], and a second-order projection method, for time-dependent, incompressible single phase Navier-Stokes equations, was defined based on the previous improving techniques for Chorin’s method. With the assumption of no external forces and homogeneous, this second-order method was applied to smooth flow for stokes flow, Reynolds number 100, and incompressible Euler equations on uniform mesh spacing.

The development of interface capturing technique in the 1990s may be one reason that the study of the Navier-Stokes equations turned to multi-phase flow. J. Zhu [38] merged modern techniques for computing the solution to the viscous Navier-Stokes equations with level set method for computing the motion of interface propagating with curvature-dependent speeds. J. B. Bell [33] also described a second-order projection method for variable density incompressible flows. However, these works were still limited in the relatively simple cases where no jump condition

of the pressure across the interface was enforced.

Direct numerical computation of incompressible two-phase flow on Cartesian grids involves solving the Poisson's Equation for variables that are discontinuous across phase boundaries. A variety of discretization methods are available to impose the jump conditions representing the discontinuity. The immersed boundary (ImB) method was among the earlier methods used for solving the Poisson's equation for discontinuous variables [8, 16, 53, 54, 55]. The interface was viewed as being immersed in the flow field and moving with local fluid velocity where a force was applied in the fluids to create the jump discontinuity at the interface. Later, Peskin [7] improved upon the basic approach and developed a first order numerical algorithm for the solution of Poisson equation in a thin finite band around the interface. The method involved finding a boundary force from interface configuration that was applied to the grid for computing fluid velocity with the help of a δ function. The new velocity was interpolated to the old boundary position and the boundary points were moved by the interpolated velocity. Numerous methods have appeared since, that aim to improve and refine the basic ImB method. Depending on how the forcing is introduced during discretization, existing approaches derived from the basic ImB method can be divided into continuous and discrete forcing methods [48]. Many later works also combined the ImB method with interface tracking schemes to solve two-phase flow problems [72, 73].

To achieve a more general treatment of interface jump conditions, the immersed interface (ImI) method was developed by LeVeque and Li [63]. The method was shown to achieve second order accuracy in space when applied to the Poisson's

equation for variables with both Dirichlet and Neumann discontinuities across the interface. The ImI method derived appropriate coefficients for Poisson's equation at the grid points on a stencil that contains extra points chosen from the set of diagonally adjacent grid points near the interface. The ImI method avoids interface smearing and produces sharp solutions across the interface. The original ImI method was improved by a direct finite difference discretization [45]. An M-matrix, whose symmetric part is negative definite, was built to guarantee that the new ImI method converges and satisfies the maximum principle. By constrained optimization techniques, the ImI method was also applied for three dimensional Poisson's equation [22]. Another notable technique, comparable to the ImI method, was obtained by Johansen and Colella [39]. The authors used a finite-volume discretization, which embeds the domain in a regular Cartesian grid, to solve the Poisson's equation with variable coefficients. The method was shown to handle only Dirichlet jump conditions and led to non-symmetric Poisson coefficient matrices.

Both the ImB and ImI methods have their limitation. The ImB method does not maintain the discontinuity of the solution, can only handle Dirichlet jump conditions and is at most first order accurate. Even though the ImI method was shown to achieve second accuracy and led to sharp solutions, the associated Poisson coefficient matrix does not satisfy the negative adjoint relationship between the gradient and divergence operators that is needed for conservation and symmetry. Consequently, it is difficult to use this method for the solution of two-phase flow as well as to employ standard, fast linear solvers. A fast iterative algorithm for the ImI method was however developed by Li [46].

The Ghost Fluid Method (GFM) introduces an artificial fluid which induces the proper interface conditions. This method can be implemented in three steps. At first, the ghost cell is used to define each fluid at every point in the computational domain. Then, each fluid is separately updated in each spatial dimension. Thirdly, the level set function, which determines the location of interface, is advected independently on the basis of the actual fluid velocity. In [58], it was shown that GFM is free of spurious oscillations. The GFM can easily be extended to multi-dimensions because the jump conditions are handled implicitly. Because of this, the GFM is ideal for problems, such as shocks, detonations, and deflagrations in [59] and compressible viscous flows in [60]. Liu [79] proposes a method that is similar to GFM to solve the variable coefficient Poisson's equation in the presence of interfaces across which both the variable coefficient and the solution itself may be discontinuous. This first order accurate method resulted in a symmetric coefficient matrix. Furthermore, it could be extended to three spatial dimensions easily as it uses a standard finite difference discretization on a Cartesian grid. However, this approach cannot be combined with the Projection method to obtain a divergence free velocity field.

In this work, a new method is presented here for solving the variable coefficient Poisson's equation on uniform Cartesian grids with both Dirichlet and Neumann jump conditions across the interface. The method achieves at least second order formal accuracy and the Poisson coefficient matrix is symmetric. The approach is based on the idea of overlapping phases in interfacial cells such that their collective effect can be expressed by a volume fraction weighted average. A correction to

the discretized truncation error is introduced to achieve high accuracy of the global solution with jumps in the magnitude and derivative across the interface. The new method is robust and can be implemented using a standard finite difference discretization on a Cartesian grid. It can be therefore easily extended to three spatial dimensions. There is no numerical smearing and the discontinuities are well preserved. The new method is thus suitable for the Navier-Stoke's equations for incompressible two-phase flow. Several two-phase flow solutions have been report here in Chapter 4.

1.4 Goals

The primary objectives of this work are as follows:

- An accurate and robust implicit interface advection scheme to decouple the advection and reinitialization.
- A boundary condition capture method to solve the Possion's equation on Irregular domains with second order accuracy
- The application of above methods for multiphase incompressible flow

1.5 Contribution

The contributions made by this works for multiphase flow computation are as follows:

- An evaluation system has been developed to quantify four kinds of errors, e.g.,

distance function error, curvature error, normal error and volume error. By this system, a numerical advection methods can be evaluated from different aspects comprehensively.

- With the help of the above evaluation system, three main popular numerical methods were studied, e.g., classical reinitialization scheme, pure implicit advection scheme and recompression scheme.
- A stepwise geometric projection scheme has been developed theoretically to overcome the challenges arising in both volume conservation and the topological distortion.
- The implementation of the geometric projection scheme has been extended for three spatial dimension.
- A second order accurate scheme, which used a correction to preserve the jump condition across the interface, has been developed.
- The second order accurate boundary capture scheme has been extended to three spatial dimension.
- The boundary capturing method is associated with the projection method to solve two-phase flow problems.

Chapter 2: Topology Preserving Advection of Implicit Interfaces on Cartesian Grids

2.1 Overview

In this chapter, the development of our new method for interface advection will be presented. It is also demonstrated that the lack of volume conservation and the distortion of the implicit interface associated with previous methods is due to the coupled, nonlinear nature of advection and reinitialization. The concurrent advection of the implicit interface associated with the corrected scalar fields, amplifies reinitialization errors. In principle, such errors can be minimized by decoupling advection and reinitialization, leaving the implicit interface unperturbed and subject only to advection errors. Reinitialization is then needed only to provide the necessary interface topology such as the interface normals and curvature on the grid. Existing reinitialization methods do not admit this form of decoupling of advection and reinitialization. This is because these methods can be only correct for small deviations from the ideal condition and cannot allow the scalar field to become too distorted over time.

To facilitate the decoupling of advection and reinitialization, a general ap-

proach is proposed for constructing a signed distance function from a set of discrete interfacial pointers. The method is developed specifically for implicit interfaces that evolve with the Eulerian advection of a scalar field but also demonstrate its general validity for interfaces that may be available in the form of a connected set of marker particles that evolve with Lagrangian advection. The approach involves the construction of the signed distance function by geometrically projecting vectors from each grid point, normal to the interface. The projected interface and its topology are embedded implicitly within the resulting signed distance function on the Cartesian grid. This procedure is applied independently at every time step, without modifying the underlying scalar field, allowing it to evolve free of reinitialization errors. Combined with high accuracy advection schemes, it is demonstrated that this approach results in substantial improvement, in comparison with existing methods, with regards to both the volume conservation errors and the representation of the interface normals and curvature on the grid.

The approach to geometric projection developed in this work is based on a stepwise improvement in the approximation to the signed distance function, implemented as the minimum distance from the grid to, 1) the interface marker, 2) a locally piecewise interface reconstruction and 3) a locally smooth interface reconstruction of associated interface segments. The method of propagation of interface topology away from the interface differs from the fast marching method in that a direct projection of the normal from all grid points where projection is needed to the interface is used without the need for building extensional velocities. This results in each grid point directly inheriting the topology of the projected interface, which

is found to result in greater accuracy than previously reported. It is shown that advection based on level one projection is as accurate as existing schemes based on the reinitialization of the level set and indicator functions but is significantly more efficient even when applied to the whole domain. It is therefore used as a means for regularizing the scalar field away from the interface when needed, without perturbing the implicit interface. Levels two and three are more accurate and but also more expensive. However, they are only needed in a narrow band around the interface. The projection method is shown to be robust in easily accounting for complex topological changes with an easily implementable extension to 3-D. It has been found that for two phase flow the level 2 projection generally performs equally well as the more expensive level 3 projection and is therefore more appealing for 3-D implementation.

This chapter is organized as follows. In section 2.2 the algorithm for geometric projection is described. In section 2.2.1 the implementation is developed for the interface advected by Lagrangian advection of marker particles. In section 2.2.2 the implementation is applied for implicit interfaces associated with the level set function. The effectiveness of the approach for two phase flow problems is demonstrated in section 2.3. Finally, the implementation of the approach in three spatial dimension was provided in Appendix A.

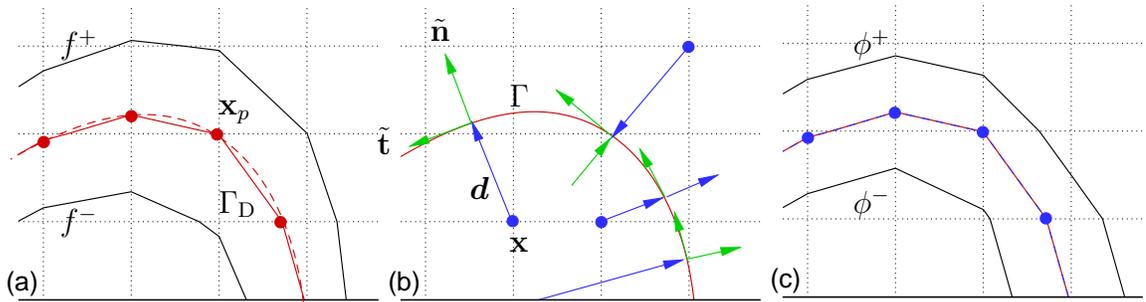


Figure 2.1: Sketch illustrating the principle of geometric projection. Plot (a) shows contours of a level set function f . Points \mathbf{x}_p correspond to $f = 0$ and mark the interface Γ_D . Plot (b) shows the smooth interface Γ associated in some manner with Γ_D . The distance vector \mathbf{d} at grid point \mathbf{x} is associated with a unique point on Γ defined by Eq. (2.1). Plot (c) shows the resulting signed distance function ϕ . Blue indicates $\phi = 0$ that coincides with $f = 0$ shown in red.

2.2 Geometric projection of interface topology

Given a level set function $f(\mathbf{x})$, a signed distance function $\phi(\mathbf{x})$ can be obtained by projecting the topology of the interface Γ that is associated with $f(\mathbf{x}) = 0$ onto the Cartesian grid G . The signed distance function $\phi(\mathbf{x})$ represents the shortest distance $|\tilde{\mathbf{d}}(\mathbf{x})|$ between points $\tilde{\mathbf{x}} \in \Gamma$ and $\mathbf{x} \in G$. The distance vector $\tilde{\mathbf{d}}(\mathbf{x})$ satisfies the condition,

$$\tilde{\mathbf{d}}(\mathbf{x}) = \{(\tilde{\mathbf{x}} - \mathbf{x}) : (\tilde{\mathbf{x}} - \mathbf{x}) \cdot \tilde{\mathbf{t}}_1 = 0, (\tilde{\mathbf{x}} - \mathbf{x}) \cdot \tilde{\mathbf{t}}_2 = 0\} , \quad (2.1)$$

where $\tilde{\mathbf{t}}_1$ and $\tilde{\mathbf{t}}_2$ are a pair of orthogonal unit vectors tangent to the interface, Γ . The projection of vectors normal to the interface onto the distance vector $\tilde{\mathbf{d}}(\mathbf{x})$ represents the signed distance function

$$\tilde{\phi}(\mathbf{x}) = -\tilde{\mathbf{n}} \cdot \tilde{\mathbf{d}}(\mathbf{x}) , \quad (2.2)$$

where $\tilde{\mathbf{n}}$ is a unit vector normal to the interface, Γ . A sketch illustrating this principle is shown in Fig. 2.1. Figure 2.1(a) shows the contours of the level set function $f(\mathbf{x})$. A discrete interface, Γ_D , associated with $f(\mathbf{x}) = 0$, exists between the positive, f^+ , and negative, f^- , contours of $f(\mathbf{x})$ and is indicated by the set of interfacial points, $\mathbf{x}_p \in \Gamma_D$. This discrete interface Γ_D is used to construct a smooth interface Γ , shown with the dashed red line in Fig. 2.1(a). The smooth interface Γ is used to form the distance function $\tilde{\mathbf{d}}(\mathbf{x})$ from grid points \mathbf{x} to points $\tilde{\mathbf{x}}$ on the smooth interface according to Eqs. (2.1) and (2.2). Figure 2.1(b) illustrates the distance vector and vectors normal and tangent to the interface. The signed distance function $\phi(\mathbf{x})$ resulting from geometric projection, as shown in Fig. 2.1(c),

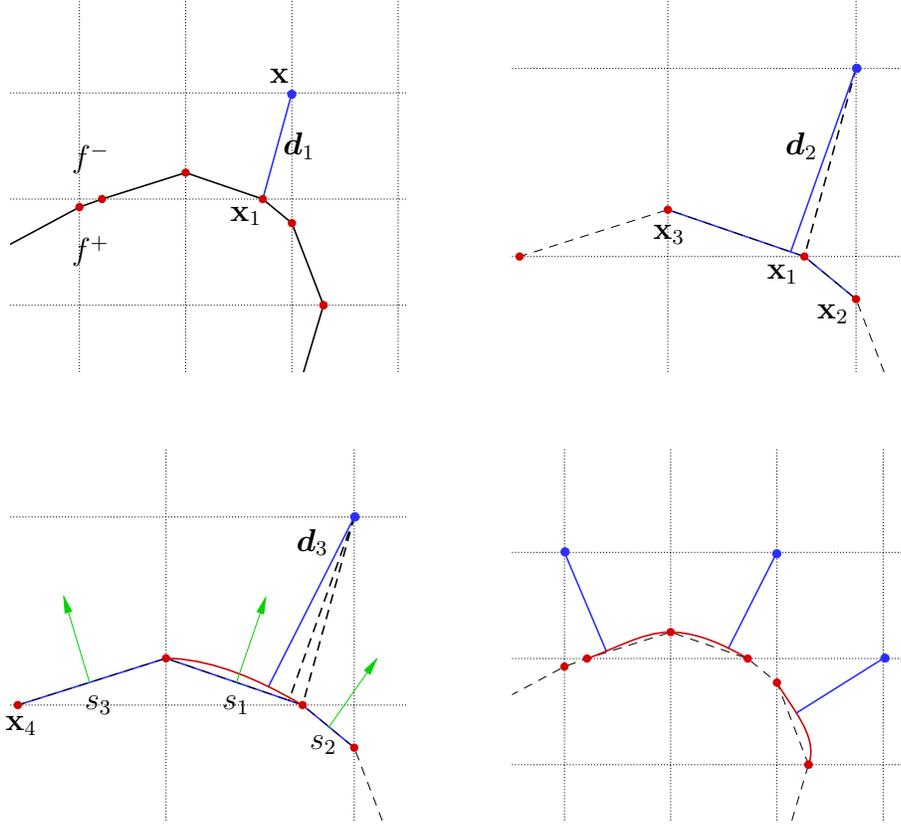


Figure 2.2: Implementation of the stepwise refinement of geometric projection.

(a) Localization of the interface on the grid, indicated with red symbols. Level 1 projection (P_1) is the shortest distance from the cell center to an interface node (blue line). (b) Two nearest neighbors of the closest node define two interface segments (blue). Projection from cell center to a segment represents Level 2 projection (P_2). (c) One more segment is added. A 3rd order curve is defined by respective segment normals and interface nodes for the segment selected by P_2 . Projection from cell center normal to this curve is Level 3 projection (P_3). (d) A few segments represented by smooth curves resulting from P_3 .

provides an approximation for both the distance function condition, $|\nabla\phi(\mathbf{x})| = 1$, and the discrete interface Γ_D with $\phi(\mathbf{x}) = 0$. The errors associated with these approximations can be clearly characterized as functions of grid size and interface curvature, as discussed in detail below.

The practical implementation of geometric projection requires, first, the localization of the discrete interface Γ_D associated with the level set function $f(\mathbf{x}) = 0$ on the Cartesian grid G . Secondly, a smooth interface Γ needs to be constructed on the basis of the discrete interface Γ_D , as shown in Fig. 2.1. Finally, a robust method is needed for determining the distance function $\mathbf{d}(\mathbf{x})$ and the signed distance function $\phi(\mathbf{x})$ based on Eqs. (2.1) and (2.2), respectively. Localization of the interface is achieved with a 2nd interpolation of $f(\mathbf{x})$ to $f = 0$. The procedures for the construction of Γ from Γ_D and the subsequent determination of $\phi(\mathbf{x})$ are described below .

Given a discrete interface, $\mathbf{x}_p \in \Gamma_D$, we develop a three level methodology for obtaining successively refined approximations to the signed distance function $\phi(\mathbf{x})$. The procedure is illustrated in Fig. 4.1 for an interface in a plane. The first step, shown in Fig. 4.1(a), is the identification of the shortest distance from the grid point \mathbf{x} to any point \mathbf{x}_p on the interface,

$$|\mathbf{d}_1(\mathbf{x})| = |\mathbf{x}_1 - \mathbf{x}| = \min_p |\mathbf{x}_p - \mathbf{x}| . \quad (2.3)$$

where \mathbf{d}_1 , indicated by the blue line in Fig. 4.1(a) represents the distance function obtained with the first step of projection, referred to as Level 1 projection (P_1). The next step, illustrated in Fig. 4.1(b), consists of finding two neighboring points, \mathbf{x}_2

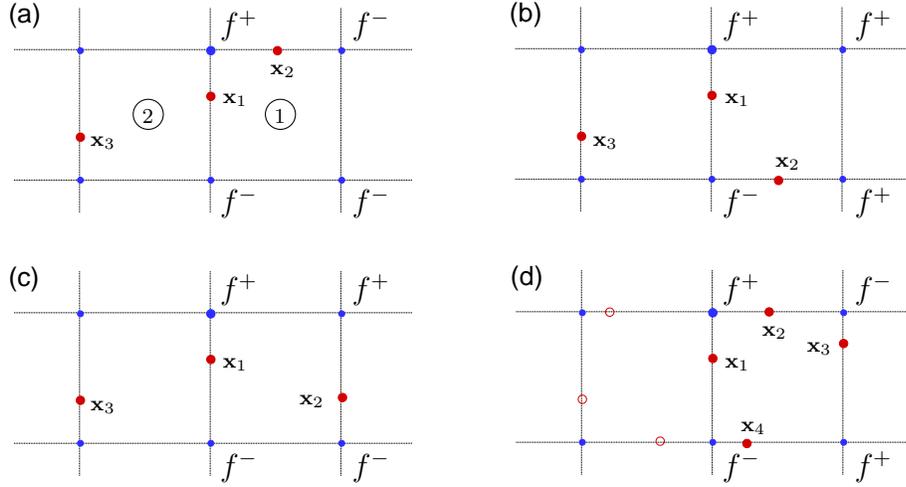


Figure 2.3: Typical arrangement of four interfacial points defining a piecewise continuous interface for the implementation of P_2 and P_3 projections.

and \mathbf{x}_3 , with respect to \mathbf{x}_1 , according to,

$$|\mathbf{x}_1 - \mathbf{x}| > \{\min_p |\mathbf{x}_p - \mathbf{x}| : \mathbf{x}_p \neq \mathbf{x}_1\} > \{\min_p |\mathbf{x}_p - \mathbf{x}| : \mathbf{x}_p \neq \mathbf{x}_1, \mathbf{x}_2\} . \quad (2.4)$$

These three points define two interface segments, as shown in Fig 4.1(c). The distance function, \mathbf{d}_2 is then obtained from the intersection of a line that originates at \mathbf{x} with one of the segments s_1 such that $\mathbf{d}_2 \cdot \mathbf{t}_{s_1} = 0$, where \mathbf{t}_{s_1} is the vector tangent to s_1 . We refer to the distance function obtained with this procedure as Level 2 projection (P_2).

The third step of geometric projection involves the construction of a smooth curve connecting points \mathbf{x}_1 and \mathbf{x}_2 as illustrated in Fig 4.1(c). This is obtained with a fourth order ODE,

$$\hat{y}^{IV}(\hat{x}) = 0 , \quad (2.5)$$

where $\hat{\mathbf{x}}$ represents a coordinate system aligned with the segment s_1 . Equation (2.6)

is solved with four boundary conditions,

$$\hat{y}|_{\hat{x}=0} = \hat{y}|_{\hat{x}=|\mathbf{x}_1-\mathbf{x}_2|} = 0, \quad \hat{y}'|_{\hat{x}=0} = \gamma_1 \quad \text{and} \quad \hat{y}'|_{\hat{x}=|\mathbf{x}_1-\mathbf{x}_2|} = \gamma_2 \quad (2.6)$$

The slopes γ_1 and γ_2 are determined by the normals associated with pairs of segments, (s_1, s_3) and (s_1, s_2) , respectively. Specification of the segment s_3 requires an additional interfacial point \mathbf{x}_4 , as shown in Fig 4.1(c), which is obtained in a manner similar to that for the point \mathbf{x}_3 , given by Eq. (2.2). This procedure of projection will be referred to as Level 3 projection (P_3). The curve obtained with P_3 requires only the specification of four interfacial points and connects smoothly with adjoining segments that are formed independently. Once all local interface segments are constructed during the process of projection to grid points near the interface, the corresponding parameters need not be computed for projection to other grid points farther away from the interface. Figure 4.1(d) shows typical curves obtained for individual segments with P_3 based on projection from various grid points. A signed distance function can be associated with the distance function at any stage of projection according to

$$\phi_i(\mathbf{x}) = \frac{f(\mathbf{x})}{|f(\mathbf{x})|} |\mathbf{d}_i(\mathbf{x})| \quad (2.7)$$

where subscript, i , indicates the level of projection, i.e, P_1 , P_2 or P_3 .

In order for the above procedure to be robust, the interfacial points $\{\mathbf{x}_i, i = 1, 2, 3, 4\}$, associated with the nodal point \mathbf{x} , need to be identified in a general manner that is effective for all complex topological changes of the interface such as merging and separation. All four interfacial points, \mathbf{x}_i , must belong to the same interface and must also be piecewise continuous. Finding \mathbf{x}_1 according to Eq. (2.2)

is straightforward and is unaffected by nearby interfaces. In order to ensure that $\{\mathbf{x}_i, i = 2, 3, 4\}$ belong to the same interface, a local search procedure is employed as illustrated in Fig. 4.2.

The search involves examining the faces of two cells around \mathbf{x}_1 that are formed by connecting grid nodes associated with the level set function, $f(\mathbf{x})$. These cells are marked as 1 and 2 in Fig. 4.2(a) where the interfacial point for projection, \mathbf{x} , is shown by the bold blue symbol. For a piecewise continuous interface, points \mathbf{x}_2 and \mathbf{x}_3 must reside on any one of the three free faces of cells 1 and 2 according to $|\mathbf{x}_1 - \mathbf{x}_2| > |\mathbf{x}_1 - \mathbf{x}_3|$. For the case shown in Fig. 4.2(a), \mathbf{x}_2 and \mathbf{x}_3 are located, for example, on the top face of cell 1 and the left face of cell 2, respectively. The cell vertices indicate the corresponding positive or negative values of $f(\mathbf{x})$. Similarly, Figs. 4.2(b) and (c) show other cases where \mathbf{x}_2 resides on the bottom and the right face of cell 1, respectively, with \mathbf{x}_3 unchanged for illustration. The fourth point, \mathbf{x}_4 , is obtained by following a similar principle. For example, \mathbf{x}_4 would lie in the cell, above cell 1, below cell 1 and to the right of cell 1 for the arrangements shown in Figs. 4.2(a), (b) and (c), respectively. The case of \mathbf{x}_1 on a horizontal cell face is dealt with in a similar manner where cells 1 and 2 would exist above or below the cell face associated with \mathbf{x}_1 .

A situation where a cell may contain more than two interfacial points is shown in Fig. 4.2(d). The corresponding arrangement of $f(\mathbf{x})$ on cell vertices indicates that there can be only either two or four interfacial points in a cell. The four points in cell 1 hence form a closed interface that is at the limit of its resolution. Three more interfacial points in cell 2 are plotted to illustrate the existence of another interface

Table 2.1:

Number of particles needed as a function of grid size by various projection schemes for representing ϕ related to a circular interface with an accuracy of $\mathcal{O}(10^{-8})$. The sign and the digit on the left indicate the exponent of, $\times 10$.

Δx	P ₁	P ₂	P ₃
1/32	1.0+3	1.0+2	1.0+1
1/64	1.0+4	1.5+3	1.0+2
1/128	5.0+4	6.0+3	5.0+2
1/256	4.0+5	4.0+4	2.5+3

that shares a common point, \mathbf{x}_1 , with the interface in cell 1. This arrangement indicates that either the two interfaces are on the cusp of merging or one interface has split into two. The examples shown in Fig. 4.2 together illustrate all possibilities with regards to the location of interfacial points. The identification of connected interfacial points in the manner described above is therefore computationally efficient and robust. The extension to 3-D is thus straightforward.

In the following sections we evaluate the accuracy of each of the three different levels of projection. In section 2.2.1, we consider projection of an interface that is available in the form of Lagrangian marker particles as well as projection from a discrete interface defined by the level set function. We then quantify cumulative errors of reinitialization and advection and compare with previous reinitialization schemes.

2.2.1 Projection of particle based interface

The effectiveness of geometric projection for the construction of the signed distance function on Cartesian grids can be demonstrated clearly when the underlying interface Γ is available as a set of marker particles, $\mathbf{x}_p \in \Gamma$. Unlike the interface defined by localizing the level set function described above, \mathbf{x}_p are independent of the grid in this case. Projection errors can therefore be evaluated as a function of the number of particles for a given grid size, independently of localization and advection errors. Consideration of projection characteristics related to particle based interfaces is also of interest when the interface is advected as a Lagrangian object, the topology of which needs to be periodically projected onto a Cartesian grid.

For the purpose of illustration, marker particles are taken to be advected with the analytical 2-D flow field produced by a single vortex in a plane

$$u = \sin^2(\pi x) \sin(2\pi y) \cos(\pi t/T) , \quad (2.8)$$

$$v = -\sin(2\pi x) \sin^2(\pi y) \cos(\pi t/T) . \quad (2.9)$$

The direction of u and v reverses periodically with time period T . The advection of Lagrangian marker particles $\tilde{\mathbf{x}}_p(t) \in \Gamma(t)$ is carried out with

$$\frac{d}{dt} \tilde{\mathbf{x}}_p(t) = \mathbf{u}(\tilde{\mathbf{x}}_p(t), t) , \quad \tilde{\mathbf{x}}_p(0) \in \Gamma(0) . \quad (2.10)$$

The initial condition $\Gamma(0)$ is defined at $t = 0$ by

$$(\tilde{x}_p(t_o) - x_o)^2 + (\tilde{y}_p(t_o) - y_o)^2 = r_o^2 , \quad (2.11)$$

for $t_o = 0$, $x_o = 0.5$ and $y_o = 0.75$ and $r = 0.15$. Equation (2.10) is solved with a standard 4th order Runge-Kutta scheme.

To understand the nature of numerical errors associated with geometric projection from a fundamental standpoint, we first examine the errors in the construction of ϕ for a circular interface at $t = 0$ for which an exact solution for the signed distance function is available as

$$\tilde{\phi}(x, y) = \sqrt{(x - x_o)^2 + (y - y_o)^2} - r . \quad (2.12)$$

The three projection schemes described above are used to compute the signed distance function, which is compared with the exact solution given by 2.12 with respect to the L^2 norm in the domain $x \in [0, 1]$ $y \in [0, 1]$. We find that given enough particles, all projection schemes can represent the exact solution to any desired accuracy. Table 1 shows the number of particles needed as a function of grid size to represent the exact solution $\tilde{\phi}$ with an accuracy of $\mathcal{O}(10^{-8})$. The P_1 projection needs about 1 and 2 orders of magnitude more particles than P_2 and P_3 projections, respectively, for the same level of accuracy on a given grid size. The number of particles needed for convergence also increases by about an order of magnitude, which is because projection errors increase for grids points closer to the interface.

The accuracy of representing the topology of an interface on the Cartesian grid can be determined on the basis of errors related to the distance function, $|\nabla\phi|$, the interface normal, n_x , and the curvature, κ . The exact solution for these quantities for a circular interface is given, respectively, by

$$|\nabla\tilde{\phi}| = 1 , \quad \tilde{n}_x = (x - x_o)/r , \quad \text{and} \quad \tilde{\kappa} = 1/r \quad (2.13)$$

Table 2.2:

Grid convergence of interface topology represented by the distance function, $|\nabla\phi|$, the interface normal, n_x and curvature κ for ϕ related to a circular interface. The sign and the digit on the left indicate the exponent of, $\times 10$. The last row lists the exponent of the rate of convergence in, $\mathcal{O}(\Delta x^e)$, based on the first and the last error value.

Δx	$1 - \nabla\phi $	n_x	κ
1/32	2.27-3	8.92-3	8.06-2
1/64	5.74-4	2.32-4	2.16-2
1/128	1.44-4	5.88-5	5.50-3
1/256	3.62-5	1.47-5	1.38-3
	1.99	3.08	1.95

These quantities are computed numerically as

$$|\nabla\phi| = \sqrt{\phi_x^2 + \phi_y^2}, \quad n_x = \phi_x/|\nabla\phi|, \quad \text{and} \quad \kappa = \nabla \cdot \nabla\phi/|\nabla\phi| \quad (2.14)$$

where ϕ results from geometric projection. The calculation of ϕ is independent of the projection scheme as long as sufficient number of particles given in Table 1 are used. A second order, central, finite difference scheme is employed for the numerical approximation of the spatial derivatives of ϕ . The respective errors, E in the numerical solution with respect to the exact solution is determined by

$$E = \left[\frac{1}{\Delta x \Delta y} \sum_{i=1}^N \sum_{j=1}^N H_{i,j} (\tilde{g}_{i,j} - g_{i,j})^2 \right]^{1/2} \quad (2.15)$$

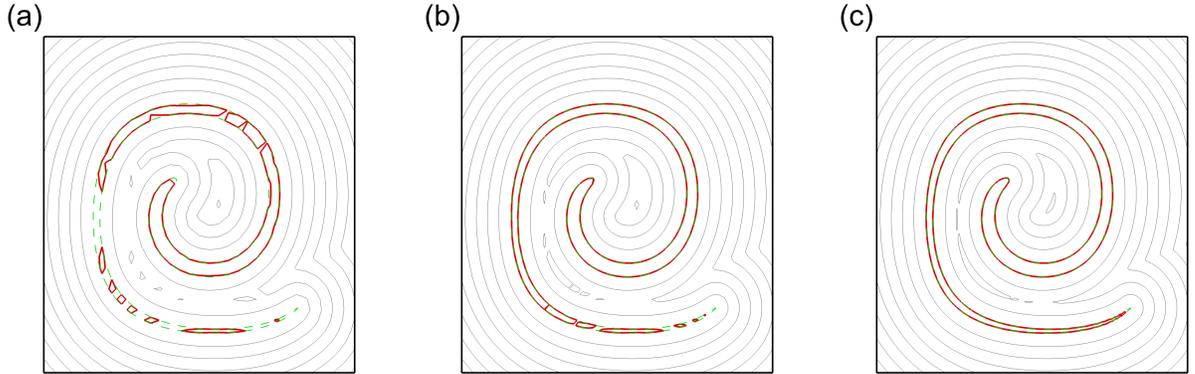


Figure 2.4: Contours of the signed distance function, $\phi(\mathbf{x}, t)$, obtained with geometric projection based on the particle interface advected with Eq. 2.10 at $t = 2$ for $T = \infty$ and grid sizes, (a) 33×33 , (b) 65×65 and (c) 129×129 . Solid red lines indicate the zero-contour of ϕ and dashed green lines show the smooth interface.

where N represents the number of grid points, and \tilde{g} and g represent the exact and numerical solutions, respectively, for the distance function, $|\nabla\phi|$, the interface normal, n_x , and the curvature, κ . The function H is a filter for excluding the singularity in the curvature at $(x = 0, y = 0)$. It takes on values of zero close to the center of the circle and unity elsewhere. The errors computed with Eq. (2.15) are listed in Table 2 as a function of the grid size. All errors decay with 2nd order accuracy, confirming the effectiveness of geometric projection.

The signed distance function obtained by geometric projection at late times, when the interface is substantially deformed with respect to the initial state, is depicted in Fig. 2.4 at $t = 2$ for $T = \infty$ for different grid sizes. The solid red lines indicate the interface Γ_D defined by $\phi = 0$ and the dashed green line marks the Lagrangian solution, $\mathbf{x}_p \in \Gamma$. The number of Lagrangian particles is taken

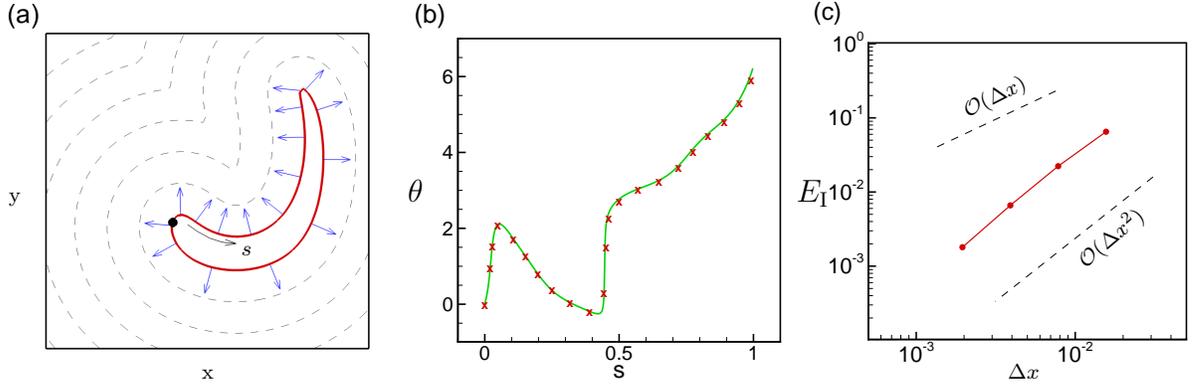


Figure 2.5: Contours of ϕ obtained with geometric projection for $t = 1$ and $T = 2$ on a 65×65 grid. (b) $\theta = \arg(\mathbf{n})$ as a function of arc length s . The origin and direction of s are indicated in (a). Solid line and symbols represent θ related to Γ and Γ_D , respectively. (c) Convergence of θ for $t = 1$ and $T = 2$.

to be consistent with the accuracy requirement noted in Table 1. A progressively more accurate representation Γ_D is obtained on successively refined grids for the same underlying Lagrangian interface, Γ . The signed distance function is moreover observed to be constructed accurately at arbitrarily large distances away from the interface.

Curvature singularities can arise, as shown in Fig. 2.4, within regions of insufficient resolution at the interface as well as on the grid due to discontinuities in the signed distance function at the intersection of projected normals. Determination of grid errors in this case is not straightforward particularly in the absence of an exact solution although, clearly, the representation of the Lagrangian interface improves with grid refinement. As an alternative, it is useful to consider the accuracy with which Γ_D represents Γ , which is known “exactly” as long as sufficient number of

particles are employed.

Table 2.3:

Errors related to the projection of the implicit interface with respect to the exact solution for a circle. The last row lists the exponent of the rate of convergence in, $\mathcal{O}(\Delta x^e)$, based on the first and the last error value.

ϕ				$ \nabla\phi $			
Δx	P_1	P_2	P_3	Δx	P_1	P_2	P_3
1/32	1.08-3	2.77-4	1.31-4	1/32	3.18-2	3.92-3	3.02-3
1/64	3.20-4	4.89-5	2.13-5	1/64	2.04-2	9.52-4	5.64-4
1/128	1.06-4	8.49-6	3.61-6	1/128	1.38-2	3.58-4	1.63-4
1/256	3.28-5	1.52-6	6.29-7	1/256	8.40-3	1.29-4	5.39-5
	1.71	2.52	2.62		0.64	1.64	1.93

n_x				κ			
Δx	P_1	P_2	P_3	Δx	P_1	P_2	P_3
1/32	1.17-2	2.26-3	9.50-4	1/32	4.71-1	1.00-1	7.85-2
1/64	5.18-3	6.47-4	3.33-4	1/64	3.83-1	5.68-2	3.52-2
1/128	2.77-3	1.86-4	1.18-4	1/128	4.20-1	2.97-2	1.95-2
1/256	1.76-3	8.11-5	3.62-5	1/256	5.16-1	2.67-2	8.44-3
	0.91	1.63	1.65		-0.04	0.63	1.07

Figure 2.5(a) plots Γ_D at $t = 1$ for $T = 2$. The argument of the normal vector $\theta = \arg(\mathbf{n})$, measured in radians, is plotted in Figure 2.5(b) as a function of position

s along the interface. The origin of arc length s and its direction are indicated in Fig. 2.5(a). The solid line Fig. 2.5(b) refers to the argument of the normal vector associated with the smooth interface Γ while symbols indicate θ related to Γ_D associated with $\phi = 0$ on a 65×65 grid. The normal vectors for Γ are obtained as $d\mathbf{x}_p/ds$ while those associated with Γ_D are obtained by a 2nd order interpolation to $\phi = 0$ from the grid values of \mathbf{n} . Steep changes in $\theta = \arg(\mathbf{n})$ coincide with the sharp ends of the interface plotted in Fig. 2.5(a). The normal vectors related to the discrete interface Γ_D are in good agreement with their counterparts related to the smooth interface, Γ , even for a small grid size. The error can be measured by

$$E_I = \left[\int [\arg(\tilde{\mathbf{n}}) - \arg(\mathbf{n})]^2 ds \right]^{1/2}, \quad (2.16)$$

where the subscript I indicates that the error is measured at the interface. Figure 2.5(c) shows that convergence with grid refinement is 2nd order accurate. The above analysis indicates that given an interface defined by sufficiently large number of particles, an accurate signed distance function can be constructed with geometric projection. Interface topology of the projected interface converges with second order accuracy with respect to both the grid solution as well as the interfacial values.

2.2.2 Projection of implicit interface

We now evaluate the accuracy of geometric projection for the case where the underlying interface is defined by an isosurface of a level set function. The set of interfacial points related to this discrete interface, $\mathbf{x}_D \in \Gamma_D$, is obtained by localizing the isosurface on the Cartesian grid. In this case the number of interfacial points are a

function of the grid size, unlike the case for the particle based interface considered above for which the grid size and the number of particles are independent.

Similar to the error analysis described above for the particle based interface, the accuracy of geometric projection of the implicit interface can be evaluated rigorously in comparison with $\tilde{\phi}$ defined by Eq. (2.12), which is an exact signed distance function. This exact solution is compared with ϕ obtained with the geometric projection of the discrete interface Γ_D associated with $\tilde{\phi} = 0$. The discrete interface is obtained by localizing $\tilde{\phi} = 0$ on the grid with a 2nd order interpolation.

Table 3 lists the errors, evaluated according to Eq. (2.15), for ϕ , $|\nabla\phi|$, n_x , and k , related to projections P_1 , P_2 and P_3 . The data in Table 3 shows that the error decreases uniformly in all cases expect for the error in κ for P_1 projection. The most accurate representation of the exact solution is achieved by P_3 projection for which the rate of convergence is highest for ϕ and decreases to 1.07 in the case of κ . The performance of P_2 projection is quite good as well expect that the rate of convergence for κ drops below 1, but the errors still decrease uniformly. It is interesting to note that though P_1 projection does not convergence for κ , it performs reasonably well for ϕ , $|\nabla\phi|$ and n_x . This implies that for problems with weak surface tension, P_1 projection would be suitable. Overall, data in table 3 demonstrates that geometric projection is an effective approach for the construction of the signed distance function, in particular for the curvature when either P_2 or P_3 projection is used.

We now turn to the projection of implicit interfaces that evolve on a Cartesian grid by the advection of a scalar field. We employ the following conservative

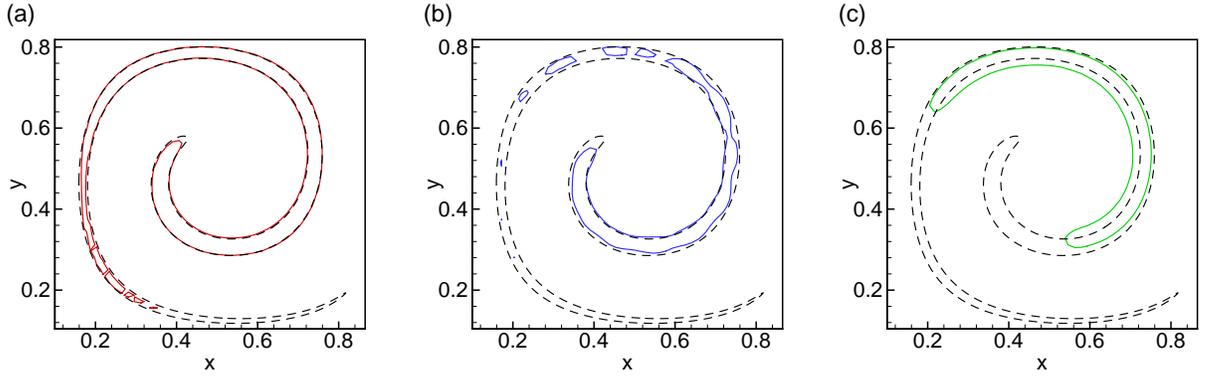


Figure 2.6: Interface profiles at $t = 2$ for $T = \infty$ obtained by various advection schemes on a 65×65 grid. (a) IA, (b) RC and (c) CR. The dashed line represents the corresponding Lagrangian solution.

hyperbolic conservation law for advection.

$$f_t + \nabla \cdot (f\mathbf{u}) = 0, \quad f(\mathbf{x}, 0) = \tilde{\phi}(\mathbf{x}). \quad (2.17)$$

The velocity field $\mathbf{u}(\mathbf{x}, t)$ is specified by Eqs. (2.8) and (2.9). At each time step, projections P_i are applied to $f(\mathbf{x}, t)$, according to $P_i : f(\mathbf{x}, t) \rightarrow \phi(\mathbf{x}, t)$, to obtain the signed distance function. The interface normal and curvature are then computed on the basis of $\phi(\mathbf{x}, t)$ leaving $f(\mathbf{x}, t)$ unchanged which is then advected to the next time step with Eq. (2.17). Advection and reinitialization are thus decoupled and are applied independently to the level set function. For the purpose of this study, we would refer to this method of advection, which is independent of reinitialization, as the Implicit Advection (IA) scheme.

We compare IA advection with two popular schemes that couple the procedures of advection and reinitialization. The first is the classical reinitialization (CR) method that advects the signed distance function ϕ with a conservative hyperbolic

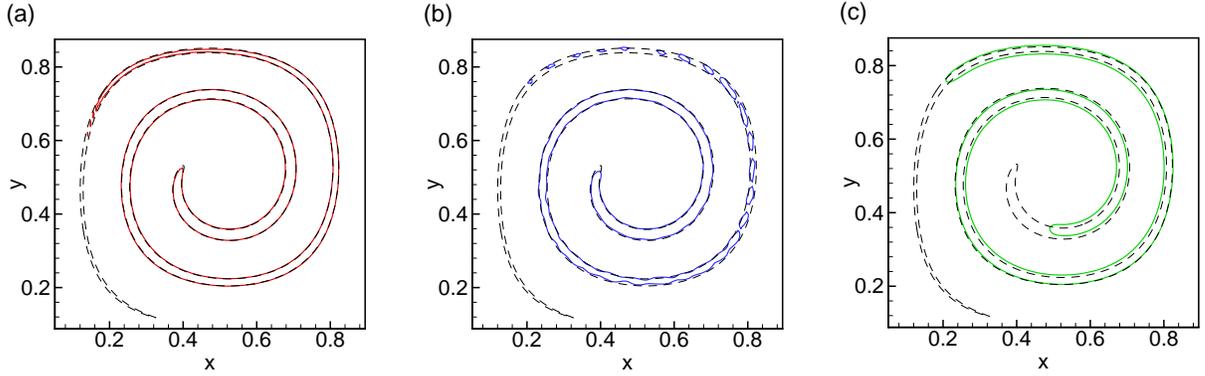


Figure 2.7: Interface profiles based on various schemes on a 129×129 grid for $t = 3$ and $T = \infty$. The dashed line depicts the smooth solution and solid lines indicate, geometric projection (a), classical recompression (b) and reinitialization (c).

conservation law,

$$\phi_t + \nabla \cdot (\phi \mathbf{u}) = 0, \quad \phi(\mathbf{x}, 0) = \tilde{\phi}(\mathbf{x}). \quad (2.18)$$

The advection of ϕ from one time level to the next does not preserve the signed distance function, i.e., $\mathcal{A} : \phi(\mathbf{x}, t - \Delta t) \rightarrow g(\mathbf{x}, t)$, where \mathcal{A} is the advection operation defined by Eq. (2.18) and $|g(\mathbf{x}, t)| \neq 1$. A procedure of “reinitialization”, $\mathcal{R} : g(\mathbf{x}, t) \rightarrow \phi(\mathbf{x}, t)$ where $|\phi(\mathbf{x}, t)| = 1$, is carried out that attempts to restore ϕ to a signed distance function. The reinitialization operation \mathcal{R} is implemented by iteratively solving the Hamilton-Jacobi equation,

$$\phi_\tau = \text{sgn}(g(\mathbf{x}, t))(1 - |\nabla \phi|), \quad \phi(\mathbf{x}, \tau = 0) = g(\mathbf{x}, t) \quad (2.19)$$

The quantity τ is a fictitious time step and $\text{sgn} = g/|g|$. A second order TVD Runge-Kutta time integration is used for Eq. (2.19) along with a second order TVD discretization of $(|\nabla \phi|)_{i,j}$, given by

$$|\nabla\phi|^2|_{i,j} = \begin{cases} \max(a_+^2, b_-^2) + \max(c_+^2, d_-^2) & , \quad g_{i,j} \geq 0 \\ \max(a_-^2, b_+^2) + \max(c_-^2, d_+^2) & , \quad g_{i,j} < 0 \end{cases} \quad (2.20)$$

where

$$a = \frac{1}{\Delta x}(\phi_{i,j} - \phi_{i-1,j}) - \frac{\Delta x}{2} \text{minmod}(D_{xx}\phi_{i,j}, D_{xx}\phi_{i-1,j}) , \quad (2.21)$$

$$b = \frac{1}{\Delta x}(\phi_{i+1,j} - \phi_{i,j}) - \frac{\Delta x}{2} \text{minmod}(D_{xx}\phi_{i+1,j}, D_{xx}\phi_{i,j}) , \quad (2.22)$$

with $a_+ = \max(a, 0)$ and $a_- = \min(a, 0)$. $D_{xx}\phi_{i,j}$ represents the 2nd order central discretization of the 2nd order derivative of the ϕ with respect to x . Similar expressions for c and d for the y -direction derivatives can be obtained.

We further compare the IA scheme with another class of methods where the implicit interface is defined by the isosurface of an indicator function (Olsson et. al, 2005,2007). An indicator function is an instance of the scalar function that varies smoothly across the implicit interface over a region of fixed width and is constant everywhere else. For example, given a signed distance function ϕ , the indicator function ψ can be defined as either

$$\psi = \frac{1}{2} \text{erf}(\phi/\varepsilon) \quad \text{or} \quad \psi = \frac{1}{2} \tanh(\phi/\varepsilon) \quad (2.23)$$

The implicit interface coincides in this case with $\psi = 1/2$. The width of the transition zone is constant along the interface and is set by ε . The indicator function is advected as before according to the hyperbolic conservation law,

$$\psi_t + \nabla \cdot (\psi \mathbf{u}) = 0 , \quad \psi(\mathbf{x}, 0) = \frac{1}{2} \text{erf}(\tilde{\phi}/\varepsilon) , \quad (2.24)$$

where $\tilde{\phi}$ is the initial condition for the IA and CR schemes described above. In order to maintain the constant width of the transition region around the interface

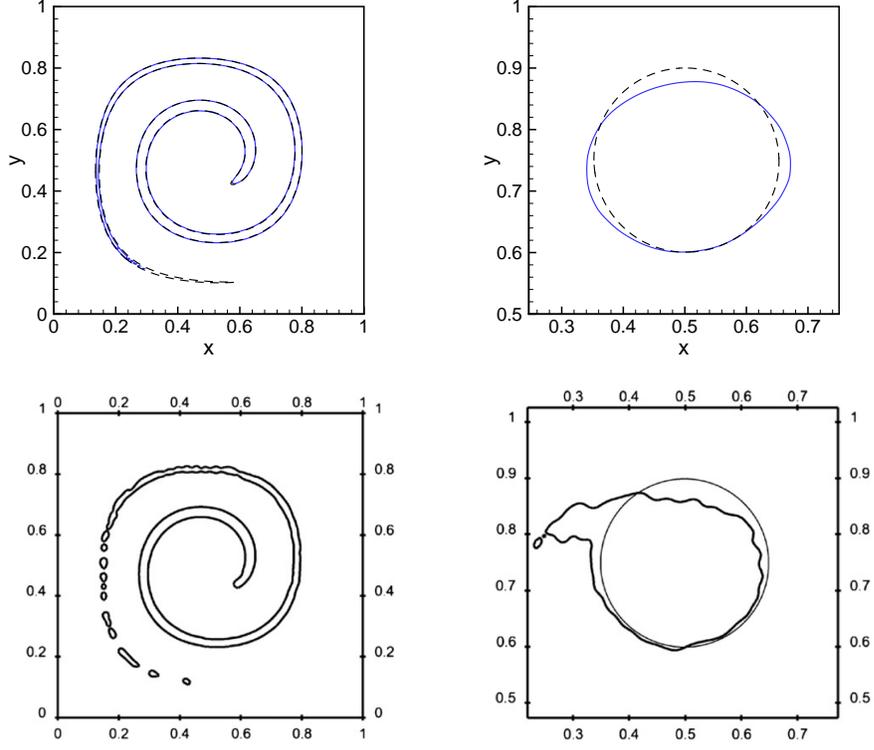


Figure 2.8: Interface profiles on a 129×129 grid at $t = 4$ (left) and $t = 8$ (right) for $T = 8$. Upper row is the interface obtained with our IA method where the dotted line depicts the Lagrangian solution. The lower row shows the solution of the modified RC method in Fig. 11 of [23].

at all times an iterative procedure of “reinitialization” is carried out at each time step according to,

$$\psi_\tau + \nabla \cdot \psi(1 - \psi)\mathbf{n} = \varepsilon \nabla \cdot (\nabla \psi \cdot \mathbf{n})\mathbf{n} , \quad \psi(\mathbf{x}, \tau = 0) = \hat{\psi}(\mathbf{x}, t) . \quad (2.25)$$

The smeared indicator function $\hat{\psi}$ results from, $\mathcal{A} : \psi(\mathbf{x}, t - \Delta t) \rightarrow \hat{\psi}(\mathbf{x}, t)$, where \mathcal{A} is the advection operation defined by Eq. (2.24). The reversal of numerical smearing is achieved by balancing nonlinear convection and linear diffusion of strength ε in

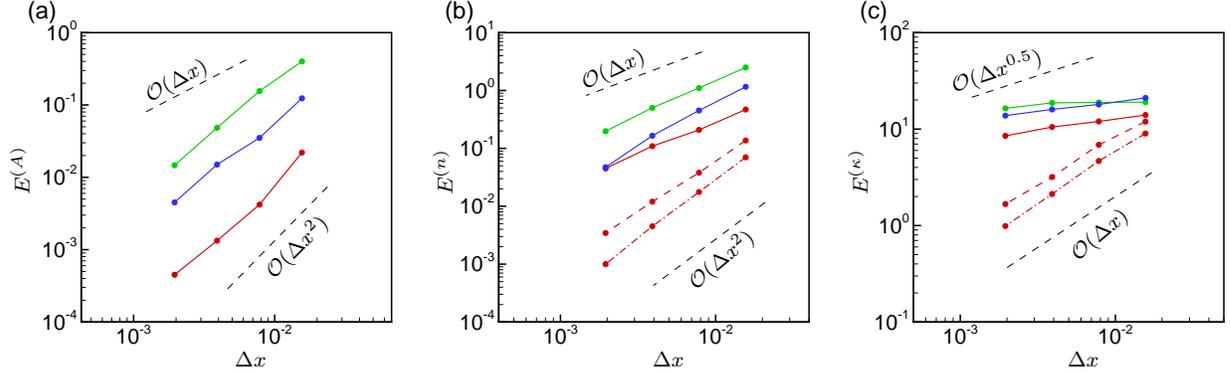


Figure 2.9: Convergence of area and normal and curvature errors at the interface for $t = 1$ and $T = 2$. Colors indicate, methods IA (red), CR (green) and RC (blue). For IA, solid, dashed and dashed-dot lines represent P_1 , P_2 and P_3 projections, respectively.

directions normal to the interface. The normal vector is defined as, $\mathbf{n} = \nabla \hat{\psi} / |\nabla \hat{\psi}|$. Equation (2.25) is iteratively solved to a steady state over the fictitious time τ . At that point, ψ would have been transformed back to an indicator function of approximately constant width around the interface, $\psi = 1/2$.

The spatial discretization of Eqs. (2.17), (2.18) and (2.24) is based on the standard, staggered grid form where a scalar function is defined at the cell center and the velocity components are considered at the center of cell faces. The semi-discrete form of Eq. (2.17) can be expressed, for example for f in 2-D, as

$$f_t|_{i,j} = -\frac{1}{\Delta x} \left(fu|_{i+1/2,j} - fu|_{i-1/2,j} \right) - \frac{1}{\Delta y} \left(fv|_{i,j+1/2} - fv|_{i,j-1/2} \right), \quad (2.26)$$

where u and v are the velocity components in the x - and y -directions, respectively. The velocity components at respective cell faces, e.g., at $(i + 1/2, j)$, $(i, j + 1/2)$, etc., can be obtained by Eqs. (2.8) and (2.9). This velocity field is divergence free

with respect to the spatial discretization in Eq. (2.26) and satisfies

$$\frac{1}{\Delta x} (u_{i+1/2,j} - u_{i-1/2,j}) + \frac{1}{\Delta y} (v_{i,j+1/2} - v_{i,j-1/2}) = 0 , \quad (2.27)$$

The numerical approximation of f at the cell faces, $(i+1/2, j)$, $(i, j+1/2)$, etc., are obtained from the cell center values by a 5th order WENO scheme. Equation (2.17) is advanced in time with an explicit, 2nd order TVD Runge Kutta scheme.

Interface profiles associated with the three schemes, IA, RC and CR are shown in Figs. 2.6 and 2.7. The discrete interface Γ_D is obtained by interpolation to $\phi(\mathbf{x}, t) = 0$ in the case of IA and CR and to $\psi(\mathbf{x}, t) = 0.5$ for RC. The discrete interface in each case is compared at $t = 1$ and $T = 2$ with the corresponding particle based interface Γ which is advected with Lagrangian particle tracking as defined in section 2.2.1. Figures 2.6 shows that Γ_D based on IA is able to track Γ with much greater precision compared with both the RC and CR schemes. While the CR is smooth, it suffers from significant area loss as a result of reinitialization based on Eq. (2.19). The RC scheme leads to relatively more accurate area conservation but breaks down into fragments in thinner regions. Figures 2.7 shows a similar behavior on a finer grid when the interface is advected for a longer period of time. Again, while the RC scheme appears to be able to stretch the interface about the same amount as the IA scheme, thinner regions again undergo fragmentation. The classical reinitialization, CR, scheme results, as before, in a more significant area loss.

An improved version of the RC scheme of Olsson et al. [51] is developed by Desjardins et al. [23] who propose to compute smoother gradients within the

transition zone on the basis of a signed distance function that is derived from their original indicator function. Their results show improvement over the original RC method. Compared to our results plotted in Fig. 2.8 at $t = 4$ and $t = 8$ for $T = 8$ based on the IA scheme, the solution in Fig. 11 of Desjardins et al. [23] still shows considerable area loss and distortion of the interface. Figures 2.6, 2.7 and 2.8 thus clearly demonstrate the consequence of attempts to modify the original interface with reinitialization that result in substantial loss of area and interface distortion. The IA scheme on the other hand provides the most accurate solution because the interface evolves free of reinitialization errors.

We next determine the accuracy of various schemes with respect to interface topology. We consider the case, $t = 1$ and $T = 2$ for which the RC scheme does not fragment and the CR scheme also does relatively better with respect to area conservation. The interface corresponding to this case is shown in Fig. 2.5. Errors related to area conservation $E^{(A)}$, interface normal vector $E^{(n)}$ and interface curvature $E^{(\kappa)}$ are plotted in Fig. (2.9) as a function of the grid size. Area conservation error is determined according to

$$E^{(A)} = \frac{1}{A_0} |A - A_0| , \quad (2.28)$$

where A_0 is the initial area and A is the area at any other time given by

$$A = \frac{1}{NM} \sum_{i,j} H_{i,j} , \quad H_{i,j} = \begin{cases} H_{i,j} = 0 & , \quad g_{i,j} \geq \mu \\ H_{i,j} = 1 & , \quad g_{i,j} < \mu \end{cases} . \quad (2.29)$$

where N and M are the number of grid points in the x and y directions. The symbol g stands for the level set function in the case of IA, the signed distance function

for CR and the indicator function when referring to the RC scheme. For the IA and CR schemes, $\mu = 0$ and is $1/2$ for the RC scheme. The interface normal and curvature errors are computed at the interface by interpolating respective normal and curvature grid-values and then comparing with the Lagrangian solution with respect to the arc length coordinate, exactly as for the errors plotted in Fig. 2.5, computed with Eq. (2.16).

The convergence of various schemes with respect to the errors at the interface is plotted in Fig. 2.9 on four different grids of size, 65, 129, 257 and 513. Area errors in Fig. 2.9(a) decay uniformly with approximately 2nd order accuracy for all schemes. The IA scheme leads to smallest error magnitudes. In the case of $E^{(n)}$, P₁ and the RC schemes lead to similar errors that decay with about 1st order accuracy. Errors related to P₂ and P₃ projections are substantially lower and converge with 2nd order accuracy. Note that the volume error plotted in Fig. 2.9(a) for the IA scheme is independent of the projection scheme and is therefore represented by just one curve. Figure 2.9(c) shows that curvature errors are the most difficult to converge and only P₂ and P₃ projections converge with 1st order accuracy. Both the RC and CR schemes perform quite poorly with respect to curvature.

To determine convergence with respect to topology on the grid, we compare with the exact solution for the circular interface at $t = 2$ for $T = 2$, i.e., after one full cycle of rotation. Figure 2.10 plots the interface based on the three advection schemes for three different grid sizes. The exact solution is indicated with the dashed line. On the 65×65 grid, the CR solution cannot make it to this time due to volume loss. The RC solution is substantially inaccurate while the IA solution is relatively

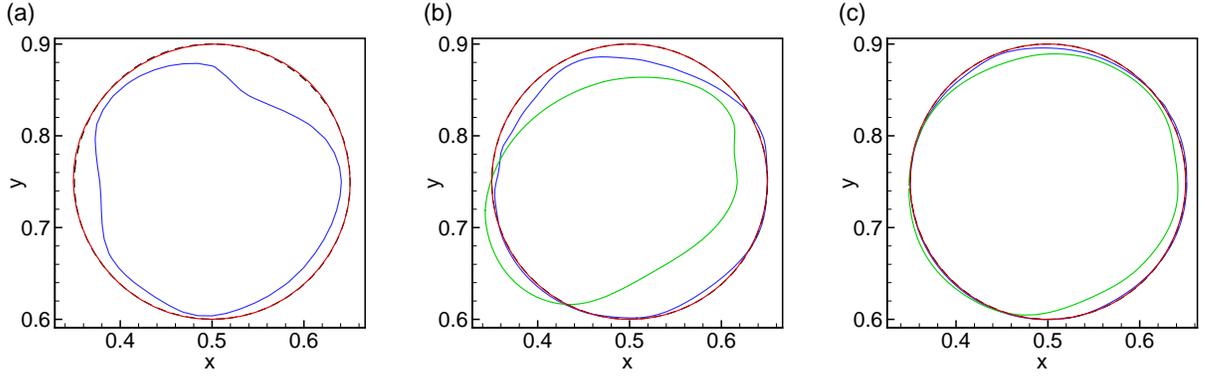


Figure 2.10: Interface profiles on coarse and fine grids after one full period of rotation for $T = 2$. (a) 65×65 , (b) 129×129 and (c) 257×257 . The dashed line depicts the smooth solution and solid lines indicates solution based on IA (red), CR (green) and RC (blue). RC solution cannot make it to $t = 2$ in (a) due to area loss.

well behaved. Both the RC and CR solutions improve upon grid refinement but the IA solution appears to do better. Topology errors on the grid for this case can be compared with the exact solution given by Eq. (2.13). Table 4 lists the errors for the distance function, the interface normal and the curvature computed with Eq. (2.15) for the three projection schemes. Errors decays uniformly in all cases expect for the curvature error in the case of P_1 projection. The P_3 projection scheme leads to 2nd order accuracy for $|\nabla\phi|$ and \hat{n}_x while the curvature error decays with 1st order accuracy. Note that although P_2 projection is exactly first order accurate with respect to curvature, it does reasonable well all, particularly for both the distance function and the normal. The comparison of errors in Fig. 2.9 and Table 4 thus clearly demonstrates that the method of geometric projection provides the most accurate representation of interface topology.

Table 2.4:

Grid errors at $t = T = 2$ with respect to the exact solution. The sign and the digit on the left indicate the exponent of, $\times 10$. The last row indicates the corresponding convergence rate.

Δx	$1 - \nabla\phi $			\hat{n}_x		
	P ₁	P ₂	P ₃	P ₁	P ₂	P ₃
1/32	1.57-2	9.03-3	7.45-3	7.81-2	4.45-2	2.24-2
1/64	7.56-3	3.22-3	1.77-3	4.34-2	1.56-2	5.41-2
1/128	3.59-3	1.18-3	4.67-4	2.26-2	3.51-3	1.35-3
1/256	1.76-3	8.45-4	1.23-4	1.05-2	9.67-4	3.24-4
	1.05	1.55	1.98	0.96	1.84	2.04

κ		
P ₁	P ₂	P ₃
8.27-1	4.83-1	7.64-2
7.99-1	2.69-1	3.70-2
7.88-1	1.72-1	1.80-2
7.78-1	8.78-2	8.77-3
0.02	0.81	1.04

Finally, we wish to determine how well does the combination of IA and geometric projection performs with respect to computational efficiency. Table 5 lists the run times for various grid sizes scaled with the time for the P₁ solution on the

Table 2.5:

Run times for projection, CR and RC schemes scaled with the time for P_1 on the smallest grid.

Δx	P_1	P_2	P_3	CR	RC
1/64	1.0	2.21	3.10	4.31	3.96
1/128	4.23	10.34	15.84	20.3	21.34
1/256	18.23	32.87	65.31	90.25	85.49
1/512	110.21	120.67	290.32	400.22	390.80

smallest grid. Since the algorithms perform extra work in computing the error, which would not be present in normal computations, a scaled run time is reported where all algorithms perform exactly the same error calculations on the same hardware. The P_1 scheme is the most efficient while CR is the least efficient. Other projection schemes also perform better than CR and RC. In computing the run times P_1 , CR and RC have been solved on the entire grid and reinitialization has been carried out at each time step. For P_2 and P_3 the projection has been limited to a distance of $|\phi| < 4\Delta x$. While similar efficiency enhancement strategies can also be adopted for CR and RC schemes, the objective here is to show that although P_1 , CR and RC are comparable with respect to the accuracy, P_1 is the most efficient and that the efficiency of P_2 and P_3 can be improved by restricting projection to the neighborhood of the interface where it is needed.

2.3 Two phase incompressible flow

2.3.1 Numerical method

We study the effectiveness of reinitialization with geometric projection for two-phase incompressible flow. Following the standard approach for solving the velocity field in two phase flow, the fluid properties are taken to transition either smoothly from one phase to the other across a region of finite width, or in a sharp manner across the interface, according to

$$\rho(\phi) = (\rho_1 - \rho_2)I + \rho_2 , \quad \mu(\phi) = (\mu_1 - \mu_2)I + \mu_2 , \quad (2.30)$$

where ρ and μ indicate density and viscosity, respectively, I is the transition function and subscripts, 1 and 2, indicate the phase. The transition function depends on the instantaneous signed distance function and is given by

$$I(\phi) = \frac{1}{2} [1 - \text{erf}(\phi/\varepsilon)] \quad \text{and} \quad I(\phi) = \phi^\pm / (\phi^+ - \phi^-) , \quad (2.31)$$

for a smooth and a sharp transition across the interface, respectively. In the case of smooth transition, ε determines the width of the interface. In the latter case, the superscripts indicate, in the denominator, the sign of the signed distance function across the interface and the position of the fluids with respect to the interface in the numerator. The condition of incompressibility in each phase implies,

$$\rho_t + (\mathbf{u} \cdot \nabla)\rho = 0 , \quad \mu_t + (\mathbf{u} \cdot \nabla)\mu = 0 , \quad (2.32)$$

The velocity \mathbf{u} is obtained from the following set of equations for incompressible flow

$$\nabla \cdot \mathbf{u} = 0 \quad (2.33)$$

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \nabla \cdot \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \rho g \hat{\mathbf{e}}_g \quad (2.34)$$

where p is the pressure, g the gravitational constant and $\hat{\mathbf{e}}_g$ is the unit vector aligned with the direction of gravity. The jump in pressure across the interface is given by

$$[p] = \sigma \kappa - [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \cdot \hat{\mathbf{n}}] \cdot \hat{\mathbf{n}} , \quad (2.35)$$

where $[\cdot]$ indicates the difference in the quantity across the sharp interface and $\hat{\mathbf{n}}$ is the unit normal vector at the interface that is directed from phase 1 towards phase 2. The first term on the right hand side represents the contribution to the jump in pressure due to surface tension σ and curvature κ . The second term on the right hand side indicates the jump in normal viscous stress due to the jump in viscosity across the interface. By taking viscosity to vary smoothly across the interface, the jump in normal stress can be avoided. The implementation of the jump in pressure given by Eq. (2.35) is thereby considerably simplified. The discretization of Eqs. 2.33 and 2.34 is carried out on a staggered grid. Spatial derivatives are treated with 2nd order central finite differences and the Pressure is obtained by Poisson equation resulting from the standard projection method. Details regarding our solution procedure are provided in Delaney et al. [21].

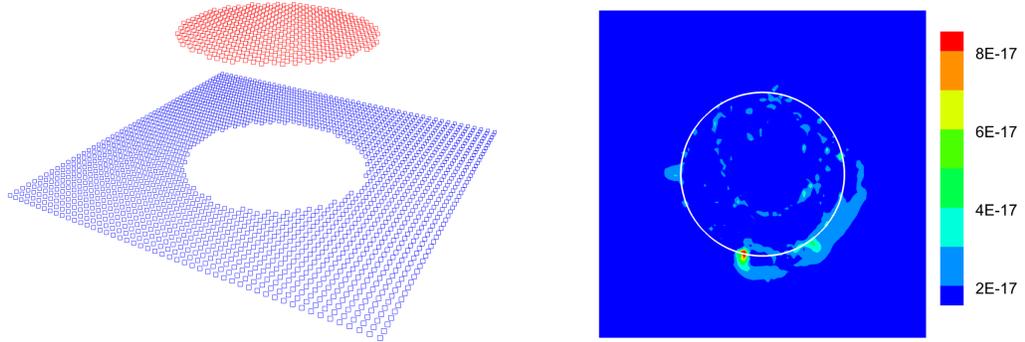


Figure 2.11: Pressure (left) and velocity (right) solutions for a circular droplet in vacuum for $We = 1$ reproduce the exact solution with machine precision when the exact values of curvature is used.

2.3.2 Validation

In order to validate the accuracy of interface topology obtained with the method of geometric projection we begin by considering the case of curvature driven motion of a droplet in vacuum for which analytical solutions are available. The problem is governed by two dependent dimensionless parameters,

$$Re = \rho UL/\mu \quad \text{and} \quad We = \rho U^2 L/\sigma \quad (2.36)$$

where Re and We are the Reynolds number and the Weber number respectively. In the absence of any external velocity scale, the specification of U couples the two parameter. We take $We = 1$ to determine the scale for U , which gives $Re = \sqrt{\rho L\sigma/\mu}$. The length scale is the droplet radius.

Table 2.6:

Error in the velocity field measured with the L^2 norm with respect to the exact solution for the circular interface with pressure jump. The curvature is computed with three projection schemes.

Δx	P ₁	P ₂	P ₃
1/100	8.36-4	1.09-4	6.26-5
1/200	5.71-4	5.92-5	2.25-5
1/400	3.33-3	2.42-5	9.44-6

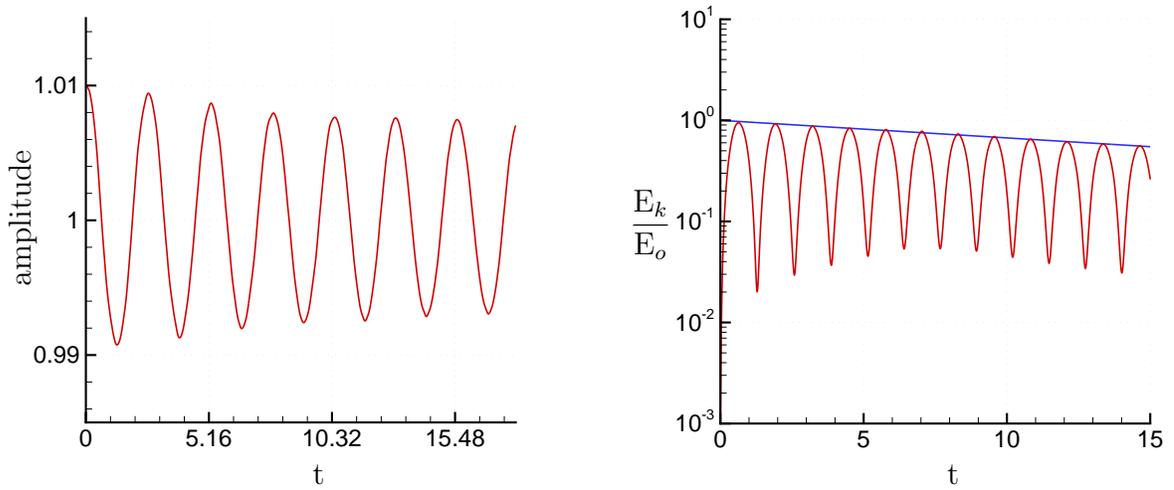


Figure 2.12: (a)Amplitude oscillation of the interface and (b) the kinetic energy for density and viscosity ratio of 10^3 , $We = 1$ and $Re = 100$. Solid blue line in (b) indicates the analytical solution.

In the case of a circular interface, the exact solution is a stationary fluid that is independent of Re for which, $[p] = 1/We$ and $\mathbf{u} = 0$. The numerical solution

is obtained for zero pressure outside the droplet with $We = 1$. The interface is represented by the exact signed distance function, $\tilde{\phi}$, that is defined in Eq. (2.12). For the dimensionless problem, the exact value of the curvature is $\kappa = 1$. This exact value is used first for the interfacial jump condition in Eq. (2.35) to show that the numerical method for the solution of the Poisson equation with discontinuous coefficients resolves the exact solution with machine precision. Figure 2.11 plots these numerical solutions for pressure $p = 1$ and velocity $|\mathbf{u}| = 0$ with an error of less than 10^{-15} .

We next compute the curvature on the basis of the signed distance function obtained with projections P_i . Table 3 shows that the grid convergence of the curvature is 1st order accurate in the case of a circular interface related to the exact signed distance function. Table 5 lists the errors in velocity magnitude, $|\mathbf{u}|$, measured with an L^2 norm in comparison with the exact solution. The curvature is related to the signed distance function obtained by the projection of the interface associated with $\tilde{\phi}$ based on the three P_i projections for different grid sizes. The error decreases substantially from P_1 to P_2 for a fixed grid size but the reduction is relatively less from P_2 to P_3 . The error however decreases with 1st order accuracy with grid refinement for both the P_2 and P_3 schemes. While a smooth transition of density over a fixed interface width $\varepsilon = 1.5\Delta x$ has been used for these calculations, as indicated by Eq. (4.8), we have found that the solution based on the sharp density jump leads to a similar convergence behavior with only a slight increase in error magnitudes.

To determine the accuracy of resolving curvature driven motion, we consider an initially elliptical interface. The competition between viscous, pressure and capillary

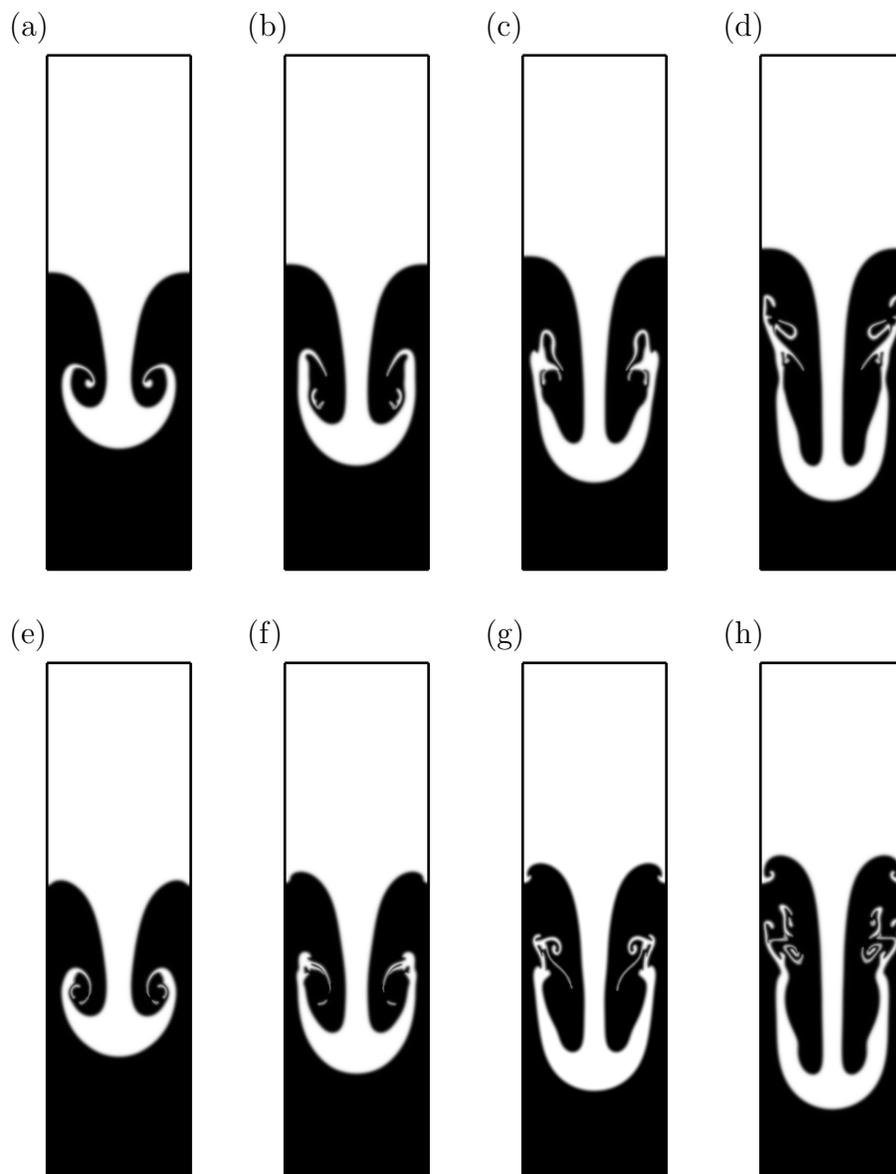


Figure 2.13: Interface evolution for 2-D unstable Rayleigh-Taylor problem, obtained on a 200×800 grid at $t = 1.75, 2.0, 2.25$ and 2.5 . (a)-(d) $Re = 3000$ and (e)-(h) $Re = 6000$.

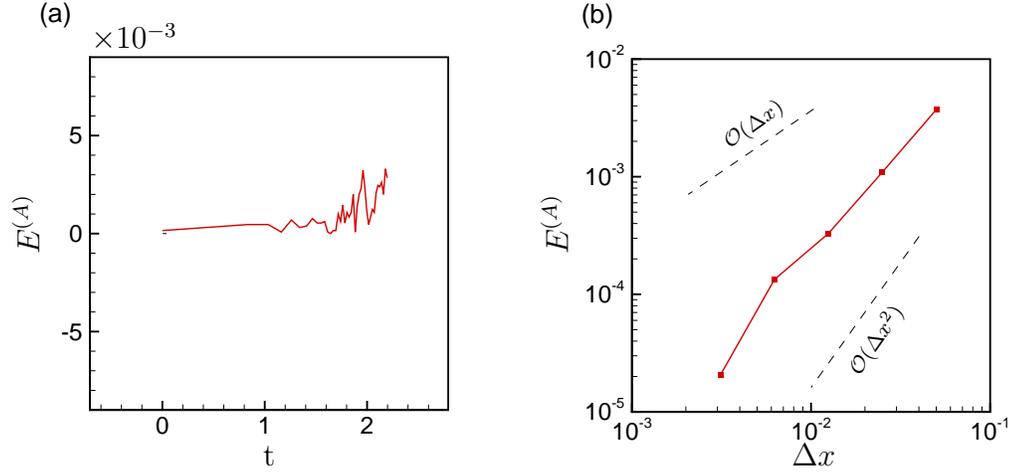


Figure 2.14: Accuracy of numerical simulation of Rayleigh-Taylor instability for $Re = 3000$ and $At = 0.5$. (a) Volume conservation error as a function of time on a 100×400 grid, (b) error on different sized grids at $t = 1$. Lines indicate reinitialization based on geometric projection (red) and recompression (green). The error is smaller in the case of geometric projection and decays with 2nd order accuracy.

forces drives the interface toward a circular equilibrium profile through damped periodic oscillations. The period of oscillations, τ , is a function only of We and is given by the analytic solution, $\tau = 2\pi\sqrt{We/6}$, for a small stretching amplitude. The amplitude of oscillations is governed by the total energy

$$E = \frac{1}{2} \int |\mathbf{u}|^2 dA + \frac{1}{We} \int d\ell \quad (2.37)$$

where the first term on the right is the kinetic energy E_k and the second term represents interfacial energy over the perimeter ℓ of the interface. For small stretching amplitudes the total energy decays according to the analytical solution

$$E = E_0 \exp(-4\sqrt{We}/Re) \quad (2.38)$$

where E_0 is the initial energy. The initial condition is the quiescent state, $\mathbf{u} = 0$ and

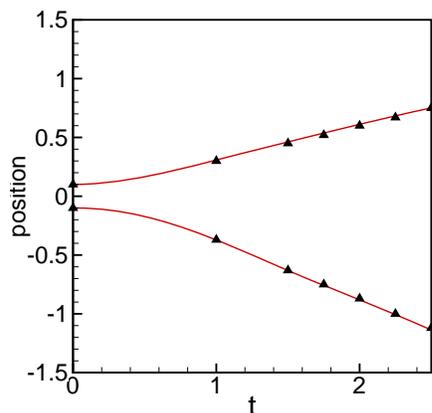


Figure 2.15: Position of the most advanced portions of the rising and falling fluids as a function of time during the evolution of Rayleigh-Taylor instability for $Re = 3000$, $We = \infty$ and $At = 0.5$. Solid lines indicates the solution obtained with geometric projection. Triangular symbols represent numerical results of [29].

hence E_o is purely interfacial. The length scale for this problem is the equilibrium radius and the initial condition for the interface is $x^2/a^2 + a^2y^2 = 1$ in the domain $[-1.5, 1.5] \times [-1.5, 1.5]$ with 200×200 grid points. The stretching amplitude a is the ratio of the major axis to equilibrium radius. The density and viscosity ratio is 10^3 with the heavier and denser fluid inside the interface and $Re = 100$. Oscillation of the interface about the equilibrium position and the associated kinetic energy for an initial stretching amplitude, $a = 1.01$, is plotted in Fig. 2.12. The period of oscillation, $T = 2.58$, matches the analytical value of 2.565 quite well. The kinetic energy E_k/E_o is also observed to be in good agreement with the analytical solution given by Eq. (4.40). Numerical solution in this case is based on P_2 projection.

We next consider the 2-D unstable Rayleigh-Taylor problem, shown in Fig,

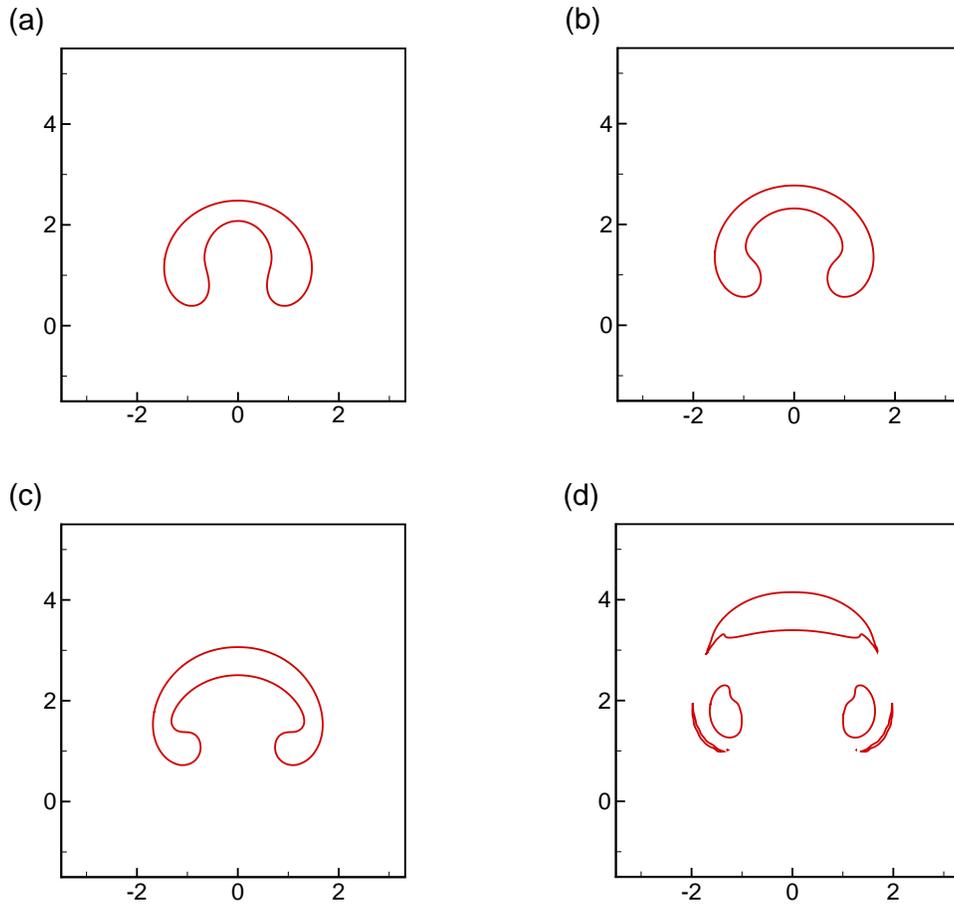


Figure 2.16: Interface profiles for the rising bubble problem on a 240×240 grid for $Re = 100$, $We = 200$ and $Fr = 1$. at (a) $t = 3.0$; (b) $t = 3.5$; (c) $t = 4.0$; (d) $t = 6.0$.

2.13, without surface tension. The physical behavior is governed by the competition between only gravitational and viscous effects. High accuracy projection is hence not needed in this case for the calculation of the curvature but P_1 projection is used to prevent the level set function from developing discontinuities close to the interface and for specifying the interfacial transition zone. Our solution for this problem demonstrates that more expensive indicator function based methods (see

Table 5) can be avoided when surface tension is small or negligible with the use of level set advection with just P_1 projection without the need to specify interface diffusion parameters.

The initial interface profile is taken as, $y(x) = (2 + 0.1\cos(2\pi x))$, in the rectangular domain $[0, 1] \times [0, 4]$. Our computations are based on the Reynolds number, $Re = \rho_1 g^{1/2} L^{3/2} / \mu_1$, the Weber number $We = \rho_1 g L^2 / \sigma = \infty$ and the Froude number $Fr = 1$. The density difference is represented by the Atwood number $At = (\rho_1 - \rho_2) / (\rho_1 + \rho_2) = 0.5$ and the viscosity ratio is 1. Results shown in Fig. 2.13 are obtained on a 200×800 grid for $Re = 3000$ and $Re = 6000$ with the width of the transition region, $\varepsilon = 3\Delta x$. These values were chosen to compare with the solutions of [25] and [69], which use comparable interface widths.

The interface profiles for $Re = 6000$, shown in Figs. 2.13(e)-(h), contain more structure than for the $Re = 3000$ case in Figs. 2.13(a)-(d), particularly at later times. Secondary unstable structures start to form along the upward moving front $Re = 6000$. The fine scale details however appear at about the same thickness in both cases, which indicates the limit of resolution on the 200×800 grid. We draw the attention of the reader to Fig. 26 of [69] Fig. 6 of [25] which show almost identical interface profiles for $Re = 3000$ based on the recompression and the Cahn-Hilliard methods, respectively. Area conservation errors associated with the numerical solution of the Rayleigh-Taylor problem are shown in Fig. 2.14 for $Re = 3000$, $We = \infty$ and $At = 0.5$. The error $E^{(A)}$, shown in Fig. 2.14(a), is relatively small throughout and the convergence rate measured at $t = 1$ decays with 2nd order accuracy as indicated in Fig. 2.14(b). The position of the upper and lower

fronts shows good agreement with previous numerical solutions of [29] and [25] which indicates that the gross behavior is predicted with relative ease for this problem but the resolution of fine scale interfacial structure requires more sophisticated methods of interface advection.

We finally consider the rising bubble problem where viscous, gravitational and interfacial forces are in competition. This problem also demonstrates the ability of projection schemes to capture interface breakup events. Figure 4.5 shows the solution on a 240×240 grid for $Re = 100$, $We = 200$, $Fr = 1$, $\mu_1/\mu_2 = 100$, $\rho_1/\rho_2 = 100$ and $\varepsilon = 3\Delta x$ at four different times. Fluid properties of the bubble are used as characteristic values of density and viscosity. The characteristic length is the bubble diameter. The breakup of the bubble is captured well with the projection schemes.

2.4 Conclusion

The current study makes two main contributions. It provides a general and accurate method for the construction of the signed distance function. On the basis of this approach it demonstrates that reinitialization when decoupled from advection leads to greater volume conservation and better topology representation on the Cartesian grid. The new method for the construction of the signed distance function is based on geometric projection of interface topology. This approach represents a generalized methodology both for the reinitialization of scalar functions associated with implicit interfaces as well as for interfaces represented by a set of marker particles.

In the case of implicit interfaces defined by scalar fields, the signed distance function need not be advected, as required by conventional methods. Reinitialization errors thus do not degrade the advection of the underlying scalar function. The resulting improvement in area conservation as well as the accuracy of resolving interface normals and curvature on the grid has been shown to be substantial in comparison with existing schemes. The new approach is also shown to be computationally more efficient in comparison with the existing methods. Application of geometric projection to both kinematic and two phase flow problems suggests that the improvement in resolving interface topology is large for P_2 projection with respect to P_1 , but is less substantial in comparison with the more expansive P_3 method. Hence, P_2 projection would be more suitable for the computation of large 3-D flow problems. In the case of two phase flow with small interfacial tension the use of the very efficient P_1 projection would be sufficient. The decoupling of advection and reinitialization opens up the possibility of further improvements that are focused on each of the aspects individually.

Chapter 3: Second Order Accurate Boundary Capture Method

3.1 Overview

For the projection method, the accuracy of the pressure and velocity calculation depends on how the Poisson equation is solved in the presence of jump discontinuities. In this Chapter, we develop a new method to solve the variable coefficient Poisson equation with discontinuous jumps in the solution and its derivatives. Instead of separating the two fluids with a sharp interface within a grid cell, the new method considers their average value that is weighted by respective volume fractions. A correction term is then used for implementing the jump conditions with 2nd order accuracy. Similar to the method proposed in [79], the new method is implemented using a standard finite difference discretization on a Cartesian grid. Therefore, it can easily be extended to three spatial dimensions for both steady and unsteady interface. Furthermore, the coefficient matrix obtained using the linear Poisson's equation is symmetric. Compared with previous methods, this new method is a robust second order scheme. For some cases, the accuracy of this method can reach even higher than second order. It is also free of numerical smearing and preserve discontinuities very well. Furthermore, our method can be used to solve Navier-Stokes equation without the need of additional sources.

In the following, we describe the Poisson equation for the volume fraction weighted average of the discontinuous variable in section 3.2. The discretization schemes incorporating both Dirichlet and Neumann jump conditions are presented in section 3.3. Validation studies are presented in section 3.4.

3.2 Two-phase Poisson equation

The following Poisson eqs. for two variables, P_1 and P_2 , need to be solved numerically on a uniform Cartesian grid,

$$\nabla \cdot (\lambda_1 \nabla P_1) = f_1 , \quad (3.1)$$

$$\nabla \cdot (\lambda_2 \nabla P_2) = f_2 . \quad (3.2)$$

with appropriate boundary conditions and the condition that P_1 and P_2 exist in spatially distinct regions separated by an interface. Coefficients, λ_1 and λ_2 are variable in general. Across the interface P_1 and P_2 are related by

$$P_{1_I} - P_{2_I} = \alpha \quad \text{and} \quad ([\lambda_1 \nabla P_1]_I - [\lambda_2 \nabla P_2]_I) \cdot \mathbf{n} = \beta , \quad (3.3)$$

where α and β are constants, subscript I refers to interfacial values and \mathbf{n} is the unit vector normal to the interface, directed from P_1 to P_2 . The interface divides the computational cell into two parts with volume fractions ϕ_1 and ϕ_2 that are related by

$$\phi_1 + \phi_2 = 1 . \quad (3.4)$$

Equations (3.1) and (3.2) can be discretized individually in a standard manner in respective regions away from the cells containing the interface. For interfacial cells

we assume that both P_1 and P_2 exist within the cell and their collective effect can be expressed by a volume fraction weighted average value,

$$\bar{P} = \phi_1 P_1 + \phi_2 P_2 . \quad (3.5)$$

as illustrated in Fig. 3.1. We now propose that the governing equation for \bar{P} is of the form,

$$\nabla \cdot (\lambda \nabla \bar{P}) = \phi_1 f_1 + \phi_2 f_2 = f . \quad (3.6)$$

The definition of λ will become explicit during the discretization step. Note that Equation (3.6) is not simply an addition of Equations (3.1) and (3.2), but reduces to the appropriate eq. in cells that do not contain the interface for respective values of $\phi_1 = 1, \phi_2 = 0$ and $\phi_1 = 0, \phi_2 = 1$. Within the interfacial cell, Eq. (3.6) is a volume average weighted model that will be shown below to facilitate high accuracy solutions of P_1 and P_2 .

3.3 Numerical Method

3.3.1 One Dimension

The discretization of Equation (3.6) will be illustrated first for a 1-D problem. For the interfacial cell centered at position i , and with right and left cell faces at $i = i + 1/2$ and $i = i - 1/2$, respectively, the discrete form of Equation (3.6) can be expressed as,

$$\left[\frac{d}{dx} \left(\lambda \frac{d\bar{P}}{dx} \right) \right]_i = [\phi_1 f_1]_i + [\phi_2 f_2]_i + r_i , \quad (3.7)$$

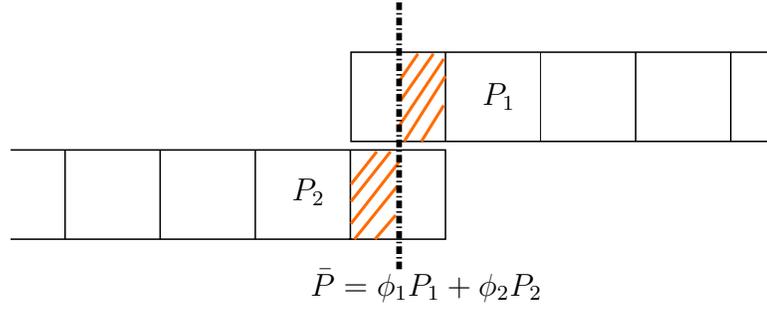


Figure 3.1: Jump condition in one dimension. Volume fraction ϕ_1 and ϕ_2 are used to weight discontinuous variable P .

where r_i is the correction needed to achieve a specified formal accuracy with respect to the truncation error. A conservative 2nd order accurate discretization of the first derivative in Equation (3.7) on a uniform grid with spacing h gives

$$\lambda_{i+1/2}(\bar{P}_{i+1} - \bar{P}_i) - \lambda_{i-1/2}(\bar{P}_i - \bar{P}_{i-1}) = h^2 \phi_{1i} f_{1i} + h^2 \phi_{2i} f_{2i} + h^2 r_i + O(h^4), \quad (3.8)$$

which is the primary governing equation for the nodal unknown \bar{P} . Using the definition of \bar{P} given by Equation (3.5) in Equation (3.8) yields

$$\begin{aligned} & \lambda_{i+1/2}[(\phi_{1i+1} P_{1i+1} - \phi_{1i} P_{1i}) + (\phi_{2i+1} P_{2i+1} - \phi_{2i} P_{2i})] \\ & - \lambda_{i-1/2}[(\phi_{1i} P_{1i} - \phi_{1i-1} P_{1i-1}) + (\phi_{2i} P_{2i} - \phi_{2i-1} P_{2i-1})] \\ & = h^2 \phi_{1i} f_{1i} + h^2 \phi_{2i} f_{2i} + h^2 r_i + O(h^4). \end{aligned} \quad (3.9)$$

The interfacial jump conditions in Equation (3.3) relate P_1 to P_2 . Because the jump is known only at the interface, we convert the jump to a volume averaged quantity at the cell center by integrating Equation (3.3) over a computational cell i of volume V_i ,

$$\frac{1}{V_i} \int_{V_i} P_1 dV - \frac{1}{V_i} \int_{V_i} P_2 dV = \frac{1}{V_i} \int_{V_i} \alpha dV \quad (3.10)$$

It can be shown that the cell center values P_{1_i} , P_{2_i} and α_i are second order approximations to the integrals above. Hence,

$$P_{1_i} - P_{2_i} = \alpha_i + O(h^2) \quad (3.11)$$

We now wish to determine the correction r_i in Equation (3.8). The discrete form of Equation (3.1) and (3.2),

$$h^2 \phi_{1_i} f_{1_i} = \lambda_{i+1/2} [(\phi_{1_i} P_{1_{i+1}} - \phi_{1_i} P_{1_i}) + (\phi_{1_i} P_{2_{i+1}} - \phi_{1_i} P_{2_i})] , \quad (3.12)$$

$$h^2 \phi_{2_i} f_{2_i} = \lambda_{i+1/2} [(\phi_{2_i} P_{1_{i+1}} - \phi_{2_i} P_{1_i}) + (\phi_{2_i} P_{2_{i+1}} - \phi_{2_i} P_{2_i})] , \quad (3.13)$$

along with Equation (3.11) can be used in Equation (3.9) to yield,

$$r_i = \frac{1}{h^2} \left(\lambda_{i+1/2} (\phi_{1_{i+1}} - \phi_{1_i}) \alpha_{i+1} - \lambda_{i-1/2} (\phi_{1_i} - \phi_{1_{i-1}}) \alpha_{i-1} \right) + O(h^2) . \quad (3.14)$$

Thus, for each grid point i , one can write a linear equation of the form,

$$\frac{\lambda_{i+1/2} (\bar{P}_{i+1} - \bar{P}_i) - \lambda_{i-1/2} (\bar{P}_i - \bar{P}_{i-1})}{h^2} = \phi_{1_i} f_{1_i} + \phi_{2_i} f_{2_i} + F^R - F^L + O(h^2) , \quad (3.15)$$

where,

$$F^R = \frac{1}{h^2} \lambda_{i+1/2} (\phi_{1_{i+1}} - \phi_{1_i}) \alpha_{i+1} , \quad (3.16)$$

$$F^L = \frac{1}{h^2} \lambda_{i-1/2} (\phi_{1_i} - \phi_{1_{i-1}}) \alpha_{i-1} , \quad (3.17)$$

and assemble the system of linear equations for the unknowns \bar{P}_i to form the Poisson coefficient matrix that would be symmetric. Equation (3.15) would be influenced by the interface in cells $i - 1$, i and $i + 1$. When the interface passes through cell $i + 1$, $\phi_{1_i} = \phi_{1_{i-1}}$. Equations (3.16) and (3.17) is then simplified to,

$$F^R = \frac{1}{h^2} \lambda_{i+1/2} (\phi_{1_{i+1}} - \phi_{1_i}) \alpha_{i+1} , \quad (3.18)$$

$$F^L = 0 . \quad (3.19)$$

Similarly, for the case when the interface is in cell $i - 1$,

$$F^R = 0, \quad (3.20)$$

$$F^L = \frac{1}{h^2} \lambda_{i-1/2} (\phi_{1_i} - \phi_{1_{i-1}}) \alpha_{i-1}. \quad (3.21)$$

Note that α_{i+1} and α_{i-1} require the knowledge of jumps in cells adjacent to the interfacial cell. But if the interface passes through the cell i , the value jumps, α_{i+1} and α_{i-1} , could not be used directly. However, we can define the jump in the gradients of average values of P_1 and P_2 at cell faces to be,

$$\beta_{i+1/2} = \frac{1}{h} \lambda_{i+1/2} (\alpha_{i+1} - \alpha_i), \quad (3.22)$$

$$\beta_{i-1/2} = \frac{1}{h} \lambda_{i-1/2} (\alpha_i - \alpha_{i-1}). \quad (3.23)$$

Thus, we can express the jumps α in cell $i - 1$ and $i + 1$ by the jumps α in cell i and the gradients jumps across across the neighboring cell faces, $\beta_{i+1/2}$ and $\beta_{i-1/2}$. By this way, for the case where that interface passes through cell i , the correction terms can be expressed as,

$$F^R = \frac{1}{h^2} (\phi_{1_{i+1}} - \phi_{1_i}) (\beta_{i+1/2} h + \lambda_{i+1/2} \alpha_i), \quad (3.24)$$

$$F^L = \frac{1}{h^2} (\phi_{1_i} - \phi_{1_{i-1}}) (\lambda_{i-1/2} \alpha_i - \beta_{i-1/2} h). \quad (3.25)$$

The value of λ still needs to be specified. The common case is $\lambda = 1/\rho$, where ρ is the mixture density in the cell specified as $\rho_i = \phi_{i_1} \rho_{i_1} + \phi_{i_2} \rho_{i_2}$. Moreover, $\lambda_{i+1/2} = 1/\rho_{i+1/2}$ and a 2nd order approximation of $\rho_{i+1/2}$ is

$$\rho_{i+1/2} = \frac{1}{2} (\rho_{i+1} + \rho_i) + O(h^2). \quad (3.26)$$

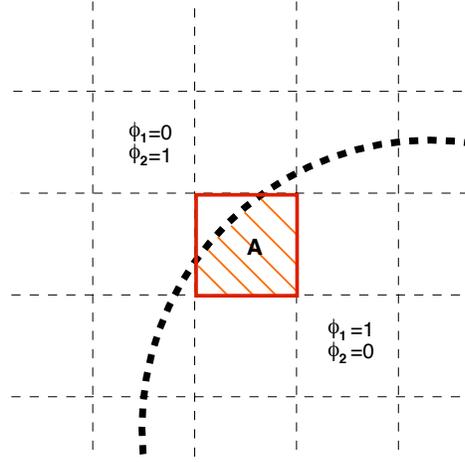


Figure 3.2: The Area fraction method used to obtain volume fraction for Two-Dimension case. A is the area inside of the interface and $\phi_{1,i,j} = \frac{A}{dxdy}$.

Therefore,

$$\lambda_{i+1/2} = \frac{2}{(\phi_{1,i+1}\rho_{1,i+1} + \phi_{2,i+1}\rho_{2,i+1}) + (\phi_{1,i}\rho_{1,i} + \phi_{2,i}\rho_{2,i})} . \quad (3.27)$$

3.3.2 Two Dimensions

Discrete form of Equation (3.6) can be expressed in a relatively simple manner for 2-D problem as,

$$\left[\frac{\partial}{\partial x} \left(\lambda \frac{\partial \bar{P}}{\partial x} \right) \right]_{i,j} + \left[\frac{\partial}{\partial y} \left(\lambda \frac{\partial \bar{P}}{\partial y} \right) \right]_{i,j} = [\phi_1 f_1]_{i,j} + [\phi_2 f_2]_{i,j} + r_{i,j} . \quad (3.28)$$

A 2nd order discretization of first derivatives at cell face locations, $(i + 1/2, j)$, $(i - 1/2, j)$, $(i, j + 1/2)$ and $(i, j - 1/2)$ gives,

$$\begin{aligned} & [\lambda_{i+1/2,j}(\bar{P}_{i+1,j} - \bar{P}_{i,j}) - \lambda_{i-1/2,j}(\bar{P}_{i,j} - \bar{P}_{i-1,j})] \\ & + [\lambda_{i,j+1/2}(\bar{P}_{i,j+1} - \bar{P}_{i,j}) - \lambda_{i,j-1/2}(\bar{P}_{i,j} - \bar{P}_{i,j-1})] \\ & = h^2 \phi_1 f_1 + h^2 \phi_2 f_2 + h^2 r_{i,j} + O(h^4) , \end{aligned} \quad (3.29)$$

where, by the similar way in Equation (3.27), the value of λ in x and y direction is specified as

$$\lambda_{i+1/2,j} = \frac{2}{(\phi_{1_{i+1,j}}\rho_{1_{i+1,j}} + \phi_{2_{i+1,j}}\rho_{2_{i+1,j}}) + (\phi_{1_{i,j}}\rho_{1_{i,j}} + \phi_{2_{i,j}}\rho_{2_{i,j}})}, \quad (3.30)$$

and

$$\lambda_{i,j+1/2} = \frac{2}{(\phi_{1_{i,j+1}}\rho_{1_{i,j+1}} + \phi_{2_{i,j+1}}\rho_{2_{i,j+1}}) + (\phi_{1_{i,j}}\rho_{1_{i,j}} + \phi_{2_{i,j}}\rho_{2_{i,j}})}, \quad (3.31)$$

We then use the two dimensional discrete form of Equation (3.1) and (3.2),

$$\begin{aligned} h^2\phi_1f_1 &= \lambda_{i+1/2,j}(\phi_{1_{i,j}}P_{1_{i+1,j}} - \phi_{1_{i,j}}P_{1_{i,j}}) - \lambda_{i-1/2,j}(\phi_{1_{i,j}}P_{1_{i,j}} - \phi_{1_{i,j}}P_{1_{i-1,j}}) \\ &+ \lambda_{i,j+1/2}(\phi_{1_{i,j}}P_{1_{i,j+1}} - \phi_{1_{i,j}}P_{1_{i,j}}) - \lambda_{i,j-1/2}(\phi_{1_{i,j}}P_{1_{i,j}} - \phi_{1_{i,j}}P_{1_{i,j-1}}), \end{aligned} \quad (3.32)$$

$$\begin{aligned} h^2\phi_2f_2 &= \lambda_{i+1/2,j}(\phi_{2_{i,j}}P_{2_{i+1,j}} - \phi_{2_{i,j}}P_{2_{i,j}}) - \lambda_{i-1/2,j}(\phi_{2_{i,j}}P_{2_{i,j}} - \phi_{2_{i,j}}P_{2_{i-1,j}}) \\ &+ \lambda_{i,j+1/2}(\phi_{2_{i,j}}P_{2_{i,j+1}} - \phi_{2_{i,j}}P_{2_{i,j}}) - \lambda_{i,j-1/2}(\phi_{2_{i,j}}P_{2_{i,j}} - \phi_{2_{i,j}}P_{2_{i,j-1}}), \end{aligned} \quad (3.33)$$

in Equation (3.29) to yield,

$$\begin{aligned} r_{i,j} &= \frac{1}{h^2} \left(\lambda_{i+1/2,j}(\phi_{1_{i+1,j}} - \phi_{1_{i,j}})\alpha_{i+1,j} - \lambda_{i-1/2,j}(\phi_{1_{i,j}} - \phi_{1_{i-1,j}})\alpha_{i-1,j} \right. \\ &\left. + \lambda_{i,j+1/2}(\phi_{1_{i,j+1}} - \phi_{1_{i,j}})\alpha_{i,j+1} - \lambda_{i,j-1/2}(\phi_{1_{i,j}} - \phi_{1_{i,j-1}})\alpha_{i,j-1} \right) + O(h^2). \end{aligned} \quad (3.34)$$

Therefore, in two dimensions, each grid point (i, j) is discretized as

$$\begin{aligned} &\frac{\lambda_{i+1/2,j}(P_{i+1,j} - P_{i,j}) - \lambda_{i-1/2,j}(P_{i,j} - P_{i-1,j})}{h^2} \\ &+ \frac{\lambda_{i,j+1/2}(P_{i,j+1} - P_{i,j}) - \lambda_{i,j-1/2}(P_{i,j} - P_{i,j-1})}{h^2} \\ &= \phi_1f_1 + \phi_2f_2 + F^R - F^L + F^T - F^B + O(h^2), \end{aligned} \quad (3.35)$$

where,

$$F^R = \frac{1}{h^2} \lambda_{i+1/2,j} (\phi_{1_{i+1,j}} - \phi_{1_{i,j}}) \alpha_{i+1,j} , \quad (3.36)$$

$$F^L = \frac{1}{h^2} \lambda_{i-1/2,j} (\phi_{1_{i,j}} - \phi_{1_{i-1,j}}) \alpha_{i-1,j} , \quad (3.37)$$

$$F^T = \frac{1}{h^2} \lambda_{i,j+1/2} (\phi_{1_{i,j+1}} - \phi_{1_{i,j}}) \alpha_{i,j+1} , \quad (3.38)$$

$$F^B = \frac{1}{h^2} \lambda_{i,j-1/2} (\phi_{1_{i,j}} - \phi_{1_{i,j-1}}) \alpha_{i,j-1} , \quad (3.39)$$

$$(3.40)$$

and included in the linear system of equation.

Similar to the 1-D case, the 2-D correction consists of the Dirichlet jump in the neighboring cells around the interfacial cell. However, this jump can only be used directly for interfacial cells where the interface passes through. For neighboring cells the Dirichlet jumps should be replaced by gradient jumps, as for the 1-D case discussed above. The gradients in directions normal and tangential to the interface can be obtained as,

$$P_n = P_x n^1 + P_y n^2 , \quad (3.41)$$

$$P_t = P_x n^2 - P_y n^1 , \quad (3.42)$$

where (n^1, n^2) are the two components of the unit vector normal to the interface.

The gradients P_x and P_y in terms of the normal and tangential gradients are

$$P_x = P_n n^1 + P_t n^2 , \quad (3.43)$$

$$P_y = P_n n^2 - P_t n^1 . \quad (3.44)$$

Premultiplying Equations (3.43) and (3.44) by λ and taking the jump across the

interface leads to

$$[\lambda P_x] = [\lambda P_n]n^1 + [\lambda P_t]n^2 , \quad (3.45)$$

$$[\lambda P_y] = [\lambda P_n]n^2 - [\lambda P_t]n^1 , \quad (3.46)$$

On the other hand, the jump in the x -direction gradient of P_1 and P_2 at cell faces can also be defined by,

$$\begin{aligned} [\lambda P_x]_{i+1/2,j} &= \frac{1}{h} \lambda_{i+1/2,j} ((P_{1_{i+1,j}} - P_{1_{i,j}}) - (P_{2_{i+1,j}} - P_{2_{i,j}})) = \\ & \frac{1}{h} \lambda_{i+1/2,j} (\alpha_{i+1,j} - \alpha_{i,j}) , \end{aligned} \quad (3.47)$$

$$\begin{aligned} [\lambda P_x]_{i-1/2,j} &= \frac{1}{h} \lambda_{i-1/2,j} ((P_{1_{i,j}} - P_{1_{i-1,j}}) - (P_{2_{i,j}} - P_{2_{i-1,j}})) = \\ & \frac{1}{h} \lambda_{i-1/2,j} (\alpha_{i,j} - \alpha_{i-1,j}) . \end{aligned} \quad (3.48)$$

The gradient jump in the y -direction is similarly,

$$[\lambda P_y]_{i,j+1/2} = \frac{1}{h} \lambda_{i,j+1/2} (\alpha_{i,j+1} - \alpha_{i,j}) , \quad (3.49)$$

$$[\lambda P_y]_{i,j-1/2} = \frac{1}{h} \lambda_{i,j-1/2} (\alpha_{i,j} - \alpha_{i,j-1}) . \quad (3.50)$$

The left hand side terms of Equations (3.47), (3.48), (3.49) and (3.50) are decided by Equations (3.45) and (3.46). Therefore, if the interface does not pass through cell $(i+1, j)$, the jump $\alpha_{i+1,j}$ can be expressed by

$$\alpha_{i+1,j} = \frac{[\lambda P_x]_{i+1/2,j} h}{\lambda_{i+1/2,j}} + \alpha_{i,j} , \quad (3.51)$$

and the term F^R in Equation (3.36) should be replaced by

$$F^R = \frac{1}{h^2} (\phi_{1_{i+1,j}} - \phi_{1_{i,j}}) ([\lambda P_x]_{i+1/2,j} h + \lambda_{i+1/2,j} \alpha_{i,j}) , \quad (3.52)$$

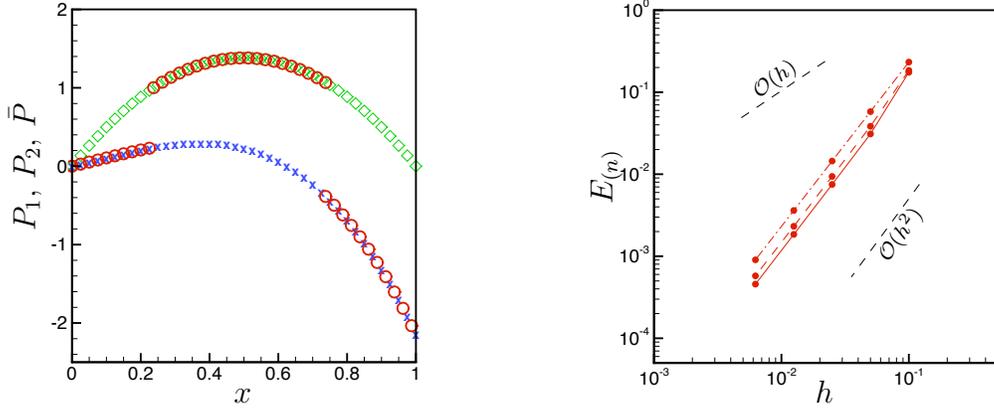


Figure 3.3: (left) Exact and computed solutions. P_1 (green diamond), P_2 (blue cross), and \bar{P} (red cycle). (right) Error vs. grid spacing h . L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line)

because $\alpha_{i+1,j}$ can not be used directly here. It's possible that interface may not pass through cell (i, j) either, but the coefficient $(\phi_{1_{i+1,j}} - \phi_{1_{i,j}})$ will be equal to zero at the same time, therefore, the value of $\alpha_{i,j}$ will not affect the computational solution in this situation. Similarly, the term F^L should be replaced by the following equation if the interface does not pass through cell $(i - 1, j)$,

$$F^L = \frac{1}{h^2}(\phi_{1_{i,j}} - \phi_{1_{i-1,j}})(-[\lambda P_x]_{i-1/2,j}h + \lambda_{i-1/2,j}\alpha_{i,j}) . \quad (3.53)$$

For the term F^T , as get

$$F^T = \frac{1}{h^2}(\phi_{1_{i,j+1}} - \phi_{1_{i,j}})([\lambda P_y]_{i,j+1/2}h + \lambda_{i,j+1/2}\alpha_{i,j}) , \quad (3.54)$$

When the interface does not pass through cell $(i, j - 1)$, F_B is replaced by

$$F^B = \frac{1}{h^2}(\phi_{1_{i,j}} - \phi_{1_{i,j-1}})(-[\lambda P_y]_{i,j-1/2}h + \lambda_{i,j-1/2}\alpha_{i,j}) . \quad (3.55)$$

3.3.3 Three Dimensions

Consider the three dimensional Poisson's equation weighted by volume fraction,

$$\begin{aligned} & \left[\frac{\partial}{\partial x} \left(\lambda \frac{\partial \bar{P}}{\partial x} \right) \right]_{i,j,k} + \left[\frac{\partial}{\partial y} \left(\lambda \frac{\partial \bar{P}}{\partial y} \right) \right]_{i,j,k} + \left[\frac{\partial}{\partial z} \left(\lambda \frac{\partial \bar{P}}{\partial z} \right) \right]_{i,j,k} \\ & = [\phi_1 f_1]_{i,j,k} + [\phi_2 f_2]_{i,j,k} + r_{i,j,k} . \end{aligned} \quad (3.56)$$

With straightforward extension of one and two dimensional discretization, the expression of correction term can be obtained as follow,

$$\begin{aligned} r_{i,j,k} = & \frac{1}{h^2} \left(\lambda_{i+1/2,j,k} (\phi_{1_{i+1,j,k}} - \phi_{1_{i,j,k}}) \alpha_{i+1,j,k} - \lambda_{i-1/2,j,k} (\phi_{1_{i,j,k}} - \phi_{1_{i-1,j,k}}) \alpha_{i-1,j,k} \right. \\ & + \lambda_{i,j+1/2,k} (\phi_{1_{i,j+1,k}} - \phi_{1_{i,j,k}}) \alpha_{i,j+1,k} - \lambda_{i,j-1/2,k} (\phi_{1_{i,j,k}} - \phi_{1_{i,j-1,k}}) \alpha_{i,j-1,k} \\ & \left. + \lambda_{i,j,k+1/2} (\phi_{1_{i,j,k+1}} - \phi_{1_{i,j,k}}) \alpha_{i,j,k+1} - \lambda_{i,j,k-1/2} (\phi_{1_{i,j,k}} - \phi_{1_{i,j,k-1}}) \alpha_{i,j,k-1} \right) + O(h^2) . \end{aligned} \quad (3.57)$$

Therefore, for each cell (i, j, k) , the three dimensional Poisson's equation can be discretized dimension by dimension with a form of

$$\begin{aligned} & \frac{\lambda_{i+1/2,j,k} (P_{i+1,j,k} - P_{i,j,k}) - \lambda_{i-1/2,j,k} (P_{i,j,k} - P_{i-1,j,k})}{h^2} \\ & + \frac{\lambda_{i,j+1/2,k} (P_{i,j+1,k} - P_{i,j,k}) - \lambda_{i,j-1/2,k} (P_{i,j,k} - P_{i,j-1,k})}{h^2} \\ & + \frac{\lambda_{i,j,k+1/2} (P_{i,j,k+1} - P_{i,j,k}) - \lambda_{i,j,k-1/2} (P_{i,j,k} - P_{i,j,k-1})}{h^2} \\ & = \phi_1 f_1 + \phi_2 f_2 + F^R - F^L + F^T - F^B + F^N - F^S + O(h^2) , \end{aligned} \quad (3.58)$$

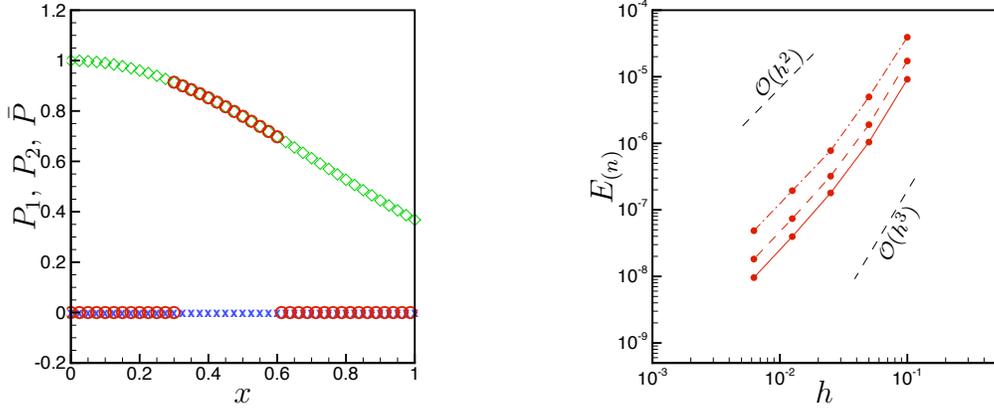


Figure 3.4: (left) Exact and computed solutions. P_1 (green diamond), P_2 (blue cross), and \bar{P} (red cycle). (right) Error vs. grid spacing h . L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line)

where,

$$F^R = \frac{1}{h^2} \lambda_{i+1/2,j,k} (\phi_{1_{i+1,j,k}} - \phi_{1_{i,j,k}}) \alpha_{i+1,j,k} , \quad (3.59)$$

$$F^L = \frac{1}{h^2} \lambda_{i-1/2,j,k} (\phi_{1_{i,j,k}} - \phi_{1_{i-1,j,k}}) \alpha_{i-1,j,k} , \quad (3.60)$$

$$F^T = \frac{1}{h^2} \lambda_{i,j+1/2,k} (\phi_{1_{i,j+1,k}} - \phi_{1_{i,j,k}}) \alpha_{i,j+1,k} , \quad (3.61)$$

$$F^B = \frac{1}{h^2} \lambda_{i,j-1/2,k} (\phi_{1_{i,j,k}} - \phi_{1_{i,j-1,k}}) \alpha_{i,j-1,k} , \quad (3.62)$$

$$F^N = \frac{1}{h^2} \lambda_{i,j,k+1/2} (\phi_{1_{i,j,k+1}} - \phi_{1_{i,j,k}}) \alpha_{i,j,k+1} , \quad (3.63)$$

$$F^S = \frac{1}{h^2} \lambda_{i,j,k-1/2} (\phi_{1_{i,j,k}} - \phi_{1_{i,j,k-1}}) \alpha_{i,j,k-1} , \quad (3.64)$$

and included in the linear system of equation. The alternative forms of correction terms for the cells where interface does not pass through are in a manner similar to the 1-D and 2-D cases described above.

3.4 Numerical results

To demonstrate the validity and accuracy of the method described above, we will use exact solutions of P_1 and P_2 to obtain the functions f_1 and f_2 in Equations (3.1) and (3.2) and the jump conditions given by Equation (3.3). The numerical solution \bar{P} given by Equations (3.15) and (3.35) will be compared with P_1 and P_2 . Three kinds of respective error are defined for numerical solution here with respect to the exact solution by,

L_1 norm of error,

$$E_1 = \frac{1}{h^2} \sum_{i=1}^N \sum_{j=1}^N (\bar{P}_{i,j} - (\phi_1 P_1 + \phi_2 P_2)_{i,j}) , \quad (3.65)$$

L_2 norm of error,

$$E_2 = \left[\frac{1}{h^2} \sum_{i=1}^N \sum_{j=1}^N (\bar{P}_{i,j} - (\phi_1 P_1 + \phi_2 P_2)_{i,j})^2 \right]^{1/2} , \quad (3.66)$$

and L_∞ norm of error,

$$E_\infty = \max |\bar{P}_{i,j} - (\phi_1 P_1 + \phi_2 P_2)_{i,j}| . \quad (3.67)$$

3.4.1 1-D examples

Exact solution are specified as

$$P_1 = c_1 x^2 + c_2 x , \quad (3.68)$$

$$P_2 = c_3 x^3 + x^2 + 1 ,$$

where constants c are chosen to represent various profiles of P_1 and P_2 . P_1 is specified in the region, $x_{I_L} \leq x \leq x_{I_R}$, where x_{I_L} and x_{I_R} are the positions of the left and right

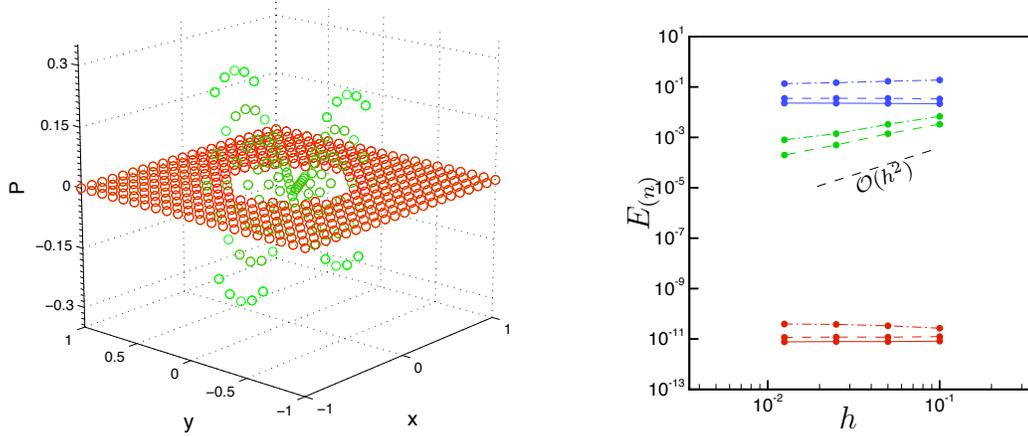


Figure 3.5: (left) Numerical solution for the 2-D case specified by Equations (3.75) for $[P] \neq 0$, $[\lambda P_n] \neq 0$. (right) Error vs. grid spacing h . Colors indicate, methods GFM (green), CSF (blue) and 2nd order scheme (red). L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line).

interface, respectively. P_2 occupies the region, $x \leq x_{I_L}$ and $x \geq x_{I_R}$. Density ratio is $\lambda_2/\lambda_1 = 10^3$. Figure 3.3 shows the exact and numerical solutions in the left plot. The right plot shows the errors that reduce as $O(h^2)$ and $O(h^3)$ for various norms. Note that the value of \bar{P} is converted to P_1 and P_2 in the interfacial cells using Equation (3.3). Therefore two values of the numerical solution can be observed at the same location in the right plot in Fig. 3.3. For the second example, reference [79] the exact solutions are specified as

$$P_1 = e^{-x^2} \tag{3.69}$$

$$P_2 = 0$$

The interior region is defined by $|x - 0.45| \leq 0.15$. We choose $\lambda_1 = 1/1000$ on the interior while on the exterior region $\lambda_2 = 1$. Figure 3.4 shows the solution computed with 80 grid points plotted on the top of the exact solution.

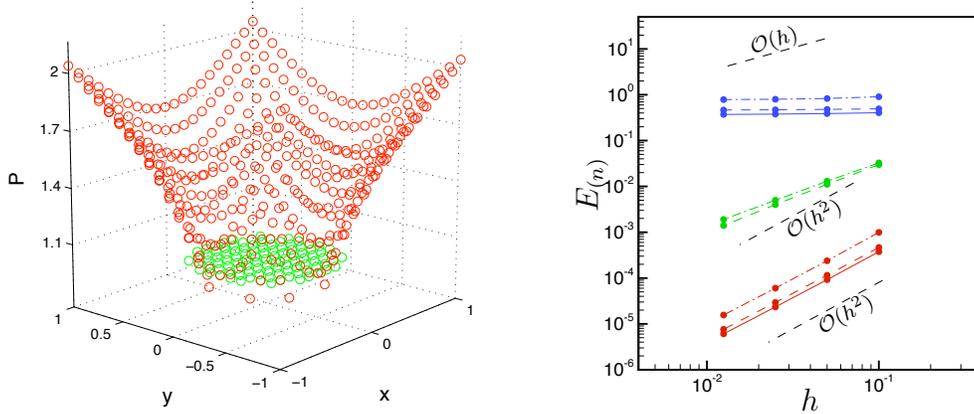


Figure 3.6: (left) Numerical solution for two spatial dimensional case, $[P] = 0$, $[\lambda P_n] \neq 0$. (right) Error vs. grid spacing h . Colors indicate, methods GFM (green), CSF (blue) and 2nd order scheme (red). Compared with GFM and CSF, the slope of new 2nd order scheme is indicated as $O(h^2)$ or larger for L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line) of the error.

3.4.2 2-D examples

Exact solution are specified as

$$\begin{aligned}
 P_1 &= x^2 - y^2, \\
 P_2 &= 0.
 \end{aligned}
 \tag{3.70}$$

P_1 and P_2 are specified in the interior and exterior of the region defined by $x^2 + y^2 = 1/4$. The density ratio is $\lambda_2/\lambda_1 = 1$. The left plot of Figure 3.5 shows the profile of numerical solutions. The right plot shows the convergence analysis for the CSF method, GFM method and our new method. Here, the CSF method is implemented as [73] with a modified delta function, $\delta(\phi)$, smoothed in similar

fashion as in [54],

$$\delta(\phi) = \begin{cases} \frac{1}{2}(1 + \cos(\pi\phi/\alpha))/\alpha & \text{if } |\phi| < \alpha \\ 0 & \text{otherwise} \end{cases}, \quad (3.71)$$

where, α is the thickness of interface, we use $\alpha = 3h$ in our computation.

The new method provides a computed solution with errors below 10^{-10} , which means the truncation error is identically zero and only rounding errors appear in the computed solution. On the other hand, GFM and CSF provide much larger truncation error.

For the second case, following reference [79], the exact solutions are specified as

$$\begin{aligned} P_1 &= 1, \\ P_2 &= 1 + \ln\left(2\sqrt{x^2 + y^2}\right). \end{aligned} \quad (3.72)$$

Consider $\Delta P = 0$ in two spatial dimension on $[-1, 1] \times [-1, 1]$ by setting $\lambda = 1$ on both interior and exterior of the interface, defined by the circle $x^2 + y^2 = 0.5^2$. Figure 3.6 shows the numerical solution with 21 grid points in each direction and also shows the convergence analysis for L_1 , L_2 and L_∞ norm of error. Compared with the data shown in TABLE III in [79] and solution of CSF scheme, our convergence rate are higher while the errors are also substantially lower.

For the third case, also considered by [79], the density ratio is $\lambda_2/\lambda_1 = 0.5$. Exact solution are specified as

$$\begin{aligned} P_1 &= e^{-x^2 - y^2}, \\ P_2 &= 0. \end{aligned} \quad (3.73)$$

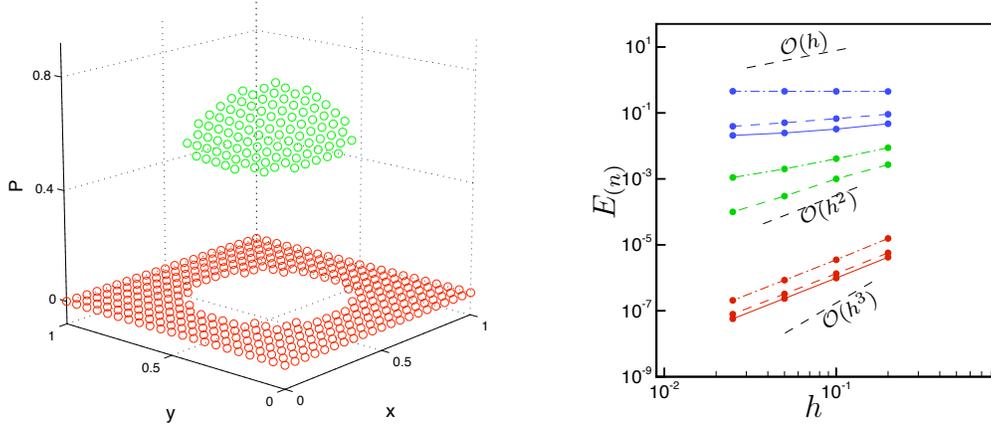


Figure 3.7: (left) Numerical solution for two spatial dimensional case, $[P] \neq 0$, $[\lambda P_n] \neq 0$. (right) Error vs. grid spacing h . Colors indicate, methods GFM (green), CSF (blue) and 2nd order scheme (red). Compared with 1st order scheme and CSF method, the slope of new 2nd order scheme is indicated as $O(h^2)$ or larger for L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line) of the error.

The interface is defined by the circle $(x - 0.5)^2 + (y - 0.5)^2 = 1/16$. P_1 is in the interior of the circle and P_2 is specified outside the circle. Figure 3.7 shows the numerical solution with 21^2 grid points and also shows error in the plot on the right. Compared with the L_2 and L_∞ norm of error shown in [79] and the errors of CSF method, the convergence rate is second order. The errors is also comparatively much smaller.

The fourth example takes the exact solutions to be

$$P_1 = e^x \cos(y) , \tag{3.74}$$

$$P_2 = 0 .$$

This example was used in both [79] and [63]. Consider $\Delta P = 0$ in two spatial

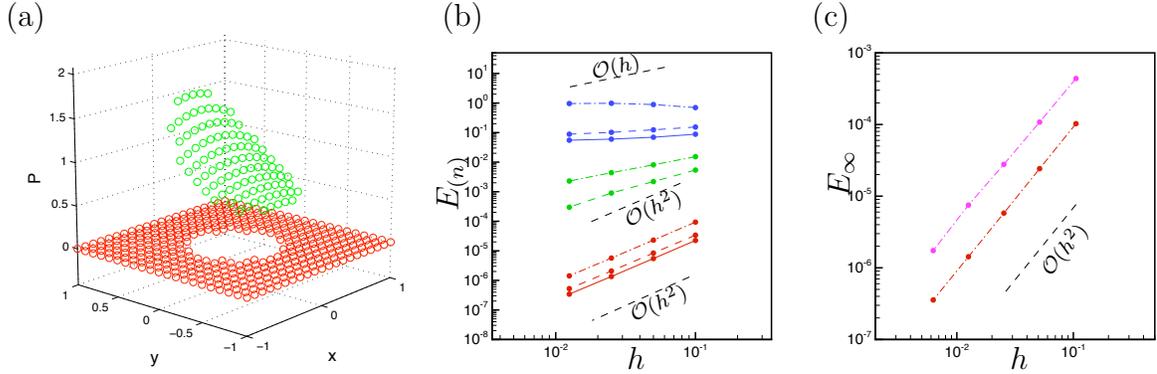


Figure 3.8: (a) Numerical solution for two spatial dimensional case, $[P] \neq 0$, $[\lambda P_n] \neq 0$. (b) Error vs. grid spacing h . Colors indicate, methods GFM (green), CSF (blue) and 2nd order scheme (red). Compared with 1st order scheme and CSF method, the slope of new 2nd order scheme is indicated as $O(h^2)$ or larger for L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line) of the error. (c) L_∞ norm of the error of new jump condition capturing method (red) and "immersed interface" method (pink). The value of new jump condition capturing method is five times lower.

dimension on $[-1, 1] \times [-1, 1]$ by setting $\lambda = 1$ on both interior and exterior of the interface, defined by the circle $x^2 + y^2 = 0.5^2$. Figure 3.8 shows the numerical solution with 21 grid points in each direction and the convergence analysis. Compared with TABLE IV in [79], our orders are higher. In order to compare our new jump condition capturing method with the "immerse interface" method, we plotted out L_∞ norm of the error of the new method and the data shown in Table 3 in [63] in picture(c) of Figure 3.8. It's clear that the error of new jump condition capturing

method is almost five times lower than "immerse interface" method, even though both methods obtain 2nd order convergence for this error.

3.4.3 3-D examples

Then, we extend our method to three spatial dimensional case. Exact solution are specified as

$$\begin{aligned}
 P_1 &= e^{-x^2-y^2-z^2} , \\
 P_2 &= 0 .
 \end{aligned}
 \tag{3.75}$$

P_1 and P_2 are specified in the interior and exterior of the region defined by $x^2+y^2+z^2 = 0.35^2$. The density ratio is $\lambda_1/\lambda_2 = 2$. The left plot of Figure 3.9 shows $P(x, y, z = 0.4)$ cross-section of the numerical solutions with 21 grid points in each direction. The right plot shows the convergence analysis for the two dimensional slice of error by the CSF method, GFM method and our new method. It's clear that our new method can let convergence rate to be even more than 3rd order accurate for L_1 norm, L_2 norm and L_∞ norm of the error.

3.5 Conclusion

A new method for the discretization of Poisson equation for discontinuous fields has been developed. Both the Neumann and Dirichlet jumps can be implemented with second order accuracy. Error magnitudes are also substantially smaller compared with previous methods. As a result the discontinuity across the interface can be preserved accurately. A truly sharp representation of the discontinuity is achieved on

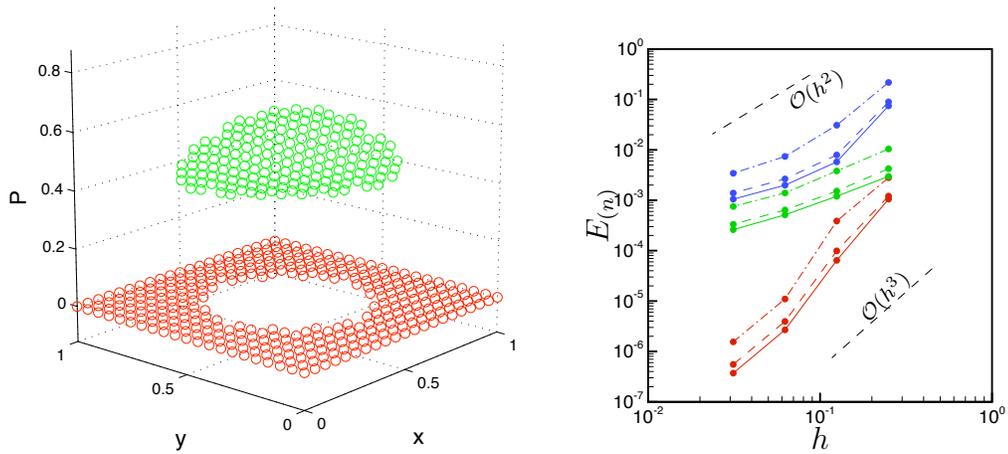


Figure 3.9: (left) $P(x, y, z = 0.4)$ cross section of numerical solution for three spatial dimensional case, $[P] \neq 0$, $[\lambda P_n] \neq 0$. (right) Error vs. grid spacing h . Colors indicate, methods GFM (green), CSF (blue) and 2nd order scheme (red). Compared with 1st order scheme and CSF method, the slope of new 2nd order scheme is indicated as more than $O(h^3)$ or larger for L_1 norm (solid line), L_2 norm (dashed line) and L_∞ norm (dashed-dot line) of the error.

the basis of the two-phase Poisson equation. The volume weighted average solution can be separated to yield two solutions within individual interfacial cells. The new method is relatively simple to implement with a dimension by dimension discretization of the jump conditions. Moreover, because the associated coefficient matrix is symmetric, standard fast linear solvers can be utilized. The new method is suitable for incompressible two-phase flow because the coefficient matrix is conservative. This extension is the subject of ongoing work.

Chapter 4: High Accurate Solution of Two Phase Flow Problems

4.1 Overview

In this chapter, a higher order accurate solver is proposed of two-phase incompressible Navier-Stokes Equation. The new advection scheme introduced in Chapter 3 is combined with the new second order jump condition capturing schemes to solve the velocity field by the projection method for pressure and velocity. The validity of these two new methods are checked in this chapter by more advanced application. Previously, Kang & R. Fedkiw [41] used a 5th order WENO scheme to solve the interface advection equation and used the classic redistance scheme to reinitialize the level set function [72]. To solve the Navier-Stokes equation, they used projection method to build a Poisson's equation and used a first order accurate boundary condition capturing method, proposed by Xu-Dong Liu & R. Fedkiw [79], to update pressure. The advantages of new advection scheme and jump condition capturing scheme has been discussed in previous chapters. In this chapter, a new method is developed to solve two phase flow problems. Similar to GFM, the new method is stepwise implemented. Intermediate vector field is projected onto divergence-free field to recover the velocity at first step. Then the Poisson's equation about pressure is solved by a second order accurate jump condition capturing scheme proposed by

[57]. By appropriate additional condition, the velocity field can be guaranteed to be divergence free in each direction. In the end, a topological projection level set method, proposed by [56], is associated with the computation to advect the interface and complete the reinitialization step for level set function.

4.2 Governing Equation

Deriving from the viscous compressible Navier-Stocks equation by the divergence free condition, $\nabla \cdot \vec{V} = 0$, the basic governing equations for viscous incompressible flow are,

$$\rho_t + \vec{V} \cdot \nabla \rho = 0, \quad (4.1)$$

$$u_t + \vec{V} \cdot \nabla u + \frac{p_x}{\rho} = \frac{(2\mu u_x)_x + (\mu(u_y + v_x))_y + (\mu(u_z + w_x))_z}{\rho}, \quad (4.2)$$

$$v_t + \vec{V} \cdot \nabla v + \frac{p_y}{\rho} = \frac{(\mu(u_y + v_x))_x + (2\mu v_y)_y + (\mu(v_z + w_y))_z}{\rho} + g, \quad (4.3)$$

$$w_t + \vec{V} \cdot \nabla w + \frac{p_z}{\rho} = \frac{(\mu(u_z + w_x))_x + (\mu(v_z + w_y))_y + (2\mu w_z)_z}{\rho}, \quad (4.4)$$

where t is the time scale, $X = \langle x, y, z \rangle$ are the spatial coordinate system, $\vec{V} = \langle u, v, w \rangle$ are the velocity field, p is the pressure, ρ is the density, μ is the viscosity, and ∇ is equal to $\frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k}$. Therefore, the incompressible Navier-Stocks equation can be rewritten as

$$\vec{V}_t + (\vec{V} \cdot \nabla) \vec{V} + \frac{\nabla p}{\rho} = \frac{\nabla \cdot \tau^T}{\rho} + \vec{g} \quad (4.5)$$

Where "T" represents the transpose operator, and the viscous term could be written as $\nabla \cdot \tau^T = \nabla \cdot (\mu(\nabla \vec{V} + (\nabla \vec{V})^T))$. Now, we can non-dimensionalize the Navier-

Stocks equation (4.5) by the following dimensionless variables,

$$x^* = \frac{x}{L}, y^* = \frac{y}{L}, z^* = \frac{z}{L}, u^* = \frac{u}{U}, t^* = \frac{t}{L/U}, p^* = \frac{p}{\rho U^2}, \rho^* = \frac{\rho}{\bar{\rho}}, \mu^* = \frac{\mu}{\bar{\mu}} \quad (4.6)$$

Thus, the non-dimensional Navier-Stocks equation can be expressed as follow:

$$\vec{V}_t + (\vec{V} \cdot \nabla) \vec{V} = -\frac{\nabla p}{\rho} + \frac{1}{Re} \frac{\nabla \cdot (\mu(\nabla \vec{V} + (\nabla \vec{V})^T))}{\rho} + \frac{1}{Fr} \quad (4.7)$$

Where, $Fr = \frac{U^2}{gL}$ and $Re = \frac{\bar{\rho}LU}{\bar{\mu}}$. Then we typed the fortran code to solve the equation (4.7).

4.3 Proposed Numerical Method

As similar to [41], we use a standard MAC grid for discretization where $p_{i,j,k}$, $\rho_{i,j,k}$, $\mu_{i,j,k}$ and $\phi_{i,j,k}$ exist at the cell centers and $u_{i\pm\frac{1}{2},j,k}$, $v_{i,j\pm\frac{1}{2},k}$ and $w_{i,j,k\pm\frac{1}{2}}$ exist at the appropriate cell walls.

4.3.1 Advection Scheme

In our work, Level set function is used to locate the interface dividing different phases. Across the interface, continuous viscosity and density are defined as

$$\rho(\phi) = (\rho_1 - \rho_2)I + \rho_2, \quad \mu(\phi) = (\mu_1 - \mu_2)I + \mu_2, \quad (4.8)$$

where ρ and μ indicate density and viscosity, respectively, I is the transition function and subscripts, 1 and 2, indicate the phase. The transition function depends on the instantaneous signed distance function and is given by

$$I(\phi) = \frac{1}{2} [1 - \text{erf}(\phi/\varepsilon)] \quad \text{and} \quad I(\phi) = \phi^\pm / (\phi^+ - \phi^-), \quad (4.9)$$

for a smooth and a sharp transition across the interface, respectively. In our work, 5th order WENO scheme is called to advect the level set function by solving following advection equation. And the topologic projection [57] works as a reinitialization step, by which, the value of ϕ is kept close to those of a signed distance function, i.e. $|\nabla\phi| = 1$,

$$\phi_t + \vec{V} \cdot \nabla\phi = 0. \quad (4.10)$$

According to the sensitive degree of the multiphase flow cases, the best choice can be made from different levels of the topologic projection, P_i scheme. At every time step, the normal and curvature of the interface are calculated with level set function, these topological properties will be used for jump condition across the interface,

$$\vec{N} = \frac{\nabla\phi}{|\nabla\phi|}, \quad (4.11)$$

$$\kappa = -\nabla \cdot \vec{N}. \quad (4.12)$$

4.3.2 Projection Method

4.3.2.1 Projection Method

First, we set a intermedian velocity value, \vec{V}^* , whose value could be defined by,

$$\frac{\vec{V}^* - \vec{V}^n}{\Delta t} + (\vec{V} \cdot \nabla)\vec{V} = \frac{\nabla \cdot \tau^T}{\rho} + \vec{g}. \quad (4.13)$$

By this intermedian value, the velocity at next time step, \vec{V}^{n+1} , can be defined by,

$$\frac{\vec{V}^{n+1} - \vec{V}^*}{\Delta t} + \frac{\nabla p}{\rho} = 0. \quad (4.14)$$

The combination of equation (4.13) and (4.14) results in equation (4.7). We take the divergence of equation (4.14) and set the divergence of \vec{V}^{n+1} to 0, a Poisson's equation about pressure could be built as follow,

$$\nabla \cdot \frac{\nabla p}{\rho} = \frac{\nabla \cdot \vec{V}^*}{\Delta t} . \quad (4.15)$$

4.3.2.2 Boundary condition

At every time step, the sequence of solution should be equation (4.13) first, and then equation (4.15), the step final velocity \vec{V}^{n+1} is solved in the end. The following compatibility condition should be applied to the velocity after computing \vec{V}^* and before solving equation (4.15),

$$\int_{\Gamma} \vec{V}^* \cdot \vec{N} = 0 , \quad (4.16)$$

Where, Γ represents the boundary of the computational domain and \vec{N} is the unit normal to that boundary. For the Poisson's Equation (4.15) about pressure, we set the boundary condition as $\nabla p \cdot \vec{N} = 0$.

4.4 Poisson's Equation

The Poisson's Equation's right hand side including \vec{V}^* can be computed from the convection and viscous terms. A new jump condition capturing method is used to make second order accurate solution.

4.4.1 Convection Terms

Standard Marker and Cells formulation is used for the discretization of Convection Term.

4.4.2 Viscous Terms

Updating \vec{V}^* also requires discretization of viscous terms. For example, the discretization of

$$\frac{1}{Re} \frac{(2\mu u_x)_x + (\mu(u_y + v_x))_y + (\mu(u_z + w_x))_z}{\rho} \quad (4.17)$$

at $\vec{x}_{i\pm\frac{1}{2},j,k}$ is used to update $u_{i\pm\frac{1}{2},j,k}^*$. Since the velocities are continuous, we compute the first derivatives in three directions using central differencing as follow.

$$(u_x)_{i,j,k} = \frac{u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}}{\Delta x}, \quad (4.18)$$

$$(u_y)_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{u_{i+\frac{1}{2},j+1,k} - u_{i+\frac{1}{2},j,k}}{\Delta x}, \quad (4.19)$$

$$(u_z)_{i+\frac{1}{2},j,k+\frac{1}{2}} = \frac{u_{i+\frac{1}{2},j,k+1} - u_{i+\frac{1}{2},j,k}}{\Delta x}. \quad (4.20)$$

We also calculate the other terms by central differencing. For example, the second term of equation (4.17), $(\mu(u_y + v_x))_y$, can be approximated as,

$$\frac{\mu_{i+\frac{1}{2},j+\frac{1}{2},k}(u_y + v_x)_{i+\frac{1}{2},j+\frac{1}{2},k} - \mu_{i+\frac{1}{2},j-\frac{1}{2},k}(u_y + v_x)_{i+\frac{1}{2},j-\frac{1}{2},k}}{\Delta y}, \quad (4.21)$$

where, the values of viscosity are calculated by averaging as,

$$\mu_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{\mu_{i+1,j+1,k} + \mu_{i,j+1,k} + \mu_{i+1,j+1,k} + \mu_{i,j+1,k}}{4}, \quad (4.22)$$

$$\mu_{i+\frac{1}{2},j-\frac{1}{2},k} = \frac{\mu_{i+1,j,k} + \mu_{i,j,k} + \mu_{i+1,j-1,k} + \mu_{i,j-1,k}}{4}. \quad (4.23)$$

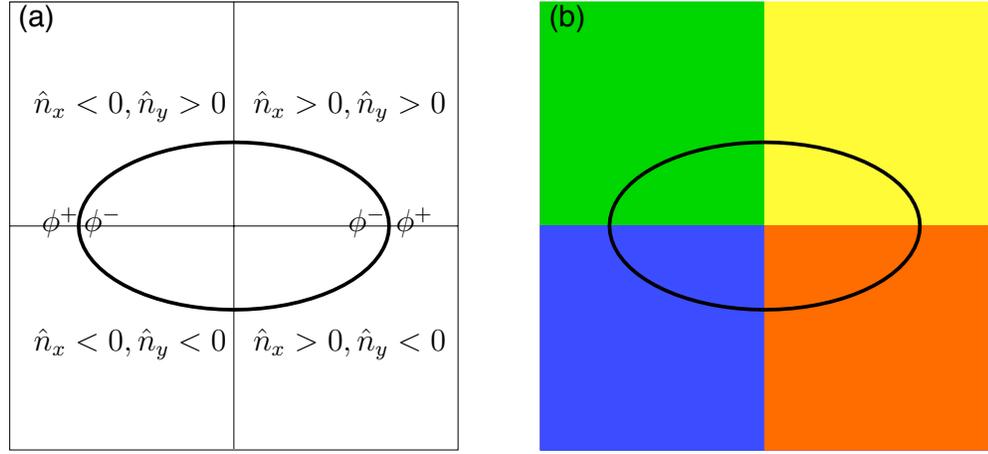


Figure 4.1: (a) Computational domain separated into four quadrants by the sign of \hat{n} . (b) "upwind" method builds conservative jump condition for each quadrant, e.g., the jump condition form is Equation (4.34) in the yellow domain, first quadrant.

4.4.3 Jump condition capturing method

After \vec{V}^* was solved, the right hand side of equation (4.15) could be computed by the same central spatial difference scheme. For two phase flow problem, the Poisson's equation about pressure can be expressed as equation (4.24) according to 2nd order accurate jump condition capturing scheme introduced in [56].

$$\nabla \cdot \frac{\nabla p}{\rho} = \frac{\nabla \cdot \vec{V}^*}{\Delta t} + R \quad (4.24)$$

Where R is the jump condition on each cell center. In [56], the discretization form of jump condition R was defined as,

$$\begin{aligned} r_{i,j} = & \frac{1}{h^2} (\lambda_{i+1/2,j}(\Phi_{1_{i+1,j}} - \Phi_{1_{i,j}})\alpha_{i+1,j} - \lambda_{i-1/2,j}(\Phi_{1_{i,j}} - \Phi_{1_{i-1,j}})\alpha_{i-1,j} \\ & + \lambda_{i,j+1/2}(\Phi_{1_{i,j+1}} - \Phi_{1_{i,j}})\alpha_{i,j+1} - \lambda_{i,j-1/2}(\Phi_{1_{i,j}} - \Phi_{1_{i,j-1}})\alpha_{i,j-1} , \end{aligned} \quad (4.25)$$

where Φ_1 is the volume fraction, α is the pressure jump and λ is density.

4.4.3.1 Conservative form of jump condition

A conservative form of jump condition is always expected because it can guarantee velocity field to be divergence free. However, the original jump condition in [56] is not conservative. The reason of non-conservative is obvious if we rewrite the discretization form of R as equation (4.26),

$$r_{i,j} = \delta_{i+1/2,j}^x - \delta_{i-1/2,j}^x + \delta_{i,j+1/2}^y - \delta_{i,j-1/2}^y , \quad (4.26)$$

where the δ functions stand for four parts of the right hand side in equation (4.25),

$$\begin{aligned} \delta_{i+1/2,j}^x &= \frac{1}{h^2} (\lambda_{i+1/2,j} (\Phi_{1_{i+1,j}} - \Phi_{1_{i,j}}) \alpha_{i+1,j}) , \\ \delta_{i-1/2,j}^x &= \frac{1}{h^2} (\lambda_{i-1/2,j} (\Phi_{1_{i,j}} - \Phi_{1_{i-1,j}}) \alpha_{i-1,j}) , \\ \delta_{i,j+1/2}^y &= \frac{1}{h^2} (\lambda_{i,j+1/2} (\Phi_{1_{i,j+1}} - \Phi_{1_{i,j}}) \alpha_{i,j+1}) , \\ \delta_{i,j-1/2}^y &= \frac{1}{h^2} (\lambda_{i,j-1/2} (\Phi_{1_{i,j}} - \Phi_{1_{i,j-1}}) \alpha_{i,j-1}) . \end{aligned} \quad (4.27)$$

It's clear that $[\delta_{i+1/2,j}^x]_{i-1,j} \neq \delta_{i-1/2,j}^x$ and $[\delta_{i,j+1/2}^y]_{i,j-1} \neq \delta_{i,j-1/2}^y$. That means other condition is needed to obtain conservative jump condition if the divergence free velocity field is expected. In this work, we use derivative jump condition in equation (4.28),

$$[\lambda(\nabla P_1 - \nabla P_2)] \cdot \hat{n} = \beta , \quad (4.28)$$

where, \hat{n} is normal of the volume fraction Φ_1 , as follow,

$$\hat{n} = \frac{1}{|\nabla\Phi|} \left(\frac{\partial\Phi}{\partial x} \hat{i} + \frac{\partial\Phi}{\partial y} \hat{j} \right) . \quad (4.29)$$

To discrete equation (4.28), an "upwind" method is used. As show in Fig. 4.1, the interface can be separated into several quadrants based on the value of normal $\hat{n} = (n_x, n_y)$. For example, equation (4.30) is used as the discretization of (4.28) in the first quadrant where both of n_x and n_y are positive. Similarly, the discretization forms in the second to fourth quadrants are given in equation (4.31) to (4.33),

$$\begin{aligned} & \lambda_{i+1/2,j}(\Phi_{1_{i+1,j}} - \Phi_{1_{i,j}})(\alpha_{i+1,j} - \alpha_{i,j}) + \lambda_{i,j+1/2}(\Phi_{1_{i,j+1}} - \Phi_{1_{i,j}})(\alpha_{i,j+1} - \alpha_{i,j}) \\ & = h^2\beta + O(h^2) , \end{aligned} \quad (4.30)$$

$$\begin{aligned} & \lambda_{i-1/2,j}(\Phi_{1_{i,j}} - \Phi_{1_{i-1,j}})(\alpha_{i,j} - \alpha_{i-1,j}) + \lambda_{i,j+1/2}(\Phi_{1_{i,j+1}} - \Phi_{1_{i,j}})(\alpha_{i,j+1} - \alpha_{i,j}) \\ & = h^2\beta + O(h^2) , \end{aligned} \quad (4.31)$$

$$\begin{aligned} & \lambda_{i-1/2,j}(\Phi_{1_{i,j}} - \Phi_{1_{i-1,j}})(\alpha_{i,j} - \alpha_{i-1,j}) + \lambda_{i,j-1/2}(\Phi_{1_{i,j}} - \Phi_{1_{i,j-1}})(\alpha_{i,j} - \alpha_{i,j-1}) \\ & = h^2\beta + O(h^2) , \end{aligned} \quad (4.32)$$

$$\begin{aligned} & \lambda_{i+1/2,j}(\Phi_{1_{i+1,j}} - \Phi_{1_{i,j}})(\alpha_{i+1,j} - \alpha_{i,j}) + \lambda_{i,j-1/2}(\Phi_{1_{i,j}} - \Phi_{1_{i,j-1}})(\alpha_{i,j} - \alpha_{i,j-1}) \\ & = h^2\beta + O(h^2) . \end{aligned} \quad (4.33)$$

Then we give equation (4.31) to (4.33) into the original jump condition, equation (4.25), the conservative forms of jump condition in each quadrants can be obtained in equation (4.34) to (4.37). Because the upwind method is second order accurate, the accuracy of modified jump condition is not reduced,

$$\begin{aligned} r_{i,j} = & \frac{1}{h^2} (\lambda_{i+1/2,j}(\Phi_{1_{i+1,j}} - \Phi_{1_{i,j}})\alpha_{i+1,j} - \lambda_{i-1/2,j}(\Phi_{1_{i,j}} - \Phi_{1_{i-1,j}})\alpha_{i,j} \\ & + \lambda_{i,j+1/2}(\Phi_{1_{i,j+1}} - \Phi_{1_{i,j}})\alpha_{i,j+1} - \lambda_{i,j-1/2}(\Phi_{1_{i,j}} - \Phi_{1_{i,j-1}})\alpha_{i,j} + \beta , \end{aligned} \quad (4.34)$$

$$\begin{aligned}
r_{i,j} = & \frac{1}{h^2} (\lambda_{i+1/2,j}(\Phi_{1_{i+1,j}} - \Phi_{1_{i,j}})\alpha_{i+1,j} - \lambda_{i-1/2,j}(\Phi_{1_{i,j}} - \Phi_{1_{i-1,j}})\alpha_{i,j} \\
& + \lambda_{i,j+1/2}(\Phi_{1_{i,j+1}} - \Phi_{1_{i,j}})\alpha_{i,j} - \lambda_{i,j-1/2}(\Phi_{1_{i,j}} - \Phi_{1_{i,j-1}})\alpha_{i,j-1} + \beta ,
\end{aligned} \tag{4.35}$$

$$\begin{aligned}
r_{i,j} = & \frac{1}{h^2} (\lambda_{i+1/2,j}(\Phi_{1_{i+1,j}} - \Phi_{1_{i,j}})\alpha_{i,j} - \lambda_{i-1/2,j}(\Phi_{1_{i,j}} - \Phi_{1_{i-1,j}})\alpha_{i-1,j} \\
& + \lambda_{i,j+1/2}(\Phi_{1_{i,j+1}} - \Phi_{1_{i,j}})\alpha_{i,j} - \lambda_{i,j-1/2}(\Phi_{1_{i,j}} - \Phi_{1_{i,j-1}})\alpha_{i,j-1} + \beta ,
\end{aligned} \tag{4.36}$$

$$\begin{aligned}
r_{i,j} = & \frac{1}{h^2} (\lambda_{i+1/2,j}(\Phi_{1_{i+1,j}} - \Phi_{1_{i,j}})\alpha_{i,j} - \lambda_{i-1/2,j}(\Phi_{1_{i,j}} - \Phi_{1_{i-1,j}})\alpha_{i-1,j} \\
& + \lambda_{i,j+1/2}(\Phi_{1_{i,j+1}} - \Phi_{1_{i,j}})\alpha_{i,j+1} - \lambda_{i,j-1/2}(\Phi_{1_{i,j}} - \Phi_{1_{i,j-1}})\alpha_{i,j} + \beta .
\end{aligned} \tag{4.37}$$

4.4.3.2 Value of α

The pressure jump equation, $\alpha_{i,j} = P_{1_{i,j}} - P_{2_{i,j}}$, is a key term in equation (4.34) to (4.37). In [56], the value of α is obtained by the exact solution on each cell center. However, the exact solution is not available for two phase flow. Therefore, we use the curvature of the closest interface segment to calculate $\alpha_{i,j}$, as shown in Fig. 4.2. By the geometric projection method, the value of level set function is the distance from cell center to the interface. Thus, $\alpha_{i,j}$ is calculated by equation (4.38),

$$\alpha_{i,j} = \delta \left(\frac{1}{\frac{1}{\kappa_{i,j}} - \phi_{i,j}} \right) , \tag{4.38}$$

where, δ is surface tension and $\kappa_{i,j}$ is curvature of cell (i, j) .

4.5 Temporal Updating Method

Here, the first order finite different method is used to update computational domain at every time step.

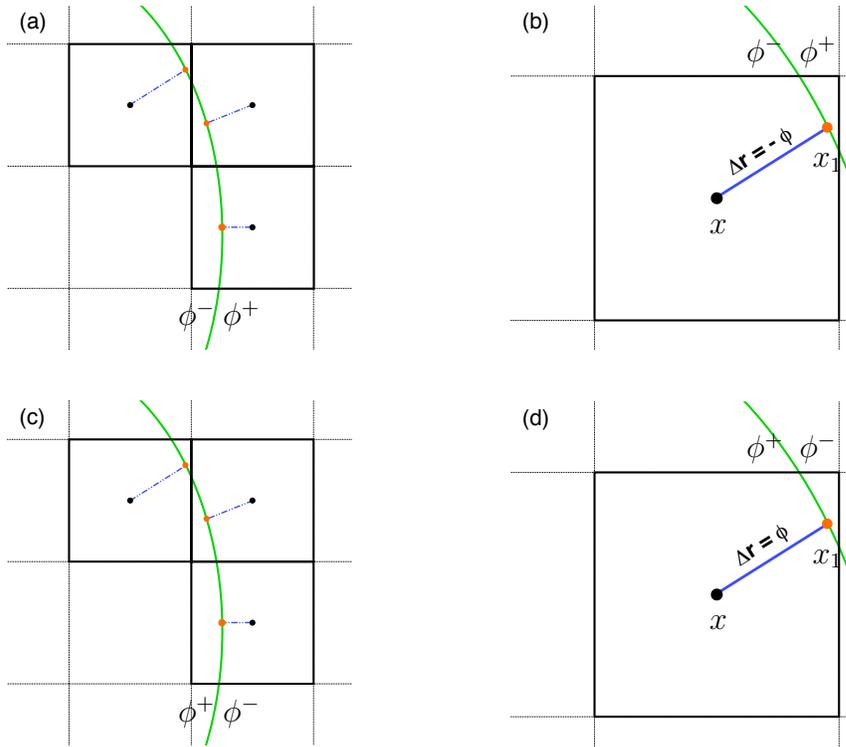


Figure 4.2: Implementation of the α . (a) Localization of grids where the interface passes through. Curvature is calculated by second order discretization on the cell centers (black symbols). The value of α of the grids is the curvature of the interface segments (red symbols) corresponding to the cell centers. The level set function is negative on the left and positive on the right hand side. (b) A cell center x and its corresponding interface segment, x_1 . The character ratio at x_1 is $r_x + \Delta r$, then $r_x - \phi$. (c) Same localization of grids as (a) with opposite level set function. (d) With the new condition, the character ratio at x_1 is $r_x - \Delta r$, then $r_x - \phi$.

4.6 Numerical Solution

4.6.1 Steady cycle drop

The first numerical example is a simple case. The circular interface is represented by the exact signed distance function, ϕ , that can be defined by $\phi = 1 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2}$. The numerical solution is obtained by 2nd jump condition capturing method for zero pressure outside the droplet with $We = 1$. For the dimensionless problem, the curvature is calculated by equation (4.12). On the other hand, the exact solution of this case is a stationary fluid that is independent of Re for which, $[p] = 1/We = 1$ and $\mathbf{u} = 0$. In test computation, we set the density and viscosity ratio as 10^2 . As comparison, CSF method and first order accurate method are used for numerical solution. Figure. 4.3 shows L2 norm of the pressure and velocity error by these three schemes. It's clear that our new scheme can reach second order accuracy for pressure field and closed to fourth order accuracy for velocity field while pressure solutions are lower than first order for both CSF and first order accurate scheme.

4.6.2 Oscillation drop

In order to evaluate the new jump condition capturing method, we apply it for a more complex oscillate case. Initial interface is set as a elliptic, $x^2/a^2 + a^2y^2 = 1$, in the domain $[0, 3] \times [0, 3]$. The stretching amplitude a is the ratio of the major axis to equilibrium radius. In this case, the competition between viscous, pressure

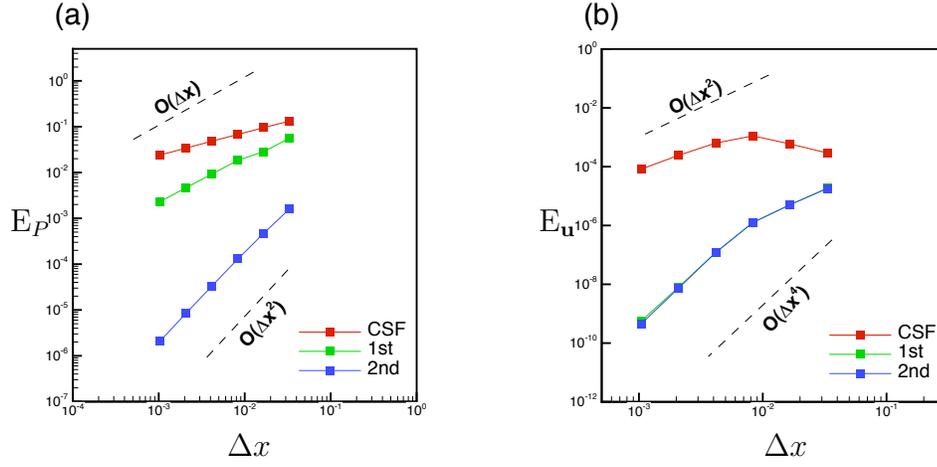


Figure 4.3: (a) Error of pressure (E_P) and (b) error of velocity (E_u) for steady drop case and with density and viscosity ratio of 10^2 and $We = 1$. Solution by CSF method, 1st order scheme and our 2nd order scheme are indicated by red line, green line and blue line.

and capillary forces drives the interface toward a circular equilibrium profile through damped periodic oscillations. According to [lambe], the period of oscillations, τ , is a function only of We and is given by the analytic solution, $\tau = 2\pi\sqrt{We/6}$, for a small stretching amplitude. Here, we let the We to be 1, so the analytical period is 2.5651. The amplitude of oscillations is governed by the total energy,

$$E = \frac{1}{2} \int |\mathbf{u}|^2 dA + \frac{1}{We} \int d\ell, \quad (4.39)$$

where the first term on the right is the kinetic energy E_k and the second term represents interfacial energy over the perimeter ℓ of the interface. For small stretching amplitudes the total energy decays according to the analytical solution,

$$E = E_o \exp(-4\sqrt{We}/Re), \quad (4.40)$$

where E_o is the initial energy. The initial condition is the quiescent state, $\mathbf{u} = 0$ and hence E_o is purely interfacial. To compare our numerical solution with the analytical one, we give small stretching to the initial interface, say $a = 1.01$. The environment outside of the interface is also vacuum, we set the density and viscosity ratio to be Inf and $Re = 100$. With 100×100 grid points, oscillation of the interface about the equilibrium position and the associated kinetic energy is plotted in Fig. 4.4. As comparison, we also use the 1st order scheme to solve this problem. From Fig. 4.4, the kinetic energy E_k/E_o by our new 2nd order scheme is observed to be in much better agreement with the analytical solution given by Eq. (4.40). The average value of first several period is compared with analytical solution and given in Table 4.2. Our new method converge this error and provide almost exact solution on high

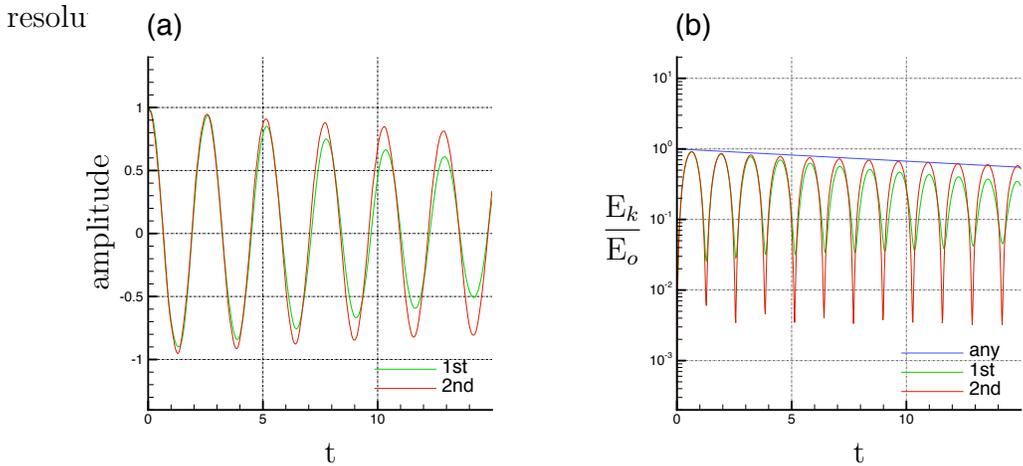


Figure 4.4: (a) Amplitude oscillation of the interface and (b) the kinetic energy for density and viscosity ratio of Inf , $We = 1$ and $Re = 100$. Solid blue line in (right) indicates the analytical solution.

Table 4.1: Numerical period by 1st order and 2nd order jump condition capturing method and the error from the analytical solution. Our new jump condition method can obtain almost exact solution on high resolution

Δx	τ_{1st}	$[\Delta\tau]_{1st}$	τ_{2nd}	$[\Delta\tau]_{2nd}$
3/25	2.6403	0.0752	2.6403	0.0752
3/50	2.5502	0.0149	2.5803	0.0152
3/100	2.5878	0.0221	2.5728	0.0077
3/200	2.5614	0.0037	2.5652	0.0001

4.6.3 Rising bubble

Next, we consider the rising bubble problem where all of gravitational, interfacial and viscous forces are in competition. In this case, we use volume conservation error to evaluate our new method. It's according to

$$E^{(A)} = \frac{1}{A_0} |A - A_0| , \quad (4.41)$$

where A_0 is the initial area and A is the area at any other time given by

$$A = \frac{1}{NM} \sum_{i,j} H_{i,j} , \quad H_{i,j} = \begin{cases} H_{i,j} = 0 & , \quad \phi_{i,j} \geq 0 \\ H_{i,j} = 1 & , \quad \phi_{i,j} < 1 \end{cases} . \quad (4.42)$$

where N and M are the grid number on x and y direction. Our computation is base on the Reynolds number, $Re = \rho_1 g^{1/2} L^{3/2} / \mu_1$, the Weber number $We = \rho_1 g L^2 / \sigma = \infty$ and the Froude number $Fr = 1$. At first, we test the ability of jump condition capturing schemes to maintain the mass conservation of the interface breakup events.

Fig. 4.5 shows the evolution of interface with $Re = 100$ and $We = 200$ by our new second order method. It's obvious to capture the interface very well even after the bubble started to break ($t=5.4$). Fig. 4.6 demonstrates further that our method can keep the volume conservation error to converge before and after breakup. Compared with 1st order scheme, the new method also reduce the absolute value of error on higher resolution. Then, we increase the effect of interfacial by decreasing Weber number. It's found that our method can always provide good solution while the 1st order scheme works worse and even fail with small Weber number. Fig. 4.7 shows the solution with $We = 2$ at different time by our new method. Fig. 4.8 demonstrates that the new method can keep the low volume conservation error and converge this error almost by 2nd order accuracy.

4.7 Conclusion

With the discretization of normal derivative jump, new high order accurate boundary condition capturing method is applied for Navier-Stokes equation solver and guarantee the velocity field to be divergence free. Compared with exact solution, the new method provides second order accurate pressure and nearly fourth order accurate velocity field for steady drop case. Associated with topological preserved interface advection method, new Navier-Stokes solver significantly improves the solution of two-phase flow cases about both topological properties and mass conservation.

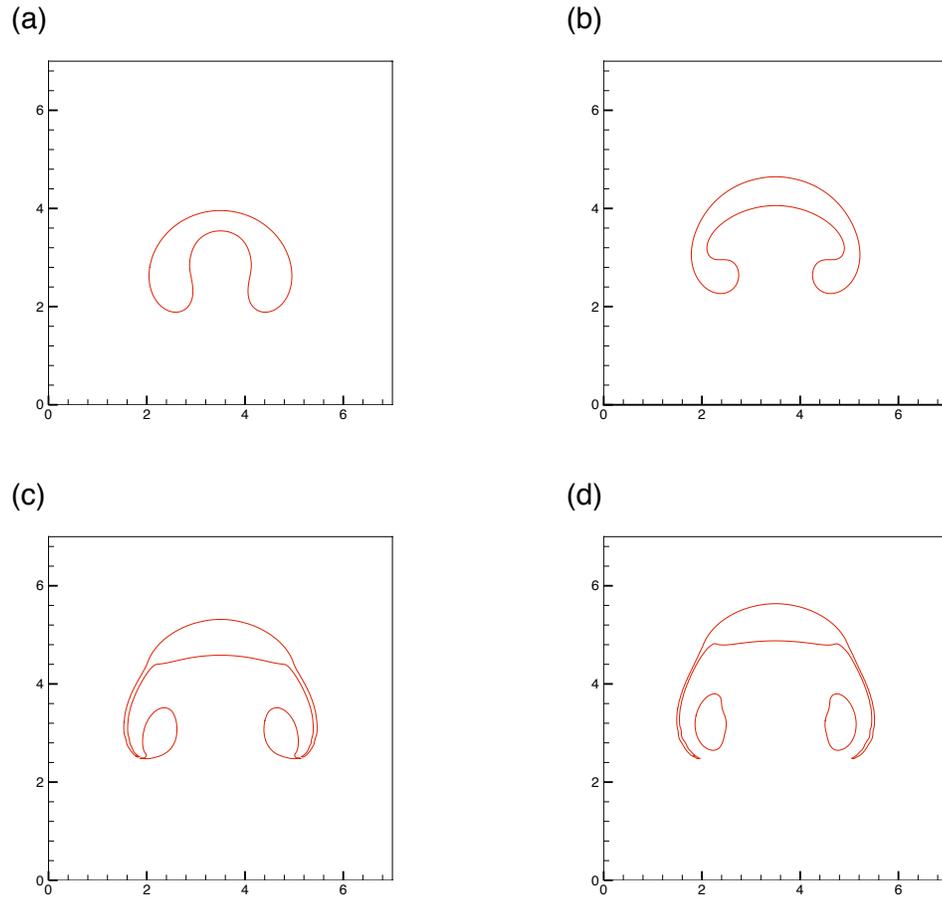


Figure 4.5: Interface profiles for the rising bubble problem on a 200×200 grid for $Re = 100$, $We = 200$ and $Fr = 1$. at (a) $t = 3.0$; (b) $t = 4.2$; (c) $t = 5.4$; (d) $t = 6.0$.

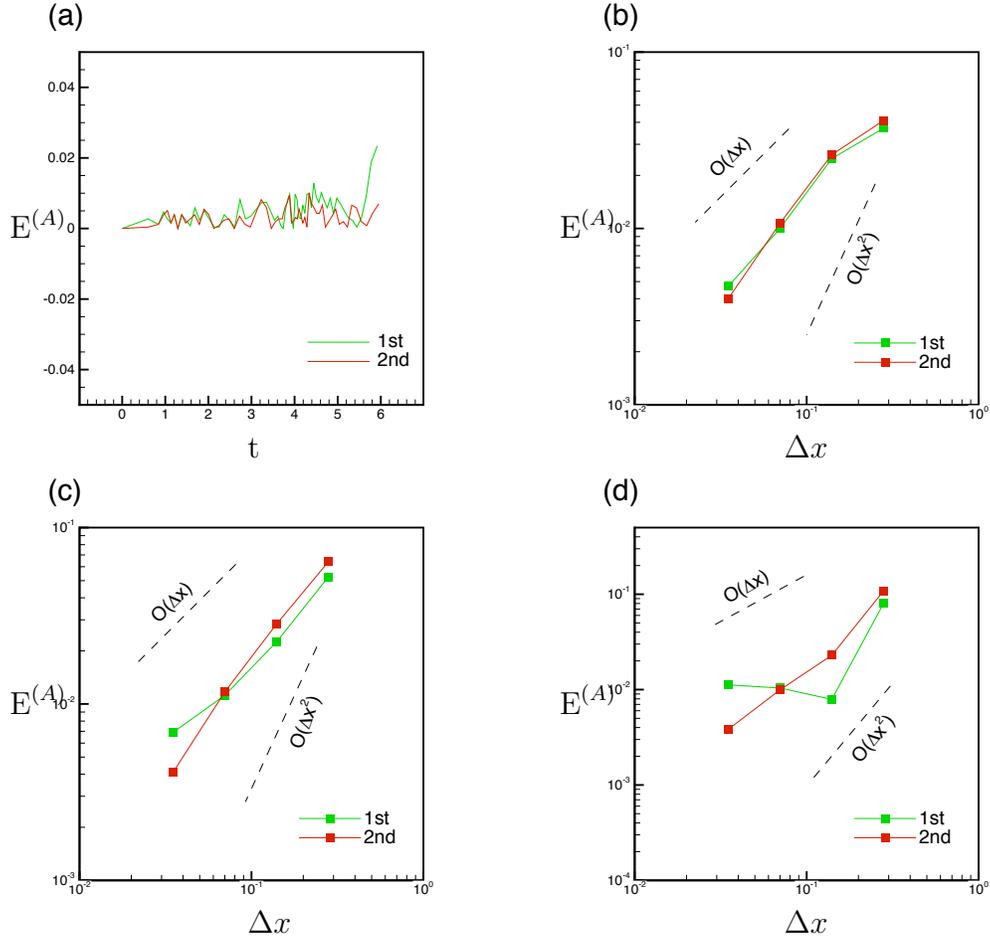


Figure 4.6: Accuracy of numerical simulation of rising bubble for $Re = 100$ and $We = 200$. (a) Volume conservation error as a function of time on a 200×200 grid, (b) error on different sized grids at $t = 4$, (c) error on different sized grids at $t = 5$, (d) error on different sized grids at $t = 6$. Lines indicate new jump condition capturing method (red) and the 1st order one (green). The error is smaller in the case of our new method and decays with 1.5th order accuracy even after bubble broken.

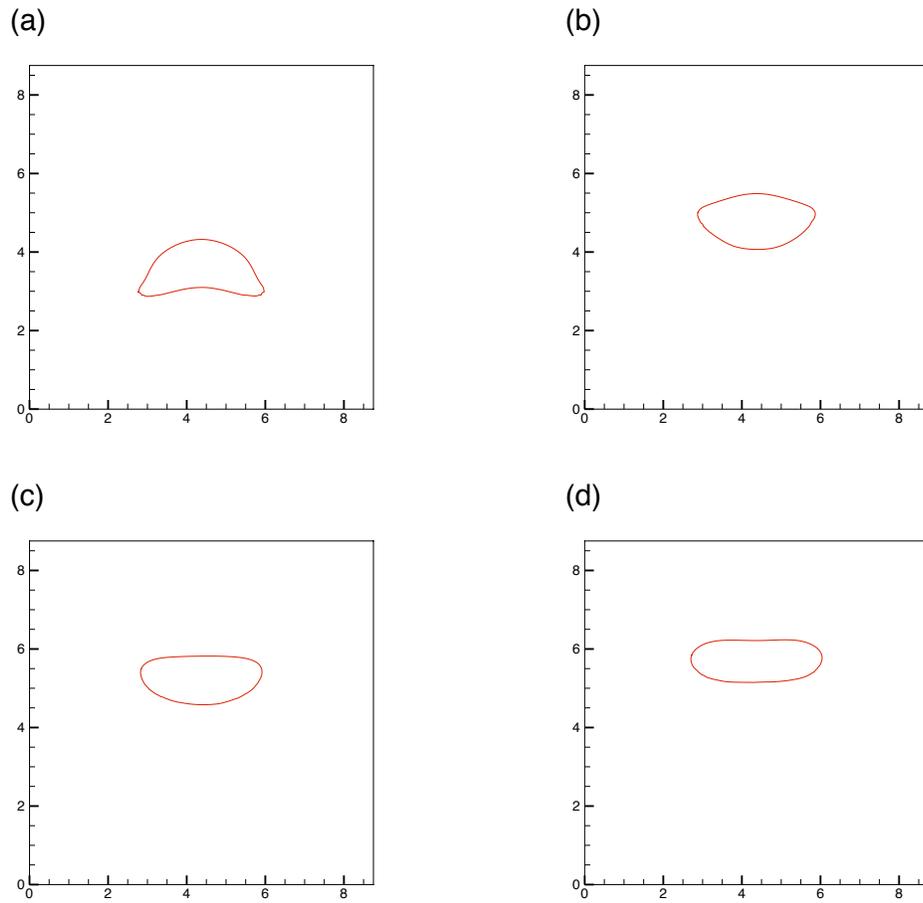


Figure 4.7: Interface profiles for the rising bubble problem on a 200×200 grid for $Re = 100$, $We = 2$ and $Fr = 1$. at (a) $t = 3.0$; (b) $t = 4.8$; (c) $t = 5.4$; (d) $t = 6.0$.

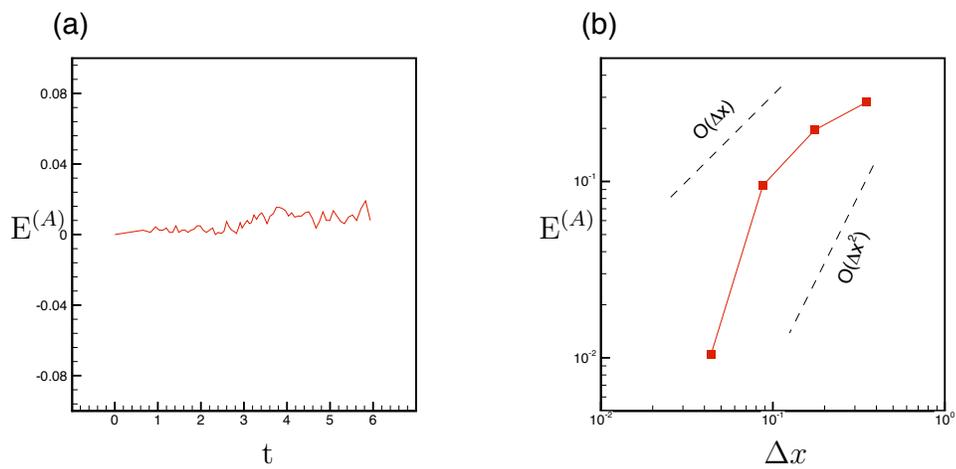


Figure 4.8: Accuracy of numerical simulation of rising bubble for $Re = 100$ and $We = 2$. (a) Volume conservation error as a function of time on a 200×200 grid, (b) error on different sized grids at $t = 6$.

Chapter A: Geometric projection in 3D

The projection of interface topology in 3-D follows essentially the same general procedure as for 2-D projection. The first step is the localization of the interface on the edges of the cells formed by connecting vertices that represent nodal values of the scalar field. The P_1 projection is shown in Figs. A.1(a) and A.2(a) where the shortest distance from the grid node \mathbf{x} to the interface point \mathbf{x}_p is represented by \mathbf{d}_1 . The second projection level P_2 requires the identification of flat interface panels in each of the four cells connected to the edge associated with \mathbf{x}_p , as shown in Figs. A.1(b) and A.2(b). Panel edges are formed by connecting interface points that share a common cell face. A sequence of such connected points is shown in Figs. A.1(b) and A.2(b) for the green panels. The projection of \mathbf{d}_2 normal to a panel represents the P_2 projection.

Note that there are always only four panels that are connected to the first point \mathbf{x}_p and the number of interface markers in each of these cells depends on how the interface intersects the cell. There are five possible arrangements of interface panels that can be formed with a minimum of three and a maximum of six interface markers, as shown in A.3. Similar to the 2-D case, a cell face cannot have more than two interface markers when a single interface intersects cell. Because a single

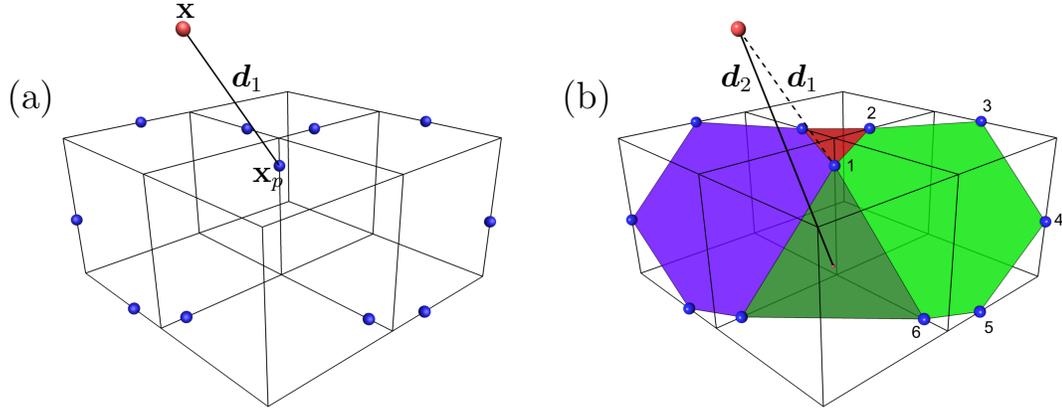


Figure A.1: (a) P_1 projection in 3-D. The distance d_1 is the shortest distance from the grid point \mathbf{x} to the interface marker \mathbf{x}_p . (b) P_2 projection. The scalar field resides on cell vertices.

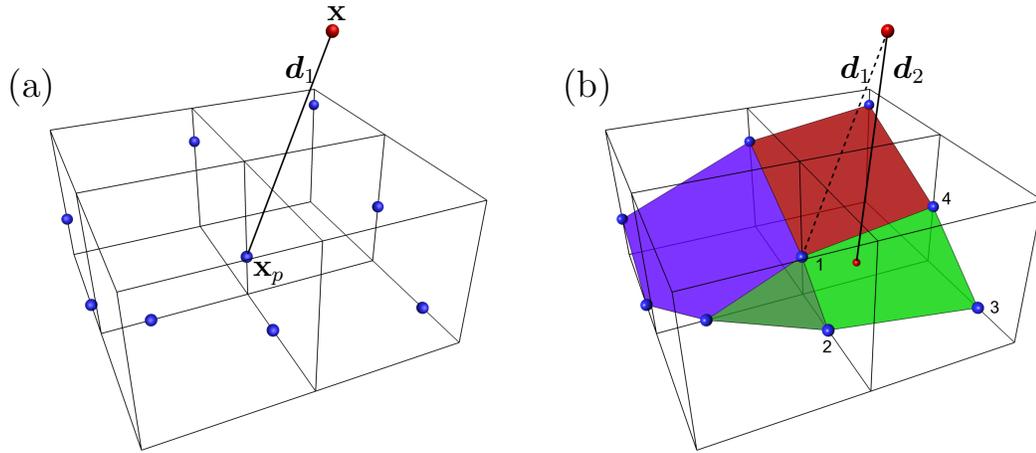


Figure A.2: P_1 and P_2 projections for a different interface configuration. The interface marker \mathbf{x}_p is always connected to four panels.

cell cannot resolve more than one interface, the presence of more than two interface markers representing a merging or breakup of an interface is trivial and does not require additional treatment.

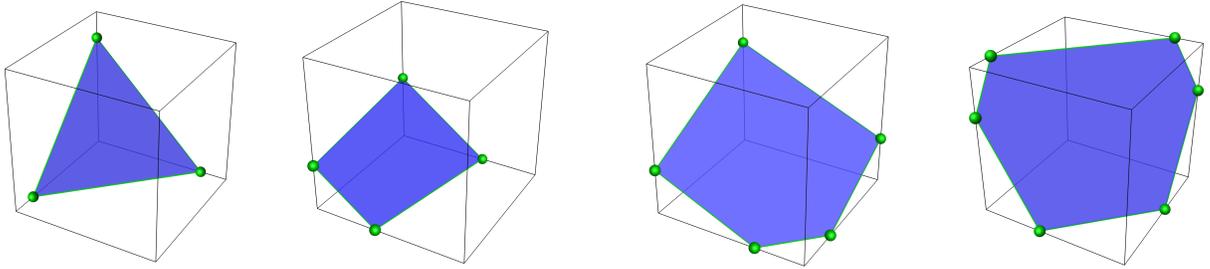


Figure A.3: Four possible configurations of interface panels based on 3,4,5 and 6 interface markers within a cell.

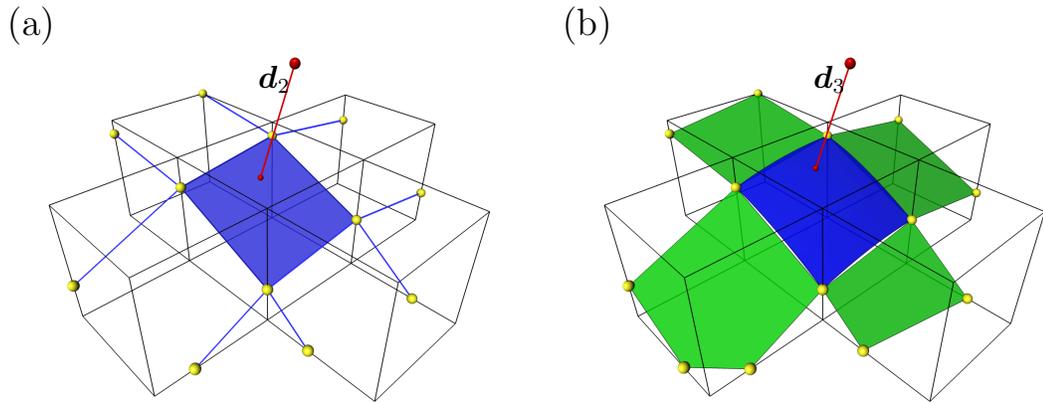


Figure A.4: P_3 projection. (a) Interface markers connected to P_2 panel. (b) Projection to the smooth interface.

The panel selected by P_2 projection is then used along with interface markers in neighboring cells to construct a smooth surface for P_3 projection. Figure A.4(a) shows a 4 point panel that is connected to interface markers in four connected cells. The slopes associated with each edge, along with the 4 nodal locations are used to provide 12 boundary conditions for the construction of a smooth surface $z = f(x, y)$,

as shown in Fig. A.4(b).

Bibliography

- [1] 1963, O. A. L. (1963). The mathematical theory of viscous incompressible flow. *Gordon and Breach*.
- [2] Adalsteinsson, D., & Sethian, J. (1999). The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, *148*(1), 2–22.
- [3] Alder, B. (1964). Methods in computational physics. *Quantum Mechanics*, *2*.
- [4] Amsden, A. A. (1966). Report. *LA-3466, Los Alamos Scientific Laboratory*.
- [5] Andrea Prosperetti, G. T. (2007). Computational methods for multiphase flow. *Cambridge University Press*.
- [6] Braza, M. (1981). These docteur-ingenieur. *I. N. P. Toulouse*.
- [7] C. S. Peskin, B. F. P. (1993). Improved volume conservation in the computation of flows with immersed elastic boundaries. *Journal of Computational Physics*, *105*, 33–46.
- [8] C. S. Peskin, D. M. M. (1989). A three- dimensional computational method for blood flow in the heart i. immersed elastic fibers. *Journal of Computational physics*, *81*, 372–405.
- [9] C. W. Hirt, B. D. N. (1981). Volume of fluid method for the dynamics of free boundaries. *Journal of Computational physics*, *39*, 201–225.
- [10] C.-W. Shu, S. O. (1989). Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii. *Journal of Computational Physics*, *83*, 32–78.
- [11] Chorin, A. J. (1966). Aec research and development report no. nyo-1480-61, new york university.
- [12] Chorin, A. J. (1967). A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, *2*, 12–26.

- [13] Chorin, A. J. (1968). Numerical solution of the navier-stokes equations. *Mathematics of Computation*, *22*, 745–762.
- [14] Chorin, A. J. (1969). On the convergence of discrete approximations to the navier-stokes equations. *Mathematics of Computation*, *23*, 341–353.
- [15] Colella, J. B. B. . P. (1989). A second-order projection method for the incompressible navier-stokes equations. *Journal of Computational Physics*, *85*, 257–283.
- [16] D. M. McQueen, C. S. P. (1989). A three- dimensional computational method for blood flow in the heart ii. contractile fibers. *Journal of Computational physics*, *82*, 289–297.
- [17] Daly, B. J. (1967). Numerical study of two fluid rayleigh-taylor instability. *The Physics of Fluids*, *10*(2), 287–297.
- [18] Daly, B. J. (1969). Numerical study of effect of surface tension on interface instability. *The Physics of Fluids*, *12*(7), 1340–1354.
- [19] Danielsson, P. (1980). Euclidean distance mapping. *Computer Graphics and Image Processing*, *14*(3), 227–248.
- [20] Debar, R. (1974). Fundamentals of the kraken code. *E. O. Lawrence Livermore National Laboratory Report*, UCIR-760.
- [21] Delaney, K., Vanella, M., Balaras, E., Riaz, A., & Qin, Z. (2014). A Two-Phase, Incompressible Flow Solver with Adaptive Mesh Refinement. Under review. *Journal of Computational Physics*.
- [22] Deng, S. Z., Ito, K., & Li, Z. L. (2003). Three-dimensional elliptic solvers for interface problems and applications. *JOURNAL OF COMPUTATIONAL PHYSICS*, *184*(1), 215–243.
- [23] Desjardins, O., Moureau, V., & Pitsch, H. (2008). An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, *227*(18), 8395–8416.
- [24] Desjardins, O., & Pitsch, H. (2009). A spectrally refined interface approach for simulating multiphase flows. *Journal of Computational Physics*, *228*(5), 1658–1677.
- [25] Ding, H., Spelt, P. D., & Shu, C. (2007). Diffuse interface model for incompressible two-phase flows with large density ratios. *Journal of Computational Physics*, *226*(2), 2078–2095.
- [26] E. Rouy, A. T. (1992). A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.*, *29*(3), 867–884.

- [27] Francis H, J. E. W., Harlow (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8(12), 2182–2189.
- [28] G. Tryggvason, H. A. (1985). Finger-interaction mechanisms in stratified heleshaw flow. *J. Fluid Mech*, 154, 287–301.
- [29] Guermond, J.-L., & Quartapelle, L. (2000). A projection fem for variable density incompressible flows. *Journal of Computational Physics*, 165(1), 167–188.
- [30] H. D. Ceniceros, A. M. R. (2010). A robust, fully adaptive hybrid level-set/front-tracking method for two-phase flows with an accurate surface tension computation. *Commun. Comput. Phys*, 8(1), 51–94.
- [31] H. Fujita, T. K. (1964). On the navier-stokes initial value problem. i. *Archive for Rational Mechanics and Analysis*, 16(4), 269–315.
- [32] H. Terashima, G. T. (2010). A front-tracking method with projected interface conditions for compressible multi-fluid flows. *Computers Fluids*, 39, 1804–1814.
- [33] J. B. Bell, D. L. M. (1992). A second-order projection method for variable-density flows. *Journal of Computational Physics*, 101, 334–348.
- [34] J. B. Bell, G. R. S., C. N. Dawson (1988). An unsplit, higher order godunov method for scalar conservation laws in multiple dimensions. *Journal of Computational Physics*, 74, 1–24.
- [35] J. B. Bell, P. C. (1989). A second-order projection method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 85, 257–283.
- [36] J. C. Phillips, L. K. (1959). New method for calculating wave functions in crystals and molecules. *Phys. Rev*, 116(287).
- [37] J. Glimm, B. L. O. A. M., J. Grove, & Tryggvason, G. (1988). The bifurcation of tracked scalar waves. *SIAM J. Sci. and Stat. Comput.*, 9(1), 61–79.
- [38] Jingyi Zhu, J. S. (1992). Projection methods coupled to level set interface techniques. *Journal of Computational Physics*, 102, 128–138.
- [39] Johansen, H., & Colella, P. (1998). A cartesian grid embedded boundary method for poissons equation on irregular domains. *Journal of Computational physics*, 147, 60.
- [40] Kan, J. V. (1986). A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM J. Sci. and Stat. Comput.*, 7(3), 870–891.

- [41] Kang, M., Fedkiw, R. P., & Liu, X.-D. (2000). A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3), 323–360.
- [42] Kim, J., & Mion, P. (1985). Application of a fractional-step method to incompressible navier-stokes equations. *Journal of Computational Physics*, 59, 308–323.
- [43] Kzivickii, A., & Ladyzhenskaya, O. A. (1966). The method of nets for the non-stationary navier-stokes equations. *Proc. Steklov Inst.*, 92, 105–112.
- [44] Leer, B. V. (1984). On the relation between the upwind-differencing schemes of godunov, engquistosher and roe. *SIAM J. Sci. and Stat. Comput.*, 5(1), 1–20.
- [45] Li, Z. K., & Ito, K. (2001). Maximum principle preserving schemes for interface problems with discontinuous coefficients. *SIAM JOURNAL ON SCIENTIFIC COMPUTING*, 23(1), 339–361.
- [46] Li, Z. L. (1998). A fast iterative algorithm for elliptic interface problems. *SIAM JOURNAL ON NUMERICAL ANALYSIS*, 35(1), 230–254.
- [47] M. Sussman, P. S. S. O., E. Fatemi (1998). An improved level set method for incompressible two-phase flows. *Computers Fluids*, 27(5-6), 663–680.
- [48] Mittal, R., & Iaccarino, G. (2005). Immersed boundary method. *Annu. Rev. Fluid Mech.*, 37(239-61).
- [49] Noh, W. F. (1964). Cel: A time-dependent two-space-dimensional coupled eulerian-lagrangian code. *Methods in Computational Physics 3*, Academic Press, New York.
- [50] Olsson, E., & Kreiss, G. (2005). A conservative level set method for two phase flow. *Journal of Computational Physics*, 210(1), 225–246.
- [51] Olsson, E., Kreiss, G., & Zahedi, S. (2007). A conservative level set method for two phase flow II. *Journal of Computational Physics*, 225(1), 785–807.
- [52] P. Colella, H. G. (1983). Efficient algorithms for the solution of the riemann problem for real gases. *Lawrencw Berkeley Laboratory Report LBL-15776*.
- [53] Peskin, C. S. (1972). Flow patterns around heart valves: A numerical method. *Journal of Computational physics*, 10, 252–271.
- [54] Peskin, C. S. (1977). Numerical analysis of blood flow in the heart. *Journal of Computational physics*, 25, 220–252.
- [55] Peskin, C. S. (1980). Modeling prosthetic heart valves for numerical analysis of blood flow in the heart. *Journal of Computational physics*, 37, 113–132.

- [56] Qin, Z., Delaney, K., Riaz, A., & Balaras, E. (2014). A second order accurate jump condition capturing scheme for sharp interfaces on Cartesian Grids. Under review. *Journal of Computational Physics*.
- [57] Qin, Z., Delaney, K., Riaz, A., & Balaras, E. (2014). Topology Preserving Advection of Implicit Interfaces on Cartesian Grids. Under review. *Journal of Computational Physics*.
- [58] R. Fedkiw, B. M. S. O., T. Aslam (1999). A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys*, *152*, 457–492.
- [59] R. Fedkiw, S. X., T. Aslam (1999). The ghost fluid method for deflagration and detonation discontinuities. *J. Comput. Phys*, *154*, 393–427.
- [60] R. Fedkiw, X. D. L., and (1998). The ghost fluid method for viscous flows. *M. Hafez, Ed.*
- [61] R. M. Beam, R. F. W. (1976). An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *Journal of Computational Physics*, *22*, 87–110.
- [62] Richtmyer, M. K., R.D. (1967). Difference methods for initial value problems. *2nd edition, Interscience, New York.*
- [63] R.J. LeVeque, Z. L. (1994). The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *Society for Industrial and Applied Mathematics*, *31*, 1019–1044.
- [64] RL Leveque, J. O. (1981). Numerical analysis project. *Manuscript NA-81-16 Computer Science Department, Stanford University.*
- [65] Rudman, M. (1997). Volume-tracking methods for interfacial flow calculations. *International Journal For Numerical Methods In Fluids*, *24*, 671–691.
- [66] Russo, G., & Smereka, P. (2000). A remark on computing distance functions. *Journal of Computational Physics*, *163*(1), 51–67.
- [67] S Osher, J. S. (1988). Fronts propagating with curvature-dependent speed - algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, *79*(1), 12–49.
- [68] Sheu, T. W., Yu, C., & Chiu, P. (2011). Development of level set method with good area preservation to predict interface in two-phase flows. *International Journal for Numerical Methods in Fluids*, *67*(1), 109–134.
- [69] Sheu, T. W. H., Yu, C. H., & Chiu, P. H. (2009). Development of a dispersively accurate conservative level set scheme for capturing interface in two-phase flows. *Journal of Computational Physics*, *228*(3), 661–686.

- [70] Sussman, E. G. P. (2000). A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *Journal of Computational physics*, 162, 301–337.
- [71] Sussman, J. B. B., A. S. Almgren (1999). An adaptive level set approach for incompressible two-phase flows. *Journal of Computational physics*, 148, 81–124.
- [72] Sussman, M., & Fatemi, E. (1999). An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing*, 20(4), 1165–1191.
- [73] Sussman, M., Smereka, P., & Osher, S. (1994). A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1), 146–159.
- [74] Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S., & Jan, Y.-J. (2001). A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169(2), 708–759.
- [75] Unverdi, S., & Tryggvason, G. (1992). A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1), 25–37.
- [76] W. F. Noh, P. W. (1976). Slic (simple line interface calculation). *The Physics of Fluids*, 59(7), 330–340.
- [77] W. Mulder, S. O. (1992). Computing interface motion in compressible gas dynamics. *Journal of Computational Physics*, 100, 209–228.
- [78] W. R. Briley, H. M. (1977). Solution of the multidimensional compressible navier-stokes equations by a generalized implicit method. *Journal of Computational Physics*, 24, 372–397.
- [79] Xu-Dong Liu, M. K., Ronald P. Fedkiw (1999). A Boundary Condition Capturing Method for Poisson’s Equation on Irregular Domains. *J. Comput. Phys*, 160, 151–178.
- [80] Zalesak, S. T. (1979). Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational physics*, 31, 335–362.