Finite-element simulations of interfacial flows with moving contact lines

Jiaqi Zhang

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mathematics

Pengtao Yue, Chair
Jeff Borggaard
Tao Lin
Tim Warburton

May 8, 2020
Blacksburg, Virginia

Keywords: level set, contact angle hysteresis, contact line pinning, slip length, drop spreading, sliding drop, GNBC, contact line friction.

Finite-element simulations of interfacial flows with moving contact lines

Jiaqi Zhang

(ABSTRACT)

In this work, we develop an interface-preserving level-set method in the finite-element framework for interfacial flows with moving contact lines. In our method, the contact line is advected naturally by the flow field. Contact angle hysteresis can be easily implemented without explicit calculation of the contact angle or the contact line velocity, and mesh-independent results can be obtained following a simple computational strategy. We have implemented the method in three dimensions and provide numerical studies that compare well with analytical solutions to verify our algorithm.

We first develop a high-order numerical method for interface-preserving level-set reinitialization. Within the interface cells, the gradient of the level set function is determined by a weighted local projection scheme and the missing additive constant is determined such that the position of the zero level set is preserved. For the non-interface cells, we compute the gradient of the level set function by solving a Hamilton-Jacobi equation as a conservation law system using the discontinuous Galerkin method. This follows the work by Hu and Shu [SIAM J. Sci. Comput. 21 (1999) 660-690]. The missing constant for these cells is recovered using the continuity of the level set function while taking into account the characteristics. To treat highly distorted initial conditions, we develop a hybrid numerical flux that combines the Lax-Friedrichs flux and a penalty flux. Our method is accurate for non-trivial test cases and handles singularities away from the interface very well. When derivative singularities

are present on the interface, a second-derivative limiter is designed to suppress the oscillations. At least $(N+1)$th order accuracy in the interface cells and Nth order accuracy in the whole domain are observed for smooth solutions when Nth degree polynomials are used. Two dimensional test cases are presented to demonstrate superior properties such as accuracy, long-term stability, interface-preserving capability, and easy treatment of contact lines.

We then develop a level-set method in the finite-element framework. The contact line singularity is removed by the slip boundary condition proposed by Ren and E [Phys. Fluids, vol. 19, p. 022101, 2007], which has two friction coefficients: β_N that controls the slip between the bulk fluids and the solid wall and β_{CL} that controls the deviation of the microscopic dynamic contact angle from the static one. The predicted contact line dynamics from our method matches the Cox theory very well. We further find that the same slip length in the Cox theory can be reproduced by different combinations of (β_N, β_{CL}). This combination leads to a computational strategy for mesh-independent results that can match the experiments. There is no need to impose the contact angle condition geometrically, and the dynamic contact angle automatically emerges as part of the numerical solution. With a little modification, our method can also be used to compute contact angle hysteresis, where the tendency of contact line motion is readily available from the level-set function. Different test cases, including code validation and mesh-convergence study, are provided to demonstrate the efficiency and capability of our method.

Lastly, we extend our method to three-dimensional simulations, where an extension equation is solved on the wall boundary to obtain the boundary condition for level-set reinitializaiton with contact lines. Reinitialization of ellipsoidal interfaces is presented to show the accuracy and stability of our method. In addition, simulations of a drop on an inclined wall are presented that are in agreement with theoretical results.

Finite-element simulations of interfacial flows with moving contact lines

Jiaqi Zhang

(GENERAL AUDIENCE ABSTRACT)

When a liquid droplet is sliding along a solid surface, a moving contact line is formed at the intersection of the three phases: liquid, air and solid. This work develops a numerical method to study problems with moving contact lines. The partial differential equations describing the problem are solved by finite element methods. Our numerical method is validated against experiments and theories. Furthermore, we have implemented our method in three-dimensional problems.

Dedication

To my parents and my wife.

Acknowledgments

I wish to express my deepest gratitude to my advisor, Professor Pengtao Yue, for his inspiring guidance, constant support, and encouragement throughout the entire period of my PhD study. I really grew as an individual and as a researcher working with him. His meticulous scrutiny and scientific approach was significant and have helped me make steady progress in this work.

I would like to thank the rest of my thesis committee: Professor Jeff Borggaard, Professor Tao Lin, and Professor Timothy Warburton, for sharing all your knowledge and insightful comments.

I wish to thank my parents for their encouragement and unconditional support. Most importantly, I would like to thank my wonderful wife and best friend, Shanshan Liang, who is always by my side, providing help and support.

This work would not have been possible without the support of the Math Department at Virginia Tech and the National Science Foundation (Grant DMS-1522604). The coding in this work is based on the open source finite element library deal.II and I would like to thank all of its developers. The computational and technical contribution of the Advanced Research Computing at Virginia Tech is truly appreciated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Moving contact lines

When a liquid droplet is resting on an isotropic, homogeneous, and smooth ideal solid surface, Young's equation [4] gives the relation between the contact angle and the surface tensions. Here the contact angle is defined as the angle where the liquid–air interface meets the solid-liquid interface. While in stationary equilibrium, the contact angle is usually referred to as the static contact angle θ_S. However, when the contact line, which is formed at the intersection of the three phases, is in motion, Young's equation fails and the dynamics at the moving contact becomes much more complicated than it is in static equilibrium. The moving contact line is ubiquitous in nature, for example, sliding rain drops on windshield glasses. Industrial applications include inkjet printing, self-cleaning surfaces, coating, lubrication, and microfluidics, among others. Despite decades of research, the underlying physics has not been well understood. Huh and Scriven [5] revealed the stress singularity at the moving contact line by studying the corner flow described by the Stokes equations with the no-slip boundary condition on the solid wall. Therefore, to address the singularity, one has to allow slip at the moving contact line, and the slip length is typically at the microscopic scale (typically 10^{-9}m), indicating the multiscale characteristic of the moving contact line problem. Based on the length scale, two contact angles are defined: the dynamic contact angle θ_D at the microscopic scale and the apparent contact angle θ_{APP} at the macroscopic

scale. By exploring the asymptotic limits of the lubrication approximation, Voinov [6] showed the existence of viscous bending at the liquid-air interface: the contact angle varies with the distance to the contact line logarithmically. Later the Cox theory [7] was proposed to describe the relation between the macroscopic and microscopic contact angles for fluids of arbitrary viscosity ratios. The theoretical results in [6, 7] have a great impact on designing numerical methods for moving contact line problems.

In past two decades, numerical simulations of moving contact lines have gained increasing interest, providing an alternative to theoretical and experimental methods in studies of the dynamic behavior of moving contact lines. This dissertation focuses on continuum numerical simulations, in which one of the challenges is, as we mentioned before, the stress singularity at the contact line caused by the discrepancy between the no-slip boundary condition and the moving contact line. Different models have been developed to relax the stress singularity. For example, the Navier slip model [8] relates the slip velocity to the viscous stress only, and does not incorporate the information about the contact angle. However, the Molecular Kinetic Theory (MKT) [9] indicates that the contact line velocity is related to the contact angles, specifically, the difference between the dynamic contact angle and the static contact angle. The slip model that includes both the contact angles and the viscous stress is the Generalized Navier Boundary Condition (GNBC) by Qian *et al.* [10], who observed that slip velocity at the wall was proportional to the sum of the tangential viscous stress and the uncompensated Young's stress (a.k.a. the unbalanced Young's stress) in molecular dynamics (MD) simulations. The velocity profiles in the vicinity of the contact line from their phase-field simulations agreed very well with the MD results. To improve upon the Navier slip condition, Ren and E later developed a sharp-interface version of the slip condition [11] by adding a contact line condition based on the force balance between the contact line friction and the unbalanced Young's stress, which is no longer restricted to the phase-field method.

Note that the unbalanced Young's stress is caused by the deviation of the dynamic contact angle θ_D from the static contact angle θ_S. We chose Ren and E's slip condition in this work. Other methods to regularize the contact line singularity include diffusion in different models, such as the Cahn-Hilliard model [12], and the conservative level set method [13]. More details on numerical methods for the moving contact line problem can be found in a recent review paper [14].

Although the slip models have been widely used, it is still challenging to obtain mesh-independent results, because the physical slip length is usually at the nanoscale and can not be resolved by the computational mesh. In the literature, a standard treatment to remove mesh dependency, as proposed in [15, 16, 17, 18], is to determine a numerical contact angle at the grid scale based on mascroscale models such as the Cox-Voinov model [6, 7]; this numerical angle is then applied at the contact line in place of the static contact angle. A drawback of this method is that it requires the input of contact line velocity, which may be difficult to obtain, especially in three dimensions. The similar idea was also used in the GNBC, however, in a different approach [19, 20]: the grid-scale contact angle from the simulation is used to determine a microscopic dynamic contact angle, which is then fed to the GNBC to compute the slip velocity. In this work, we propose a novel approach that does not rely on hydrodynamic models and is thus much easier to implement. In addition, by properly choosing the friction coefficients, we show that the slip condition by Ren and E itself is sufficient to reproduce the well-established Cox theory [7] with realistic slip lengths.

1.2 Contact angle hysteresis

One interesting topic associated with contact lines is the contact angle hysteresis (CAH). Due to chemical or physical defects of the solid surface, the static contact angle is not unique,

resulting in the CAH that is defined by the difference between the greatest and the smallest static contact angles. The greatest static contact angle is called the advancing contact angle (θ_A), while the smallest one is called the receding contact angle (θ_R). When the contact angle exceeds θ_A, the contact line advances; while the contact line recedes if the contact angle is smaller than θ_R; otherwise, the contact line is pinned. An easy example of CAH is the sticking of drops to an inclined solid surface. Because of CAH, there is a net force generated at the contact line that balances the gravity, and hence the drop can be stationary.

Due to its practical value, there has been a sizable literature exploring the underlying physics of contact angle hysteresis, e.g., [21, 22, 23, 24]. It should be noted that contact angle hysteresis is observed even on inherently smooth surfaces [25, 26]. Thus contact angle hysteresis should be considered in almost all numerical simulations of moving contact lines. In this dissertation, we do not intend to explore the origin of contact angle hysteresis. Instead, we focus on the simulation of contact line dynamics with θ_A and θ_R specified from experiments or other sources. Spelt developed an iterative procedure to implement contact angle hysteresis in the level-set method [27]. This procedure was later followed by different researchers using different methods including the phase-field method [28, 29] and the volume-of-fluid method [30, 31, 32]. However, this treatment requires ghost points outside the boundary and the evaluation of contact angle and contact-line velocity. This can be very challenging on curved boundaries and unstructured meshes. Thus there is a demand to develop a more efficient method that is easy to implement.

1.3 Descriptions of the interface

Another challenge in simulations of interfacial flows is the representation of the interface, as the interface may undergo large deformations and topological changes, such as break-up

and coalescence. There are two popular groups of methods: interface-tracking and interface-capturing. In the first group, the interface is explicitly represented by either the cell edges (faces in 3D) of a moving mesh or markers on a stationary mesh. Numerical methods on a moving mesh usually rely on the Arbitrary Lagrangian-Eulerian (ALE) framework [33, 34, 35], where a mesh is generated to fit the interface and the two phases are computed separately. Though the method can represent the interface and handle different matching conditions very well, remeshing and interpolation are required when the interface-fitted mesh gets too distorted. In addition, the method falls short when there are topological transitions. To avoid moving the mesh, one can use markers to track the interface on a stationary mesh, e.g., the front tracking methods [36, 37, 38]. Similar to the remeshing in the previous category, this method requires redistribution of markers as the interface deforms, an it can be very complicated to capture topological changes as well. To deal with the moving contact line problem, slip models can be embedded into interface-tracking methods, such as the Navier slip and the GNBC in front-tracking methods [19, 38], and the GNBC in the ALE method [35].

The most popular interface-capturing methods, wherein the interface is represented implicitly by a scalar field, include the volume-of-fluid (VOF) method [39, 40], the level-set method [41, 42], and the phase-field method [43]. The VOF method captures the interface by tracking the volume fraction in each cell. Then the interface is reconstructed cellwise based on the volume fraction, in which the interface can be approximated by piecewise constants [39], piecewise linear functions [44], piecewise quadratic functions [45], or piecewise cubic splines [46]. This reconstruction could be very complicated and usually leads to a discontinuous interface. In VOF methods, the system of equations are usually discretized on a staggered grid and the velocity is placed on edge centers, leading to a numerical slip length which is half of the grid size [15, 40]. Thus using the no-slip boundary condition does not result in

the contact line singularity. However, introducing slip models in the VOF method provides another way to predict the contact line dynamics [47, 48]. Although the VOF method conserves mass very well, it is not easy to implement on an unstructured mesh.

Due to its simplicity and ability to capture topological changes of the interface, the level-set method has been applied to a lot of areas in computational physics and engineering [42, 49, 50, 51]. More recent advances and applications of the level-set method can be found in a recent review by Gibou *et al.* [52] and references therein. Generally, the level-set function ϕ is initialized to a signed distance function, which satisfies the Eikonal equation $|\nabla\phi| = 1$. Nevertheless, it will usually become too steep or too flat during the evolution of the level set function, thus causing large errors in simulations. Therefore, it is inevitable that ϕ needs to be reinitialized to a signed distance function while still preserving the position of the interface. However, this is not easy to achieve numerically, and the subsequent mass loss is the major issue of the level-set method. To address the stress singularity, slip boundary conditions for the flow equations are usually used in the level-set method [27, 53]. Note that level-set methods with interface-preserving reinitialization may have trouble capturing topological transitions of the interface. For instance, in the simulations of a drop impacting on a solid wall, we found that the interface cannot merge with the solid wall automatically, forming a thin air film that never ruptures.

All aforementioned methods use a sharp-interface description, which means the interface thickness is zero. Although, for numerical stability, some of them may need to introduce an interface thickness of the order of the mesh size. The phase-field method, also known as the diffuse-interface method, has a finite-thickness interface built in the physical model. Therefore, physical properties changes steeply but smoothly across the diffuse interface. To conserve mass, the Cahn-Hilliard equation is usually used to evolve the phase-field variable. Compared with the sharp-interface methods, the phase-field method can handle topologi-

cal changes automatically. In addition, the intrinsic Cahn-Hilliard diffusion can relax the contact-line singularity [12, 54]. Since the introduction of GNBC in the phase-field framework in [10], the GNBC-based phase-field methods has gained increasing popularity in simulations of moving contact lines [55, 56, 57, 58, 59], which also encourages the development of efficient energy stable numerical schemes [60, 61, 62, 63]. However, the phase-field method is computationally expensive, since it requires a comparatively thick layer of fine mesh to accurately resolve the interface.

1.4 Thesis outline

The ultimate goal of this work is to develop a numerical method, which is efficient and reliable, to simulate interfacial flows with moving contact lines. There are several challenges that are unique to this field. One is the accurate representation of the interface as well as capturing topological transitions of the interface. A second is the stress singularity at the contact line, which may cause mesh-dependent results if not handled appropriately. A third is the implementation of the contact angle hysteresis. We focus on these three issues in this work. Other challenges, such as the restrictive time steps and the parasitic currents near the interface caused by the explicit treatment of the surface tension, are not addressed in the current work.

We employ the level-set method to represent the interface implicitly. One concern about the level-set method is the unphysical mass loss caused mainly by the procedure of reinitialization. Therefore we first address level-set reinitialization in Chapter 2, where we propose a high-order and interface-preserving level-set reinitialization technique in the discontinuous Galerkin (DG) framework in Sections 2.2 and 2.3. Boundary conditions for level-set reinitialization in 2D are discussed in Section 2.4.6.

When the level-set reinitialization method is ready, we couple the level-set equation with the Navier-Stokes equations in Chapter 3. The weak form of the Navier-Stokes equations with Ren and E's slip condition is given in this chapter. In addition, a simple approach to implement contact angle hysteresis is developed without explicit computation of the contact angle or the contact line velocity.

In Chapter 4, we extend the level-set method to three-dimensional simulations. Special attention is paid to the level-set reinitialization on the wall boundary, where an extension equation is solved to supply the boundary condition. Simulations of a drop on an tilted wall are conducted to illustrate the capability of our method in modeling moving contact line problems with CAH.

We present a brief summary of this work in Chapter 5.

Chapter 2

Level-set reinitialization

This chapter is devoted to level-set reinitialization that leads to a singed distance function. To prevent mass loss, we propose a weighted local projection method that computes $\nabla\phi$ in part of the interface cells, and only use the PDE-based method in all the other cells. In contrast to traditional methods in the literature, we solve for the gradient $\nabla\phi$ first and then reconstruct ϕ by the interface location and the continuity of ϕ. The work in this chapter has been published in [64].

2.1 Introduction

A lot of methods have been proposed for level-set reinitialization, among which solving the Eikonal equation using some fast method is a popular one, e.g., the fast marching methods [65, 66] and the fast sweeping methods [67, 68]. An alternative is to compute the Hamilton-Jacobi (HJ) (2.1) equation to steady state such that $|\nabla\phi| = 1$ [42]:

$$\phi_\tau + H(\nabla\phi) = 0, \text{ in } \Omega \times [0,T] \subset \mathbb{R}^n \times \mathbb{R}, \quad \phi(\boldsymbol{x}, 0) = \phi_0, \tag{2.1}$$

where $H(\nabla\phi) = S(\phi_0)\,(|\nabla\phi| - 1)$, S is the sign function, ϕ_0 is the initial level-set function, and τ is the pseudo time. Following the literature, we replace the discontinuous sign function

S by the smoothed sign function given as follows to stabilize the solution,

$$S_\eta(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \eta^2}}, \tag{2.2}$$

where η is the smoothing parameter usually chosen to be the smallest mesh size $h_{\min}$. Our reinitialization algorithm is built on top of this PDE-based method, the applications of which can be found in [69, 70, 71, 72]. Another category of methods are based on variational formulations. The first variational level-set method, which was proposed Li *et al.* [73], attains $|\nabla\phi| = 1$ by minimizing an energy functional. Basting and Kuzmin [74] modified the energy functional and extended this method to the elliptic reinitialization, in which the elliptic problem can be solved by a Ritz-Galerkin finite element method or a discontinuous Galerkin (DG) method [75].

A consensus in the level-set literature is that directly solving the HJ equation (2.1) results in displacement of the interface, leading to unphysical mass loss in numerical simulations of interfacial flows. Therefore a lot of work has been done to improve mass conservation during reinitialization. For instance, Sussman *et al.* [76, 77] added a constraint to the HJ equation by the Lagrange multiplier to preserve mass. A variation of the smooth sign function was proposed by Peng *et al.* [78] to keep the shift the interface within one cell in the process of reinitialization. Techniques of subcell fix, first introduced by Russo and Smereka [79] and improved by Min [80] and du Chéné *et al.* [81], were developed to prevent the shift of the interface as well. Hartman *et al.* [82, 83] proposed a constrained reinitialization scheme that solves a least-squares problem to compute the level-set function in each interface cell. Sophisticated methods are also proposed to correct the level-set function by other mass conserving techniques, such as the particle level-set method [84] and the coupled level-set/volume-of-fluid method [85, 86, 87]. It should be noted that the conservative level-set method [88, 89] proposed by Olsson *et al.* takes a different approach to conserve mass and

it is more like a variant of the phase-field method [43, 90]. Some recent developments of this method can be found in [91, 92].

Most high order methods for level-set reinitialization are based on structured meshes, e.g., [68, 81, 93]. To achieve high order accuracy on unstructured meshes, DG methods, which have been very successful in solving conservation laws [94, 95, 96], seem to be the top choice. Although Zhang and Shu [97] and Levy *et al.* [98] have successfully constructed finite-volume Weighted Essentially Non-Oscillatory (WENO) schemes for the HJ equation on unstructured meshes, DG methods still have the advantages of compactness, easy implementation, and superior scalability. Since the gradient of the HJ equation form a system of conservation laws, DG methods can be readily adapted to solve the HJ equation. Following this idea, Hu and Shu [99] designed the first DG method for the HJ equation, which was later reinterpreted and simplified by Li and Shu [100]. Later, different DG methods are proposed to directly solve the HJ equation [101, 102]. A recent review on DG methods for HJ equations can be found in [103]. In literature, the particular HJ equation (2.1) for level-set reinitialization was however mostly solved by the finite difference methods or the finite volume methods. Sometimes, people still stick to the more mature finite volume methods for (2.1) even though they use the DG methods for other equations. For example, Fechter and Munz used a fifth-order WENO scheme in the finite volume subcells of each DG grid cell [104]; Marchandise *et al.* completely avoided reinitialization and relied on special algorithms that did not require ϕ to be a signed distance function [105]. There are only a few successful implementations of DG methods, both of which add an additional second-order diffusion term to the right-hand side of (2.1) and use some filtering technique to stabilize the solution [106, 107].

We develop an interface-preserving DG method for (2.1). The computation cells are divided into interface cells and non-interface cells and the solution of ϕ is decomposed into $\nabla\phi$ and an additive constant. In the interface cells, we construct $\nabla\phi$ using a weighted local projection

method and determine the additive constant such that the interface location is preserved. In the non-interface cells, instead of solving (2.1), we follow Hu and Shu [99] and rewrite it as a conservation law system by taking the gradient:

$$\frac{\partial \nabla \phi}{\partial \tau} + \nabla H(\nabla \phi) = 0, \text{ in } \Omega \times [0, T] \subset \mathbb{R}^n \times \mathbb{R}, \quad \nabla \phi(\boldsymbol{x}, 0) = \nabla \phi_0. \tag{2.3}$$

It should be noted that components of $\nabla\phi$ are not completely independent, e.g., $\nabla \times \nabla \phi = 0$ is always satisfied. We then recover ϕ based on continuity. Our method is very stable and does not need additional diffusion terms or filtering techniques. For smooth ϕ with piecewise Nth degree polynomial space, we can achieve $(N+1)$th order accuracy in the interface cells and Nth order in the whole domain. An additional benefit is that our method can be directly applied to moving contact line simulations without complicated treatments on the boundary [27, 106].

This chapter is organized as follows: we first describe the algorithm to compute $\nabla\phi$ in both interface and non-interface cells in Section 2.2. In addition, a novel hybrid numerical flux and a second-derivative limiter are developed for the DG method. In Section 2.3, we present the interface-preserving reconstruction of level-set function in the interface cells and non-interface cells. Numerical results are illustrated in Section 2.4.

2.2 Computation of $\nabla\phi_h$

In this section, we develop a high-order and stable method to compute $\nabla\phi$. Computational cells are divided into interface cells and non-interface cells. In the interface cells, we anchor $\nabla\phi$ by the weighted local projection method. We then compute $\nabla\phi$ in all the other cells by the DG method.

2.2.1 Interface cells

We denote the zero level set by

$$\Gamma = \{\boldsymbol{x} \in \Omega : \phi_0(\boldsymbol{x}) = 0\}, \tag{2.4}$$

and the set of interface cells by

$$I = \{K \in \mathcal{T}_h : K \cap \Gamma \neq \emptyset\}. \tag{2.5}$$

Theoretically, $\nabla\phi$ remains normal to the interface and propagates away from the interface along $S_\eta(\phi_0)\nabla\phi$ when we evolve (2.3). Let $\mathbf{t}_\phi$ be the unit tangent vector to the interface Γ, then $\mathbf{t}_\phi \cdot \nabla\phi_0 = 0$ at $\boldsymbol{x} \in \Gamma$. Multiplying (2.3) by $\mathbf{t}_\phi$, we have

$$\mathbf{t}_\phi \cdot \frac{\partial \nabla\phi}{\partial \tau} + \mathbf{t}_\phi \cdot \nabla H(\nabla\phi) = 0. \tag{2.6}$$

Since $S(\phi_0) = 0$ for any $\boldsymbol{x} \in \Gamma$, Γ is also the zero level set of $H(\nabla\phi)$. Consequently, $\mathbf{t}_\phi \cdot \nabla H(\nabla\phi) = 0$ and $\frac{\partial(\mathbf{t}_\phi \cdot \nabla\phi)}{\partial\tau} = \mathbf{t}_\phi \cdot \frac{\partial \nabla\phi}{\partial\tau} = 0$ on Γ, where we have used the fact that Γ and thus $\mathbf{t}_\phi$ are independent of τ. Thus if we start with initial condition $\nabla\phi = \nabla\phi_0$, we should expect $\mathbf{t}_\phi \cdot \nabla\phi(\boldsymbol{x}, \tau) = \mathbf{t}_\phi \cdot \nabla\phi(\boldsymbol{x}, 0) = 0$ for any $\boldsymbol{x} \in \Gamma$. However, after discretization, $\frac{\partial(\mathbf{t}_\phi \cdot \nabla\phi)}{\partial\tau}$ could not remain exactly zero on Γ. The errors may accumulate and eventually destroy the zero level set in long-term simulations, as shown later in Sec. 2.4.2. This kind of instability is common in PDE-based method. For example, it is well-known that the interface tends to shift if the HJ equation (2.1) is evolved without any constraints [79, 108]. Thus $\nabla\phi$ in the interface cells has to be determined by a different approach and fixed during the pseudo time evolution.

Let $\nabla\phi_h$ be a signed distance function in an interface cell $K \in I$, it needs to satisfy both

$$\nabla\phi_h = \frac{\nabla\phi_0}{|\nabla\phi_0|} \quad \text{on } \Gamma \cap K, \tag{2.7}$$

and

$$|\nabla\phi_h| = 1 \quad \text{in } K. \tag{2.8}$$

Generally, these two conditions cannot be satisfied simultaneously, but we can seek the optimal approximation by minimizing the energy functional

$$E(\nabla\phi) = \frac{1}{2}\int_{\Gamma\cap K}\left(\nabla\phi - \frac{\nabla\phi_0}{|\nabla\phi_0|}\right)^2 \mathrm{d}s + \frac{\lambda}{4}\int_K \left(|\nabla\phi|^2 - 1\right)^2 \mathrm{d}\boldsymbol{x}, \tag{2.9}$$

where $\lambda \geq 0$ is the weight of the constraint $|\nabla\phi_h| = 1$.

We will use the same finite element spaces as in the discontinuous Galerkin method to be discussed subsequently. Following [100], we introduce two spaces of polynomials:

$$V_D^N = \{\phi : \phi \in P^N(K), \forall K \in \mathcal{T}_h\} \tag{2.10}$$

and

$$W_D^N = \left\{\mathbf{w} : \mathbf{w} = \nabla v, v \in V_D^N\right\}. \tag{2.11}$$

Then the problem can be formulated as: find $\nabla\phi_h \in W_D^N$ such that $E(\nabla\phi_h) = \min E(\nabla\phi)$ for all $\nabla\phi \in W_D^N$. The corresponding variational form is

$$\int_{\Gamma\cap K}\left(\nabla\phi_h - \frac{\nabla\phi_0}{|\nabla\phi_0|}\right)\cdot\mathbf{w}\mathrm{d}s + \lambda\int_K \left(|\nabla\phi_h|^2 - 1\right)\nabla\phi_h\cdot\mathbf{w}\mathrm{d}\boldsymbol{x} = 0, \forall\mathbf{w} \in W_D^N. \tag{2.12}$$

The first term in (2.12) requires a surface integral on $\Gamma \cap K$, which is not known explicitly. We replace it by a volume integral that is easy to compute numerically, and rewrite (2.12)

as

$$\int_K \left(\nabla\phi_h - \frac{\nabla\phi_0}{|\nabla\phi_0|}\right)\cdot \mathbf{w}\bar{\delta}_\xi(\phi_0)\mathrm{d}\boldsymbol{x} + \lambda\int_K \left(|\nabla\phi_h|^2-1\right)\nabla\phi_h\cdot\mathbf{w}\mathrm{d}\boldsymbol{x} = 0, \forall \mathbf{w}\in W_D^N, \quad (2.13)$$

where $\bar{\delta}_\xi$ is a shifted smooth delta function defined as

$$\bar{\delta}_\xi(\phi_0) = \begin{cases} \frac{1}{2\xi}\left(1+\cos\left(\frac{\pi}{\xi}\frac{\phi_0}{|\nabla\phi_0|}\right)\right)+\frac{\xi_1}{\xi}, & \text{if } \frac{|\phi_0|}{|\nabla\phi_0|}<\xi, \\ \frac{\xi_1}{\xi}, & \text{otherwise.}\end{cases} \quad (2.14)$$

Here ξ is the half width of the narrow band and ξ_1 is a small positive parameter to avoid singular matrices. We will discuss the choices of ξ and ξ_1 toward the end of this subsection. It should be noted that $\phi_0/|\nabla\phi_0|$ is used here, instead of ϕ_0, to take care of the extreme cases when $|\nabla\phi_0| \ll 1$ or $\gg 1$. This method essentially projects $\frac{\nabla\phi_0}{|\nabla\phi_0|}$ to a gradient space, as can be seen in (2.13), it is thus referred to as the weighted local projection (WLP) method hereinafter. The idea of local projection was first seen in [109], where the author projected $\frac{\phi_0}{|\nabla\phi_0|}$ to ϕ_h in the interface cells. This is however inaccurate unless $|\nabla\phi_0|$ is a constant.

Note that we do not need ϕ_0 to be a signed distance function here. When ϕ_0 is close to a signed distance function, for example, the contours of ϕ_0 are parallel lines or concentric circles, $\nabla\phi_h = \frac{\nabla\phi_0}{|\nabla\phi_0|}$ automatically satisfies the conditions (2.7) and (2.8), hence the penalty in (2.13) plays no role. For the more general case, $\frac{\nabla\phi_0}{|\nabla\phi_0|}$ does not even reproduce the gradient of any scalar function, because $\nabla\times\left(\frac{\nabla\phi_0}{|\nabla\phi_0|}\right)=0$ cannot be guaranteed (although $\nabla\times\nabla\phi_0=0$ is satisfied). Thus the penalty term in (2.13) is necessary to maintain the signed distance function in the cell K, which will be further discussed in Sec. 2.4.1.

One thing we need to pay attention to is that the quality of the WLP may deteriorate if the length of the interface in cell K is very small. For instance, when the interface only cuts a small portion of the cell at one corner, we observe that $\nabla\phi_h$ computed by (2.13) leads to

contours of ϕ_h being concentric circles. This may be incorrect. An easy way to avoid this problem is to anchor $\nabla\phi_h$ by the WLP in part of the interface cells only, and treat all the other cells as non-interface cells where $\nabla\phi_h$ is computed by solving (2.3).

We thus define another set $I_p \subset I$ that only contains cells with sufficient amount of interface:

$$I_p = \{K \in I : |K \cap \Gamma| \geq p\}, \tag{2.15}$$

where $|K \cap \Gamma|$ is the length (or area in 3D) of the interface segment in K, and $p < h$ ($p < h^2$ in 3D) is a positive number. If the interface Γ intersects K at two points, then $|K \cap \Gamma|$ can be approximated by the distance between these two intersections. While in 3D, we only consider the cases when the interface cuts a cell at three and four points. In both cases, the area is easy to compute with the coordinates of those intersections. As for other complicated cases in 2D with more than two intersections, and how to locate intersections will be discussed in Sec. 2.3. In practice, we choose $p = h/2$ (or $p = h^2/2$ in 3D). The WLP method (2.13) is applied to every cell $K \in I_p$ and the DG method (2.20) is used in all other cells. Thus the WLP solution in I_p can be considered as a boundary condition for the DG method, therefore the accuracy of the WLP affects the solution in the whole computational domain.

We employ Newton's method to solve the non-linear problem (2.13). Denoting the solution at the kth iteration by $\nabla\phi_h^k$, the solution at the $(k+1)$th iteration can be written as

$$\nabla\phi_h^{k+1} = \nabla\phi_h^k + \delta\nabla\phi_h, \tag{2.16}$$

where $\delta\nabla\phi_h$ is the increment to be determined. Substituting (2.16) into (2.13) for $\nabla\phi_h$ and

dropping the higher order terms of $\delta\nabla\phi_h$, we obtain the linear system for $\delta\nabla\phi_h$

$$\int_K \left[\left(\bar{\delta}_\epsilon(\phi_0) + \lambda\left(|\nabla\phi_h^k|^2 - 1\right)\right)\delta\nabla\phi_h \cdot \mathbf{w} + 2\lambda\left(\nabla\phi_h^k \cdot \delta\nabla\phi_h\right)\left(\nabla\phi_h^k \cdot \mathbf{w}\right)\right] \mathrm{d}\boldsymbol{x}$$
$$= -\int_K \left[\left(\nabla\phi_h^k - \frac{\nabla\phi_0}{|\nabla\phi_0|}\right)\bar{\delta}_\epsilon(\phi_0) + \lambda\left(|\nabla\phi_h^k|^2 - 1\right)\nabla\phi_h^k\right] \cdot \mathbf{w}\mathrm{d}\boldsymbol{x} \quad (2.17)$$

We solve (2.17) for $\delta\nabla\phi$ and update $\nabla\phi^k$ according to (2.16) repeatedly, until the residual, i.e., the right-hand side of (2.17), is smaller than a prescribed tolerance.

An initial guess plays a crucial role in Newton's method, otherwise it may not converge or converge to a wrong solution. An easy way to find a good initial guess is to solve

$$\int_K \left(\nabla\phi_h^0 - \frac{\nabla\phi_0}{|\nabla\phi_0|}\right) \cdot \mathbf{w}\mathrm{d}\boldsymbol{x} = 0, \forall\ \mathbf{w} \in W_D^N. \quad (2.18)$$

for $\nabla\phi_0$. Different from the Newton's method proposed by Chopp [66] to compute the closest point that may fail to converge, especially in three dimensions [93, 110], our method always converges with $\nabla\phi_0$ computed from (2.18). Typically, it takes less than five iterations for Newton's method to converge to a tolerance of 10^{-10}.

The WLP method can be summarized as follows. Firstly, identify the set I_p (2.15). Secondly, for each cell $K \in I_p$ solve (2.18) to get the initial guess $\nabla\phi_h^0$. Finally, solve (2.13) by Newton's method (2.17).

We now discuss the parameters in the WLP. If the number of quadrature points is less than the degrees of freedom in $\nabla\phi_h$, the matrix in (2.17) is singular. Thus distance of two quadrature points should not exceed the bandwidth 2ξ, which is the support of the smooth delta function (2.14), so that we have sufficient quadrature points to prevent the matrix in (2.17) from being singular. However, the bandwidth 2ξ cannot be too small, otherwise we will have too many quadrature points, which increases computational cost. In general, when

a signed distance function is advected in a flow field for a very short amount of time, ϕ_0 is close to a signed distance function. In this case, a large ξ, such as $\xi = 0.1h \sim h$ can produce desirable results. For extreme cases when ϕ_0 is highly distorted, we will need a smaller ξ, such as $\xi = 0.01h \sim 0.001h$, to ensure high-order accuracy. As for the quadrature rule, we adopt a composite Gauss-Lobatto quadrature rule in each interface cell $K \in I_p$ to take care of the interface that almost overlaps with a cell edge and maintain sufficient degree of precision as well. As an illustration, suppose cell K is divided into $Q \times Q$ subcells, and a $2N$-point Gauss-Lobatto quadrature is used in each subcell to accurately integrate (2.17). This leads to $(2NQ-Q+1)^2$ quadrature points in total. For example, in order to resolve $\epsilon = 0.01h$ with $N = 3$, we may need at least 10×10 subcells with 2601 quadrature points. Oftentimes in practice, this condition can be relaxed to save quadrature points without severely affecting the solution accuracy.

Another parameter ξ_1 in the WLP is introduced to further improve conditioning of the matrix. We usually choose $\xi_1 = 0.01/h$, since the numerical results are not sensitive to it. Extra attention should be paid to the choice of λ if $|\nabla\phi_0|$ is highly non-uniform or if a high order DG method is used. A too small λ may cause ϕ_h to deviate from a signed distance function while a too big λ may shift the interface. We typically choose $\lambda = 100$.

Note that there are other minimization-based methods for level-set reinitialization, such as the elliptic reinitialization [74, 75] that minimizes the following target functional

$$F(\phi) = \frac{1}{2}\int_\Omega (|\nabla\phi| - 1)^2 d\boldsymbol{x} + \frac{\alpha_{ER}}{2}\int_\Gamma \phi^2 ds, \tag{2.19}$$

where α_{ER} is a parameter similar to our $1/\lambda$. The WLP method is different from the minimization-based elliptic reinitialization (ER) methods. First, ER is a global method while the WLP is a local method that is easy to parallelize. Second, ER directly enforces

$\phi = 0$ while the WLP imposes $\nabla\phi = \frac{\nabla\phi_0}{|\nabla\phi_0|}$ on Γ. The surface integral of (2.19) is very sensitive to the location of the interface, because ϕ varies along the normal direction to the interface. Further, the surface integral may require sophisticated techniques such as moment-fitting [111]. On the contrary, $\nabla\phi_0$ is nearly constant along the interface normal unless ϕ_0 is heavily distorted. Thus the surface integral in the WLP (2.12) is not that sensitive to the interface location and we can safely replace it with a volume integral as in (2.13).

2.2.2 Non-interface cells

In this section, we employ the DG method [99, 100] to solve for $\nabla\phi_h$ in $\mathcal{T}_h/I_p$ and develop a new hybrid flux to handle highly distorted ϕ_0.

By multiplying (2.3) with the test function $\mathbf{w} \in W_D^N$, and performing integration by parts, we obtain the weak formulation

$$\left(\frac{\partial\nabla\phi_h}{\partial\tau}, \mathbf{w}\right)_K - \left(H\left(\nabla\phi_h\right), \nabla\cdot\mathbf{w}\right)_K + \left(\hat{H}(\nabla\phi_h^-, \nabla\phi_h^+), \mathbf{w}\right)_{\partial K} = 0, \ \forall \mathbf{w} \in W_D^N, \tag{2.20}$$

where $(.,.)_K$ and $(.,.)_{\partial K}$ denote the inner products in the cell K and on its boundary ∂K, respectively, e.g.,

$$\left(\nabla\phi_h, \mathbf{w}\right)_K = \int_K \nabla\phi_h \cdot \mathbf{w} \mathrm{d}\boldsymbol{x}, \tag{2.21}$$

$$\left(\hat{H}(\nabla\phi_h^-, \nabla\phi_h^+), \mathbf{w}\right)_{\partial K} = \sum_{e\in\partial K} \int_e \hat{H}(\nabla\phi_h^-, \nabla\phi_h^+) \cdot \mathbf{w} ds. \tag{2.22}$$

$\hat{H}(\nabla\phi_h^-, \nabla\phi_h^+)$ is the numerical flux approximating $H(\nabla\phi_h)\boldsymbol{n}_c$, where $\boldsymbol{n}_c$ is the unit outward pointing normal to the cell edge. $\nabla\phi_h^-$ is the trace from the interior of cell K, while $\nabla\phi_h^+$ from the interior of the neighboring cell. Details of $\hat{H}(\nabla\phi_h^-, \nabla\phi_h^+)$ are given later. Note that if $|\nabla\phi_0| \ll 1$ or $\gg 1$ on the interface, $S_\eta(\phi_0)$ should be replaced by $S_\eta(\frac{\phi_0}{|\nabla\phi_0|})$ to maintain the thickness of the transition layer, and the components of $\nabla\phi_h$ are not completely independent,

e.g., $\nabla \times \nabla \phi_h = 0$ is always satisfied.

To simplify the procedure of reconstructing ϕ_h, we choose the Legendre polynomial space as the finite dimensional space V_D^N. This is because the basis functions are L_2-orthogonal and normalized in the reference cell, the first basis function v_0 is constant and equal to 1 in the reference cell. Therefore, once $\nabla \phi_h$ is obtained, we only have one missing constant to reconstruct ϕ_h. We approximate ϕ by

$$\phi_h^K = \sum_{i=0}^{m} c_i v_i, \tag{2.23}$$

where $v_i \in V_D^N$, $i = 0, 1, \ldots, m$, are the basis polynomials, and $m+1$ is the number of degrees of freedom. We drop the superscript K from ϕ_h^K hereinafter for the ease of presentation. Then $\nabla \phi$ is approximated by

$$\nabla \phi_h = \sum_{i=0}^{m} c_i \nabla v_i. \tag{2.24}$$

Let $\mathbf{w}_i = \nabla v_i \in W_D^N$. Since v_0 is constant, we can further simplify (2.24) to

$$\nabla \phi_h = \sum_{i=1}^{m} c_i \mathbf{w}_i. \tag{2.25}$$

Substituting (2.24) into (2.20), we can obtain

$$\mathbf{A} \frac{d\mathbf{c}}{dt} = \mathbf{F}, \tag{2.26}$$

where $\mathbf{A} \in \mathbb{R}^{m \times m}$ and $\mathbf{A}_{i,j} = (\mathbf{w}_i, \mathbf{w}_j)_K$, $\mathbf{c} = [c_1, \ldots, c_m]^\intercal$, and $\mathbf{F} \in \mathbb{R}^m$ with

$$\mathbf{F}_i = \left(H\left(\nabla \phi_h\right), \nabla \cdot \mathbf{w}_{i+1}\right)_K - \left(\hat{H}\left(\nabla \phi_h^-, \nabla \phi_h^+\right), \mathbf{w}_{i+1}\right)_{\partial K}. \tag{2.27}$$

These m equations uniquely determine $\nabla \phi_h$ in K. Thus the calculation of $\nabla \phi_h$ is completely

decoupled from c_0. To recover ϕ_h, we just need c_0. Further, this additional degree of freedom can be used to preserve the interface, as discussed in Sec. 2.3.

The hybrid numerical flux

Now we focus on the numerical flux $H(\nabla\phi_h)\boldsymbol{n}_c$, whose Jacobian matrix is a rank-one matrix

$$J = S_\eta(\phi_0)\boldsymbol{n}_c\frac{\nabla\phi_h^T}{|\nabla\phi_h|}. \tag{2.28}$$

with only one non-zero eigenvalue

$$a = S_\eta(\phi_0)\boldsymbol{n}_c\cdot\frac{\nabla\phi_h}{|\nabla\phi_h|}. \tag{2.29}$$

Here $\boldsymbol{n}_c$ is the outward unit normal, as shown in Fig. 2.1. Note that the sign of the non-zero eigenvalue a indicates the wind direction.

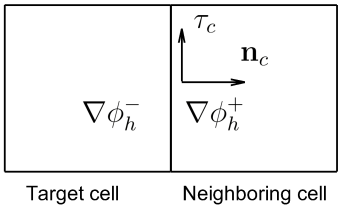


Figure 2.1: The edge between two cells.

We tested two popular numerical fluxes before we develop the hybrid numerical flux. The first one is the Roe flux

$$\hat{H}_{Roe}(\nabla\phi_h^-,\nabla\phi_h^+) = \begin{cases} H(\nabla\phi_h^-)\boldsymbol{n}_c, & \text{if } \frac{a^-+a^+}{2}\geq 0, \\ H(\nabla\phi_h^+)\boldsymbol{n}_c, & \text{otherwise.} \end{cases} \tag{2.30}$$

However it is entropy violating and unstable in our numerical tests. The second one is the Lax-Friedrichs (LF) flux following [99] with slight modifications:

$$\hat{H}_{LF}(\nabla\phi_h^-, \nabla\phi_h^+) = H(\frac{\nabla\phi_h^- + \nabla\phi_h^+}{2})\boldsymbol{n}_c - \frac{\alpha}{2}[\![\nabla\phi_h]\!], \tag{2.31}$$

where $\alpha = \max|a|$ is taken over the relevant range and $[\![\nabla\phi_h]\!] = \nabla\phi_h^+ - \nabla\phi_h^-$. We have tested the original local LF in [99], the local LF (maximum taken over the two cells sharing the same edge), and the global LF (maximum taken over the whole computational domain); they all produce satisfactory and almost identical solutions when ϕ_0 is close to a signed distance function. In the rest of this work, we simply choose the global LF flux, correspondingly $\alpha = 1$, for computational efficiency.

Nevertheless, the LF flux is deficient if ϕ_0 is far away from a signed distance function. For example, it takes extremely long time to generate continuous contours of ϕ in the test case of Sec. 2.4.1. After investigating the numerical results carefully, we observe that ϕ_h contours are likely to be broken where the cell edge is normal to the ϕ_h contours, i.e., when $\boldsymbol{n}_c \cdot \nabla\phi_h \approx 0$. Further looking into $H(\nabla\phi_h)\boldsymbol{n}_c$, we learn that the flux only affects the component of $\nabla\phi_h$ that is normal to the cell edge, denoted by $\nabla\phi_n \equiv (\nabla\phi_h \cdot \boldsymbol{n}_c)\boldsymbol{n}_c$, but has no effect on the tangential component $\nabla\phi_\tau \equiv \nabla\phi_h - (\nabla\phi_h \cdot \boldsymbol{n}_c)\boldsymbol{n}_c$.

Therefore, we add a penalty term to the LF flux to control $\nabla\phi_\tau$ and the resulting hybrid numerical flux is

$$\hat{H}(\nabla\phi_h^-, \nabla\phi_h^+) = H(\frac{\nabla\phi_h^- + \nabla\phi_h^+}{2})\boldsymbol{n}_c - \frac{\alpha}{2}[\![\nabla\phi_n]\!] - \frac{\beta}{2}[\![\nabla\phi_\tau]\!], \tag{2.32}$$

where β is the penalty parameter. If $\beta = \alpha$, (2.32) reduces to the LF flux (2.31). The value of β is adjusted based on the wind direction to deal with severely distorted ϕ_0. Let $a^\pm = S_\eta(\phi_0)\,\boldsymbol{n}_c \cdot \frac{\nabla\phi_h^\pm}{|\nabla\phi_h^\pm|}$, then we can determine β using Algorithm 1. We take $\beta_{\min} = 1$ and

$\beta_{\max} \geq \beta_{\min}$. That is, when there is an expansion wave or the target cell is downwind, we need a larger β. The rationale is that $\nabla\phi_\tau^\pm$ should be changed more in the downwind cell than in the upwind cell toward the final goal $\nabla\phi_\tau^- = \nabla\phi_\tau^+$.

Algorithm 1 Determination of β

```
if a^- ≤ 0 and a^+ ≥ 0 then
    β = β_max
else if (a^- + a^+)/2 > 0 then
    β = β_min
else
    β = β_max
end if
```

We remark that the hybrid numerical flux is monotone for piecewise constant $\nabla\phi_h$, but not conservative any more, because β takes different values at two sides of a cell edge. The modified CFL conditions for forward Euler in time are given by

$$\Delta t \leq \min_{\Omega} \left(\frac{h}{\alpha + \beta_{\max}} \right) \tag{2.33}$$

in two dimensions and

$$\Delta t \leq \min_{\Omega} \left(\frac{h}{\alpha + 2\beta_{\max}} \right) \tag{2.34}$$

in three dimensions.

Slope limiter for $\nabla\phi_h$

The combination of the WLP in Sec. 2.2.1 and the DG method in Sec. 2.2.2 can handle discontinuities in $\nabla\phi_h$ quite well in most cases. However, in some extreme case, such as sharp corners on a square interface as shown in Sec. 2.4.3, numerical oscillations may develop at the singular area where characteristics intersect. In this case we may need limiters to keep

the solution stable. Since the limiters are rarely used, we simply follow the ideas of Cockburn and Shu [112]. Other options of limiters can be found in [113, 114, 115].

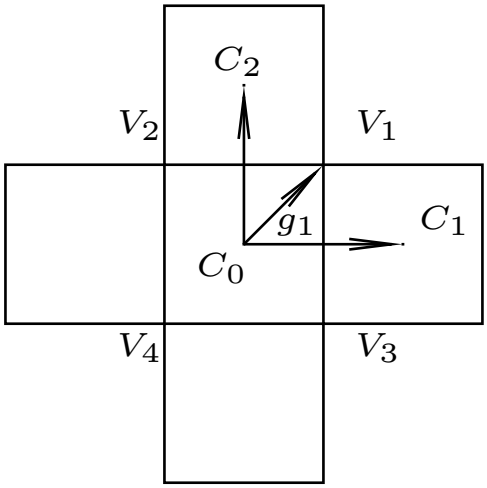


Figure 2.2: Limiters in a quadrilateral mesh.

We construct the slope limiter based on $\nabla\phi_h$ at vertices so that it is easy to extend to adaptive quadrilateral meshes. Consider a target cell with four neighbors in two-dimensional quadrilaterals as shown in Fig. 2.2. We denote the barycenter of the quadrilateral K_i by C_i, $i = 0, \ldots, 4$, and the vertex of cell K_0 by V_i, $i = 1, \ldots, 4$. Observe that

$$V_1 - C_0 = \alpha_1 (C_1 - C_0) + \alpha_2 (C_2 - C_0) \tag{2.35}$$

for some nonnegative coefficients α_1 and α_2 that can be evaluated from geometric positions of barycenters and vertices. Then we can compute the difference of $\nabla\phi_h$ between vertex V_1 and cell center C_0 based on the cell averages:

$$\mathbf{g}_1 = \alpha_1 \left(\nabla\bar{\phi}_{h,1} - \nabla\bar{\phi}_{h,0} \right) + \alpha_2 \left(\nabla\bar{\phi}_{h,2} - \nabla\bar{\phi}_{h,0} \right), \tag{2.36}$$

where

$$\nabla\bar{\phi}_{h,i} = \frac{1}{|K_i|}\int_{K_i} \nabla\phi_h \mathrm{d}\boldsymbol{x},\ i = 0, \ldots, 4. \tag{2.37}$$

Here $|K_i|$ denotes the area of cell K_i. Similarly, we can compute $\mathbf{g}_i,\ i = 2, 3, 4$.

Let

$$\Delta_i = \text{minmod}\left(\nabla\phi_h(V_i) - \nabla\bar{\phi}_{h,0}, \nu\mathbf{g}_i\right), \tag{2.38}$$

where the minmod function is defined as

$$\text{minmod}(a, b) = \begin{cases} s\min(|a|, |b|), & \text{if } s = \text{sign}(a) = \text{sign}(b), \\ 0, & \text{otherwise}, \end{cases} \tag{2.39}$$

and $\nu \in [1, 2]$ is a constant. If a and b are vectors, then the minmod function is applied component-wise. If $\Delta_i = \nabla\phi_h(V_i) - \nabla\bar{\phi}_{h,0}$ for all $i = 1, \ldots, 4$, limiting is not necessary in cell K_0. Otherwise, $\nabla\phi_h \in W_D^N$ $(N \geq 2)$ is limited to $\nabla\tilde{\phi}$ with

$$\nabla\tilde{\phi} = \sum_{i=1}^{5} \tilde{c}_i \mathbf{w}_i \in W_D^2, \tag{2.40}$$

where $\tilde{c}_i, i = 1, \cdots, 5$ are the unknowns.

The limiter seeks to compute $\nabla\tilde{\phi}$ that approximates the limited directional derivatives along the two diagonals of K_0 by solving the least squares problem:

$$\min_{\tilde{c}_3, \tilde{c}_4, \tilde{c}_5} \left\| \begin{bmatrix} (V_4 - V_1)\cdot\nabla\nabla\tilde{\phi} \\ (V_3 - V_2)\cdot\nabla\nabla\tilde{\phi} \end{bmatrix} - \begin{bmatrix} 2\text{minmod}\left(\Delta_4, -\Delta_1\right) \\ 2\text{minmod}\left(\Delta_3, -\Delta_2\right) \end{bmatrix} \right\|_2, \tag{2.41}$$

where $\nabla\nabla\tilde{\phi} = \begin{bmatrix} 2\tilde{c}_4 & \tilde{c}_3 \\ \tilde{c}_3 & 2\tilde{c}_5 \end{bmatrix}$. In addition, the conservation of $\nabla\phi$ requires

$$\int_{K_0} \nabla\tilde{\phi} \mathrm{d}\boldsymbol{x} = \int_{K_0} \nabla\phi_h \mathrm{d}\boldsymbol{x}. \tag{2.42}$$

For a 2D quadrilateral cell, (2.41) and (2.42) provides five constraints in total, which uniquely determines the five coefficients in $\nabla\tilde{\phi}$ (gradient of a quadratic polynomial). In 3D, $\nabla\tilde{\phi}$ has nine coefficients which can be determined from the six constraints in (2.41) and three in (2.42). Once $\nabla\tilde{\phi}$ is obtained, we replace $\nabla\phi_h$ with $\nabla\tilde{\phi}$ in K_0.

We remark that the limiter is only applied when $-S(\phi_0)\nabla^2\phi > \frac{1}{h}$ in K_0 where the characteristics are converging and the curvature of ϕ contours is very high.

2.3 Reconstruction of ϕ_h

So far we have computed $\nabla\phi_h$ in both interface and non-interface cells, we just need c_0 to recover the complete ϕ_h as shown in (2.23). Suggested by Hu and Shu [99], there are two ways to obtain c_0. The first one is

$$\int_K \frac{\partial\phi_h}{\partial\tau} + \nabla H(\phi_h) = 0, \quad \forall K \in \mathcal{T}_h. \tag{2.43}$$

The second one is to first use the first method to compute ϕ_0 in one or a few cells, then integrate $\nabla\phi_h$

$$\phi_h(\boldsymbol{x},\tau) = \phi_h(\boldsymbol{x}_0,\tau) + \int_{\boldsymbol{x}_0}^{\boldsymbol{x}} \nabla\phi_h \cdot d\boldsymbol{s} \tag{2.44}$$

along some path from $\boldsymbol{x}_0$ to $\boldsymbol{x}$. Similar to most other PDE-based reinitialization methods, the first method inevitably leads to the shift of the interface. We thus adopt the second

method but determine $\phi_h(\boldsymbol{x}_0, \tau)$ in an interface-preserving way.

2.3.1 Interface cells

We identify the interface cells by checking the ϕ_0 values at the vertices: a cell is a non-interface cell if ϕ_0 at its four vertices (eight vertices for hexagonal cells in 3D) are all positive or all negative, otherwise it is an interface cell. In this way, we cannot identify cases such as Fig. 2.3 (d) and (e). However, Fig. 2.3 (d) can be taken care of by the neighboring cells, and the curvature of the interface in Fig. 2.3 (e) is too high ($> \frac{1}{h}$) for the DG polynomial space to recover the interface accurately. Usually, the only way to deal with high-curvature interface is mesh refinement, which is beyond the scope of this work.

An interface may cut a cell in many different ways, see Fig. 2.3 for illustrations. For simplicity, we only discuss the cases with two detected intersections, denoted by $\hat{A}$ and $\hat{B}$ in Fig. 2.3. We consider the problem in the reference cell $[0,1] \times [0,1]$. When a point (x, y) in the physical cell is mapped into the reference cell, the target is $(\hat{x}, \hat{y})$ and $\phi_0(x, y) = \hat{\phi}_0(\hat{x}, \hat{y})$, where we have used $\hat{}$ to denote the quantities in the reference frame. The exact locations of $\hat{A}$ and $\hat{B}$ can be obtained from $\hat{\phi}_0(\hat{A}) = \hat{\phi}_0(\hat{B}) = 0$. For example, the location of $\hat{B}$ in Fig. 2.3 (a) can be obtained by solving $\hat{\phi}_0(\hat{x}_B, 1) = 0$ using the secant method with starting values 0 and 1 (two ends of the edge). If the secant method fails, we switch to the root-bracketing false position method. As long as $\hat{B}$ is known, we can easily compute the coordinate of B through the mapping. Two intersections will give two constraints with only one unknown variable c_0, therefore we just need to solve an easy least squares problem

$$\min_{c_0} \left\| \begin{bmatrix} \phi_h(A) - \phi_0(A) \\ \phi_h(B) - \phi_0(B) \end{bmatrix} \right\|_2 = 0. \tag{2.45}$$

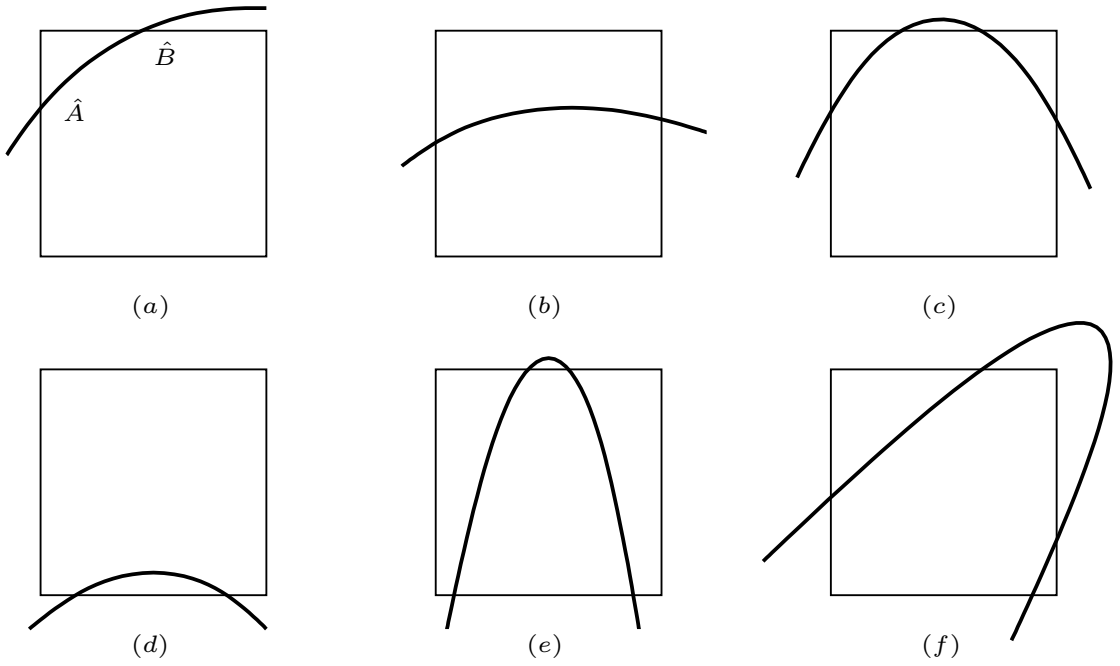


Figure 2.3: Some typical configurations of interface cells in the reference frame. (a) and (b) are the regular interface cells that can be easily identified. Only two intersections of (c) can be detected and it is treated in the same way as (b). (d) and (e) are not numerically detected as in interface cells. (f) is detected but the interface curvature is too high. Numerically, (d), (e), and (f) are not included in set I. Other cases, including the extremely rare cases of interface passing through one or more vertices, are not included here but considered in our code.

This problem has a simple solution

$$c_0 = -\frac{\phi_h(A)|_{c_0=0} + \phi_h(B)|_{c_0=0}}{2}, \tag{2.46}$$

where $\phi_h(\cdot)|_{c_0=0}$ denotes $\phi_h(\cdot)$ evaluated with $c_0 = 0$. To preserve the interface, we perform this procedure on every cell $K \in I$. This method can be easily extended to 3D where we can still use intersections between the interface and the edges (instead of faces of the 3D cell) to determine c_0 .

2.3.2 Non-interface cells

When ϕ_h is recovered in interface cells, the second method (2.44) can be used to compute c_0 in all the non-interface cells. However, (2.44) also implies that $\phi_h(\boldsymbol{x})$ is continuous across the cells, which leads to a more efficient way to reconstruct ϕ_h in all the other cells.

Suppose we have two adjacent cells K_A and K_B and c_0 in K_A, denoted by $c_{0,A}$, is already obtained. To compute $c_{0,B}$ in K_B, we just need one constraint by forcing ϕ_h to be continuous across the shared edge at any point. Here, we choose the center E of the shared edge and $c_{0,B}$ can be determined from

$$c_{0,B} = \phi_{h,A}(E) - \phi_{h,B}(E)|_{c_{0,B}=0}, \tag{2.47}$$

which is equivalent to (2.44) with any integration path from K_A to K_B crossing the shared edge E. This procedure can be carried out based on mesh connectivity: we first update the direct neighbors of interface cells, then update the neighbors of neighbors, and so on. Until c_0 is propagated from the interface cells to all non-interface cells.

Theoretically, the line integral in (2.44) is independent of integration paths. However, this is not the case in numerical simulations, especially when the path goes through singularities formed by intersecting characteristics (e.g., the center of a circular interface). Therefore we need to make sure that the integration path is following the characteristics. For example, if the characteristic direction $S_\eta(\phi_0)\nabla\phi$ points from K_A to K_B, the upwind cell is K_A and the downwind cell is K_B, then we use K_A to update $c_{0,B}$ in cell K_B. In light of the Fast Marching method [65] and the Fast Sweeping method [67], we develop Algorithm 2 to guarantee that all information is propagated away from the zero level set along the characteristics. In this algorithm we have used the fact that c_0 is a good approximately to the cell average of ϕ_h (exact average in the reference cell), and thus the solution propagates in the increasing order

of c_0. The additional condition on $S(\phi_0)\nabla\phi_h \cdot \boldsymbol{n}_c$ is just to double confirm that the neighbor is the upwind cell. If the target cell has multiple upwind neighbors, then we take the c_0 with the smallest magnitude. If the target cell does not have any upwind neighbors, which rarely happens in real calculations, we simple keep the c_0 determined from mesh connectivity.

Algorithm 2 Upwind update of c_0 in non-interface cells.

```
1 Compute an approximate c_0 in all non-interface cells based on mesh connectivity.
2 Sort all non-interface cells in an increasing order of |c_0|.
  for all sorted non-interface cells do
    Set c = Inf.
    for all edges of the target cell do
      Identify the neighboring cell sharing the same edge.
      if S(φ_0)∇φ_h · n < 0 AND |c_0^target| > |c_0^neighbor| then
        Compute c_0^new in the target cell based on the current neighbor
        if |c_0^new| > |c_0^neighbor| then
          Update c = min(c, |c_0^new|).
        end if
      end if
    end for
    if c ≠ Inf then
      Set c_0^target = sign(c_0^target) c in the target cell.
    end if
  end for
```

2.4 Numerical examples

In this section, we show the accuracy, long-term stability, and interface preserving properties of our algorithm. Unless otherwise stated, we conduct numerical tests in a unit square with 64×64 uniform cells using the following parameters: $N = 3$, $\beta_{\max} = 3$, $\lambda = 100$, $\xi = c_\xi h$ with $c_\xi = 0.1$, $\xi_1 = c_{\xi_1} c_\xi$ with $c_{\xi_1} = 0.01$, and $Q = 5$.

Discontinuous Galerkin methods are usually combined with the total variation diminishing (TVD) Runge-Kutta (RK) methods [116] to achieve high orders in both space and time. For

convection dominated problems, when a DG method with polynomial degree N is paired with an $(N+1)$th-order TVD RK method, stability requires $CFL \leq \frac{1}{2N+1}$ [117]. For other RK and DG combinations, stability condition has to be established based on numerical tests. For example, the third-order TVD RK is stable for different DG polynomial degrees with the following maximum CFL numbers: 0.130 for $N = 3$, 0.089 for $N = 4$, and 0.066 for $N = 5$. More details can be found in [117] and references therein. In our simulations, we adopt the third-oder TVD RK for time integration and choose $\Delta t = 0.1h$ for $N = 3$, such that $CFL = 0.1$.

The numerical algorithm is summarized as follows:

(i) Prepare initial condition. If ϕ_0 is not given in the DG solution space, project it to $\phi_{h,0} \in V_D^N$ and start the simulation with $\phi_{h,0}$.

(ii) Use the WLP method to determine $\nabla\phi_h$ in I_p (Sec. 2.2.1).

(iii) Solve the HJ equation using DG for $\nabla\phi_h$ in $\mathcal{T}_h/I_p$ until convergence (Sec. 2.2.2). Apply the second-derivative limiter if necessary (Sec. 2.2.2).

(iv) Determine c_0 in I and propagate it to all cells (Sec. 2.3). Now we get the complete solution ϕ_h in the computational domain.

The code is developed based on the deal.II finite element library [118, 119].

2.4.1 Convergence tests

We choose a simple interface shape, circular interface, to test the accuracy of our method in a square domain $\Omega = [-2, 2]^2$. The initial condition is given by

$$\phi_0 = g(\boldsymbol{x}) \left(\sqrt{x^2 + y^2} - r \right), \tag{2.48}$$

where $r = 0.9$. The signed distance function for this interface (exact solution of (2.1)) is $\phi^* = \sqrt{x^2 + y^2} - r$. Two different initial conditions are tested. The first one is ϕ_0 with $g(\boldsymbol{x}) = g_1(\boldsymbol{x}) = 0.8$, which has a uniform $|\nabla\phi_0|$, and $\frac{\nabla\phi_0}{|\nabla\phi_0|}$ directly yields the exact solution $\nabla\phi^*$. In the second test, the initial condition is distorted by $g(\boldsymbol{x}) = g_2(\boldsymbol{x}) = 0.1 + (x - r)^2 + (y - r)^2$ such that $|\nabla\phi_0|$ is highly non-uniform.

We define the following errors to investigate the accuracy of the WLP method. In order to measure the displacement of the interface from its initial position, we introduce

$$E_I = \frac{1}{L} \int_{K \in I} |H_\epsilon(\phi_h) - H_\epsilon(\phi^*)| \, \mathrm{d}\boldsymbol{x}, \tag{2.49}$$

where L is the length of the interface, ϕ^* is the exact solution, and H_ϵ is a smooth Heaviside function defined as

$$H_\epsilon(\phi) = \begin{cases} 0, & \text{if } \phi < -\epsilon, \\ 1, & \text{if } \phi > \epsilon, \\ \frac{1}{2}\left(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\epsilon}\right)\right), & \text{otherwise.} \end{cases} \tag{2.50}$$

We also define the averaged L_2 error

$$E_2 = \sqrt{\frac{1}{n_p} \sum_{K \in I_p} \frac{1}{|K|} \int_K (\phi_h - \phi^*)^2 \, \mathrm{d}\boldsymbol{x}} \tag{2.51}$$

and the L_∞ error

$$E_\infty = \max_{K \in I_p} |\phi_h - \phi^*| \tag{2.52}$$

to measure the error in the interface cells. Here n_p is the number of cells in I_p and $|K|$ is the area of cell K.

To investigate the accuracy in the whole computational domain, we define the L_2 error

$$e_2 = \sqrt{\sum_{K\in\Omega_0} (\phi_h - \phi^*)^2}, \tag{2.53}$$

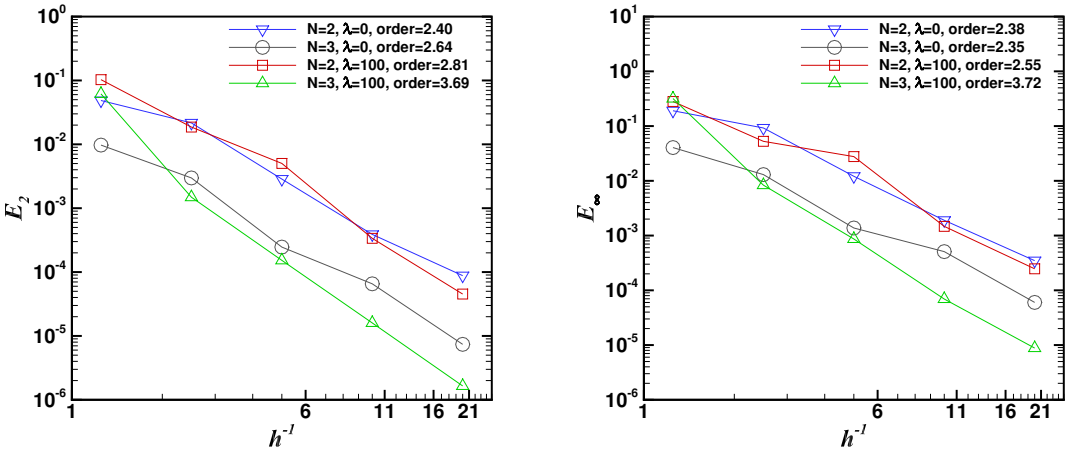
and the L_∞ error

$$e_\infty = \max_{K\in\Omega_0} |\phi_h - \phi^*| . \tag{2.54}$$

Here we rule out the central area near the center, which is a singular point, and take $\Omega_0 = [-2,2]^2 \setminus [-0.4,0.4]^2$. We run the tests on five uniform grids and the grid sizes are $h = \frac{0.8}{2^l}$, where $l = 0, 1, \cdots, 4$ is the level of refinement. We compare $\nabla\phi_h$ every 200 pseudo time steps to check if the steady state is achieved or not. If the L_2-norm of the difference in Ω_0 less than below 10^{-15}, then it is considered as steady state and we stop the computation; otherwise, we carry on the test up to pseudo time $\tau = 15$.

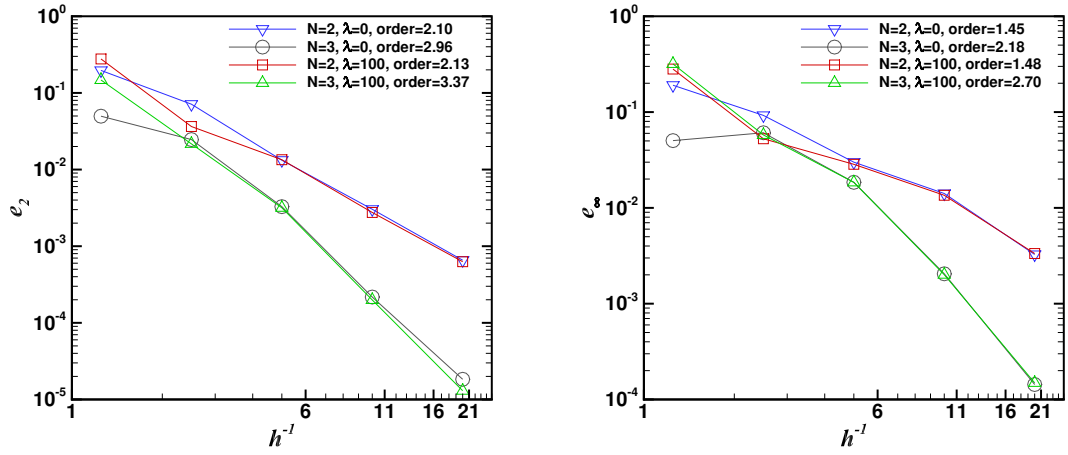
Convergence tests for $g(\boldsymbol{x}) = g_1(\boldsymbol{x})$

We first test ϕ_0 that is close to the signed distance function by scaling ϕ^* with a constant 0.8, therefore $\frac{\nabla\phi_0}{|\nabla\phi_0|}$ is uniform along the characteristics and $\left|\frac{\nabla\phi_0}{|\nabla\phi_0|}\right| = 1$. For this less challenging initial condition, we can use relatively large ξ in the WLP by setting $c_\xi = 0.1$, $Q = 5$, and test $\lambda = 0$ and 100. Due to the symmetry in $\nabla\phi_h$, we only show the errors of $(\phi_h)_x$ (derivative with respect to x) in Fig. 2.4. The orders in the legends, in this figure and all the following figures, are computed based on a power fitting without considering the data points from the coarsest mesh. When the polynomial degree N (roughly degree $N-1$ for $\nabla\phi_h$) is used, the WLP can achieve at least order N for $\nabla\phi_h$, except for the case with $N = 3$ and $\lambda = 0$. Theoretically, the value of λ in (2.13) should not play a role when $\left|\frac{\nabla\phi_0}{|\nabla\phi_0|}\right| = 1$. However, after projecting ϕ_0 to $\phi_{h,0} \in V_D^N$, $\left|\frac{\nabla\phi_0}{|\nabla\phi_0|}\right| = 1$ is no longer satisfied. This explains why $\lambda = 100$ produces better results than $\lambda = 0$ does when $N = 3$.



(a) E_2 of $(\phi_h)_x$ (b) E_∞ of $(\phi_h)_x$

Figure 2.4: Convergence of the WLP method for the circular interface with $g = g_1$. The errors are evaluated in the interface cells I_p. $c_\xi = 0.1$ and $Q = 5$.



(a) e_2 of $(\phi_h)_x$ (b) e_∞ of $(\phi_h)_x$

Figure 2.5: Convergence of the DG method for the circular interface with $g = g_1$. The errors are evaluated in $\Omega_0 = [-2, 2]^2/[-0.4, 0.4]^2$.

The solution from the WLP is then fed to the Hamilton-Jacobi equation as "boundary conditions". For both $N = 2$ and 3, the steady state is achieved within $\tau = 6$. The errors of $\nabla\phi_h$ away from the central singularity are presented in Fig. 2.5. The convergence orders of

$\nabla\phi_h$ are essentially of order N in L_2-norm as expected, though they are slightly lower than those in the WLP. While the convergence orders decrease to around $N - \frac{1}{2}$ in L_∞-norm, which is properly caused by poor quality of the solution in some individual cells. Note that e_∞ of $(\phi_h)_x$ even increases going from the first to the second data points in the case when $N = 3$ and $\lambda = 0$ in Fig. 2.5(b). This is because in the coarsest 5×5 mesh, most cells in Ω_0 are interface cells, which are computed by the WLP instead of the DG method. This kind of abnormal behavior has an influence on computing the convergence orders, which is also the reason why we skip the first data point.

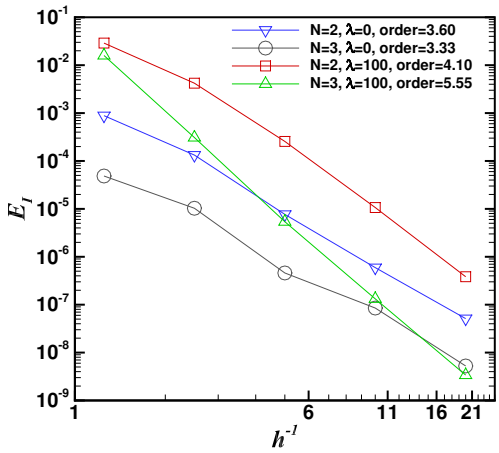
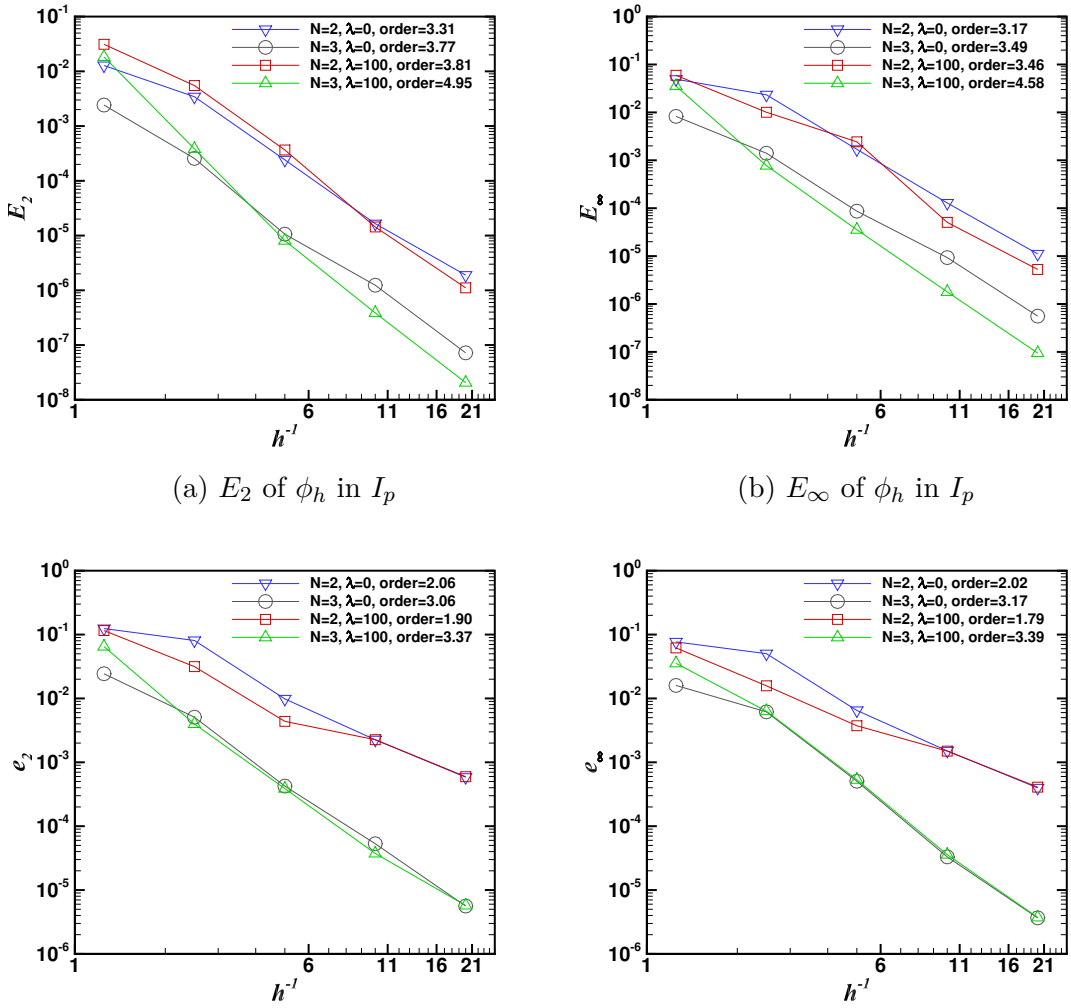


Figure 2.6: Interface displacement, measured by E_I in the interface cells I, for the circular interface with $g = g_1$.

We reconstruct ϕ_h by computing c_0 in each cell as discussed in Sec. 2.3. The displacement of the interface evaluated by E_I are shown in Fig. 2.6. For all λ and N we tested, the convergence orders of E_I are above N, which is sufficient for the overall Nth order convergence for ϕ_h to be discussed later. In particular, large λ is likely to shift the interface slightly, consequently in most cases E_I with $\lambda = 100$ is greater than that with $\lambda = 0$. However, large λ seems to provide higher order of convergence, which is at least at least $N+2$ with $\lambda = 100$ versus N with $\lambda = 0$.



(a) E_2 of ϕ_h in I_p

(b) E_∞ of ϕ_h in I_p

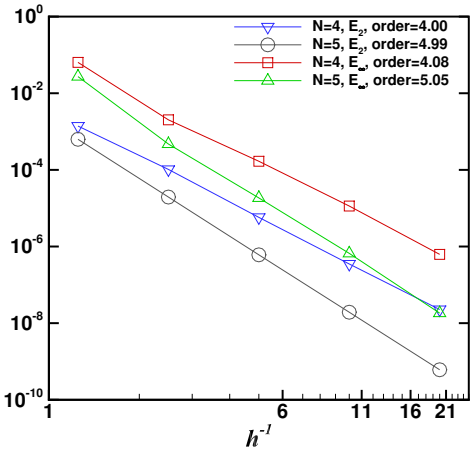
(c) e_2 of ϕ_h in Ω_0

(d) e_∞ of ϕ_h in Ω_0

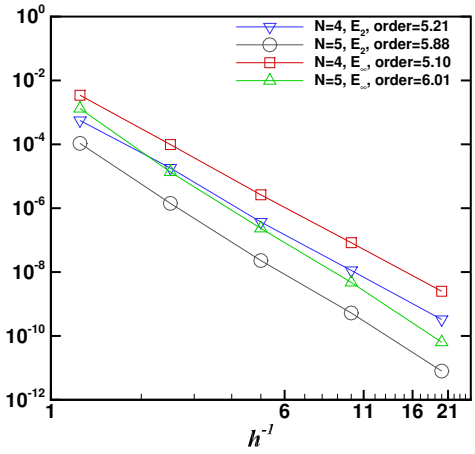
Figure 2.7: Errors of ϕ_h for the circular interface with $g = g_1$.

As for ϕ_h, we observe in Fig. 2.7 that the convergence order is $N+1$ in I_p, but reduces to N in Ω_0. This is expected, because of the approach we use to compute c_0 in Ω_0 is equivalent to the line integral of $\nabla\phi_h$. Consequently, in I_p where the integral path is $O(h)$, the convergence order is $N+1$. While in Ω_0, the integral path is $O(1)$, leading to the decrease of the convergence order to N.

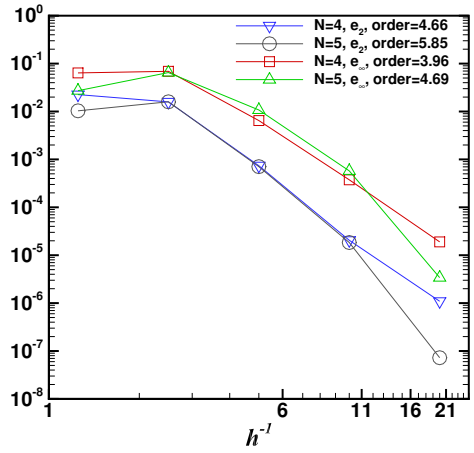
For higher polynomial degree N, our method can still maintain the optimal convergence



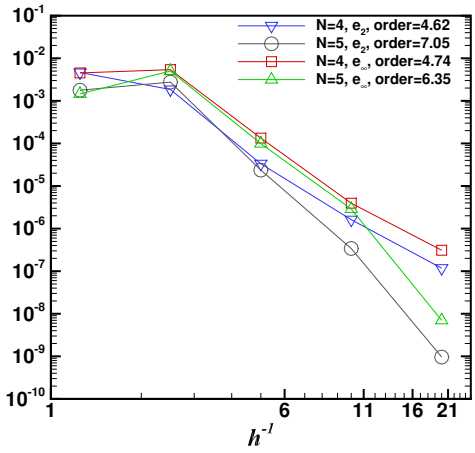
(a) Errors of $(\phi_h)_x$ in I_p



(b) Errors of ϕ_h in I_p



(c) Errors of $(\phi_h)_x$ in Ω_0



(d) Errors of ϕ_h in Ω_0

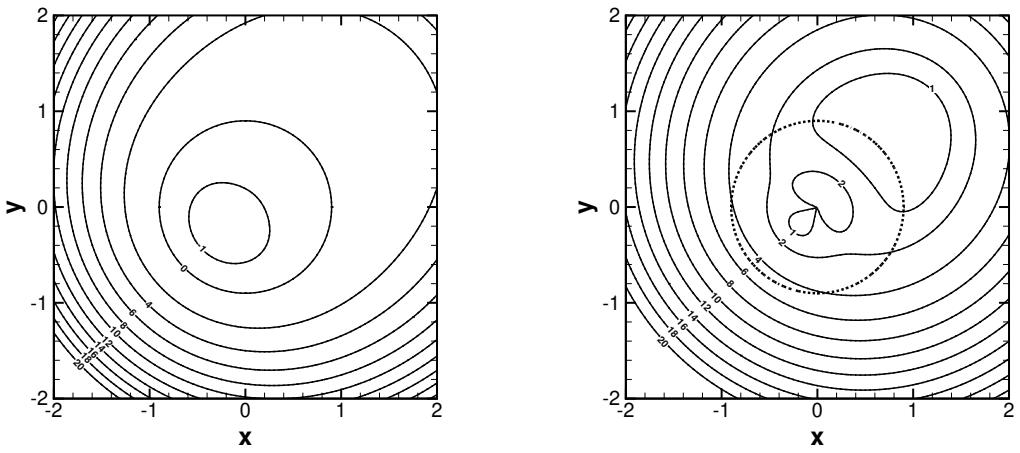
Figure 2.8: Convergence tests of higher order methods for circular interface with $g = g_1$. $\lambda = 0$, $Q = 5$, $c_\xi = 0.1$.

orders: order N for $\nabla\phi_h$, order $N+1$ for ϕ_h in interface cells, and (at least) order N for ϕ_h in the whole domain. The results for $N = 4$ and 5 are presented in Fig. 2.8. To reduce the sources of error, the exact initial condition (2.55) is directly used without projecting onto V_D^N in the WLP of these tests.

It should be noted that the closest point algorithm by Saye [93] can also achieve arbitrarily

high order. A key component of this algorithm is the least squares polynomial approximation of ϕ_0 around the interface, which always leads to very wide stencils in the finite difference framework. In principle, this algorithm can be adapted to DG and become another option for high order accuracy on unstructured meshes.

Convergence tests for $g(\boldsymbol{x}) = g_2(\boldsymbol{x})$

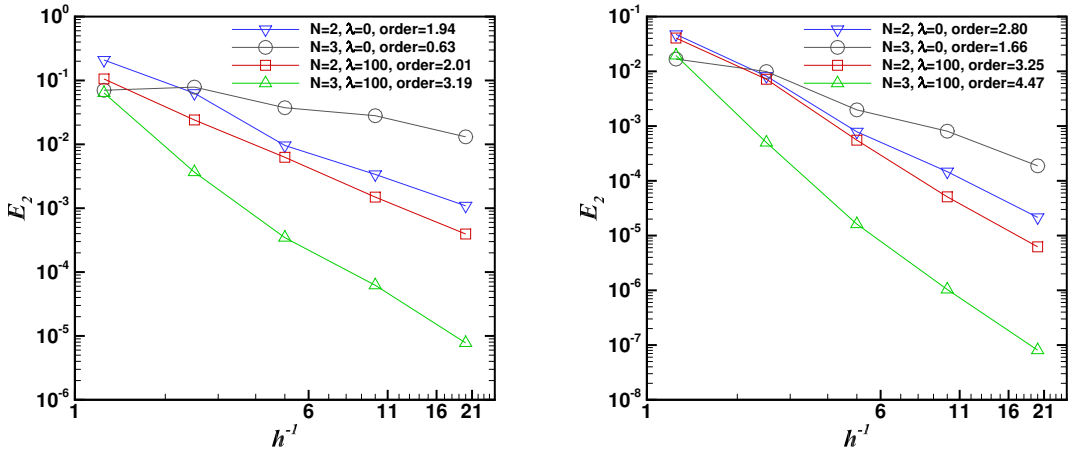


(a) ϕ_0 (b) $|\nabla\phi_0|$

Figure 2.9: The initial condition for circular interface with $g(\boldsymbol{x}) = g_2(\boldsymbol{x})$.

In the second convergence test, we use a more challenging ϕ_0. Fig. 2.9 displays the contours of both ϕ_0 and $\nabla\phi_0$, where we can see contour levels of ϕ_0 are highly non-uniform, and $|\nabla\phi_0|$ at the interface ranges from 0.24 to 4.82. To deal with such distorted ϕ_0, we have to carefully choose the parameters, such as the penalty parameter λ in the WLP, the bandwidth parameter $c_\xi = \xi/h$ of the smooth delta function, and β_{max} in the numerical flux.

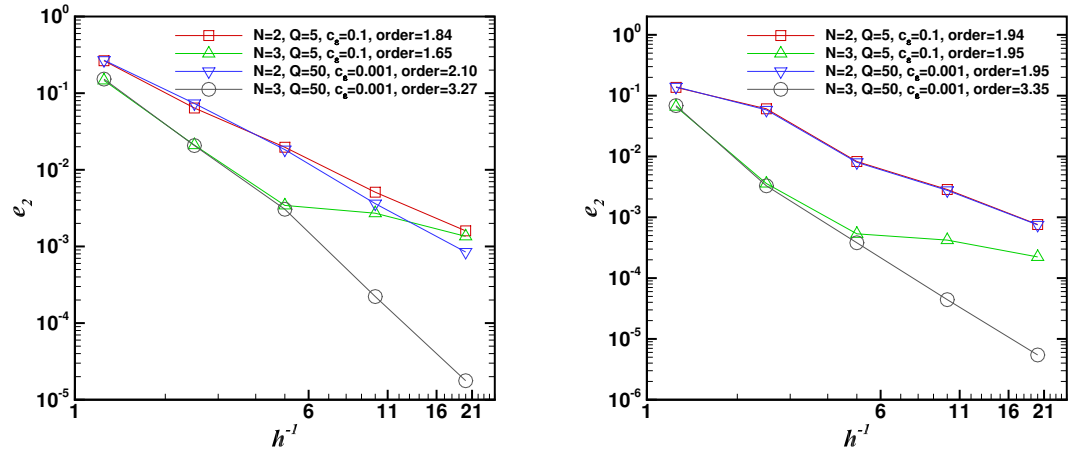
To investigate the impact of λ in the WLP, we test $\lambda = 0$ and 100 for $N = 2$ and 3. In addition, we use 50×50 subcells with a small bandwidth $c_\xi = 0.001$ in each interface cell to ensure the accuracy of the surface integral. Fig. 2.10 displays the convergence orders of all the test cases, where we can see that optimal convergence orders are attained for both



(a) E_2 of $(\phi_h)_x$ (b) E_2 of ϕ_h

Figure 2.10: Effect of λ for circular interface with $g = g_2$. The errors are evaluated in the interface cells I_p. $c_\xi = 0.001$, $Q = 50$, and $\beta_{max} = 3$.

$N = 2$ and 3 with $\lambda = 100$. As for $\lambda = 0$, higher orders of polynomials seems to deteriorate the solution, indicating that the penalty term in (2.13) plays a crucial role in obtaining a signed distance function, especially when high degree polynomials are used.



(a) e_2 of $(\phi_h)_x$ (b) e_2 of ϕ_h

Figure 2.11: Effect of ξ for the circular interface with $g = g_2$. The errors are evaluated in $\Omega_0 = [-2, 2]^2 \setminus [-0.4, 0.4]^2$. $\lambda = 100$, $\beta_{\max} = 3$.

We then test the effect of the bandwidth ξ in approximating the surface integral by a volume integral using the smooth delta function. Errors computed in Ω_0 with $c_\xi = 0.1$ and 0.001 are given in Fig. 2.11. As expected, the smaller bandwidth $c_\xi = 0.001$ produces better results than the larger one $c_\xi = 0.1$ does, especially for $N = 3$. While for $N = 2$, c_ξ does not seem to make any difference, because $\nabla\phi_h$ is almost linear, so is $\frac{\nabla\phi_{h,0}}{|\nabla\phi_{h,0}|}$ after projection onto V_D^N, therefore the integration of $\nabla\phi_h$ in the narrow band is only dependent on the values at the center of the narrow band, i.e., the interface. But this is not the case when $N = 3$, as a result, $c_\xi = 0.1$ causes much larger errors.

Last but not least, influence of $\beta_{\max}$ in the numerical flux is presented in Fig. 2.12, where we compare the convergence results for $\beta_{\max} = 1$, 2, and 3. Solutions with $\beta = 2$ and 3 both converge as expected, and the former reaches the steady state at $\tau = 14.55$ and the latter at $\tau = 7.24$. As for $\beta = 1$, the solution does not converge with mesh refinement at $\tau = 15$. This is because ϕ_0 is far from a signed distance function, after we apply the WLP in I_p, a big jump is generated between cells of I_p and their neighbors. Fig. 2.12 reveals that larger $\beta_{\max}$ is able to dissipate away the oscillations more efficiently. That being said, given sufficient pseudo time, $\beta_{\max} = 1$ may be able to create a desirable convergence order. Contours of $(\phi_h)_x$ in Fig. 2.13 provides a clear comparison among these three cases. The exact solution is reproduced almost perfectly with $\beta_{\max} = 3$. As $\beta_{\max}$ decreases, more oscillations are generated. A sequence of snapshots for the contour of ϕ_h with $\beta_{\max} = 2$ and 3 are given in Fig. 2.14, with $\beta_{\max} = 3$, the signed distance function is recovered in the whole domain at an instant between $\tau = 1$ and 2. Theoretically, it should take $\tau = 1$ to fully reproduce the exact solution. However, because of the smooth sign function (2.2), the numerical speed of propagation is actually smaller than one.

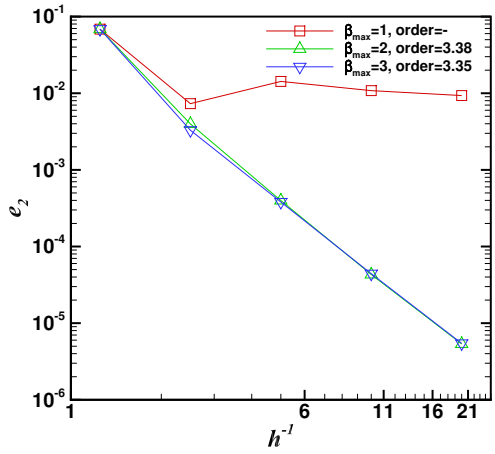
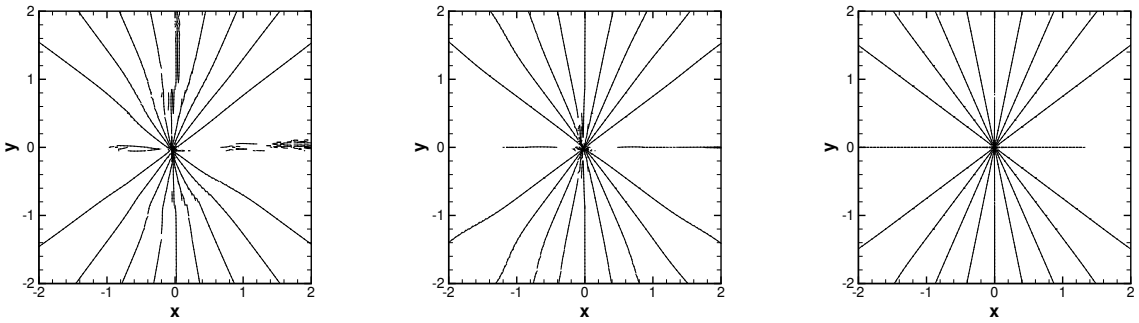


Figure 2.12: Effect of $\beta_{\max}$ for the circular interface with $g = g_2$. The errors are e_2 of ϕ_h evaluated in $\Omega_0 = [-2, 2]^2/[-0.4, 0.4]^2$. $N = 3$, $\lambda = 100$, $c_\xi = 0.001$, and $Q = 50$.

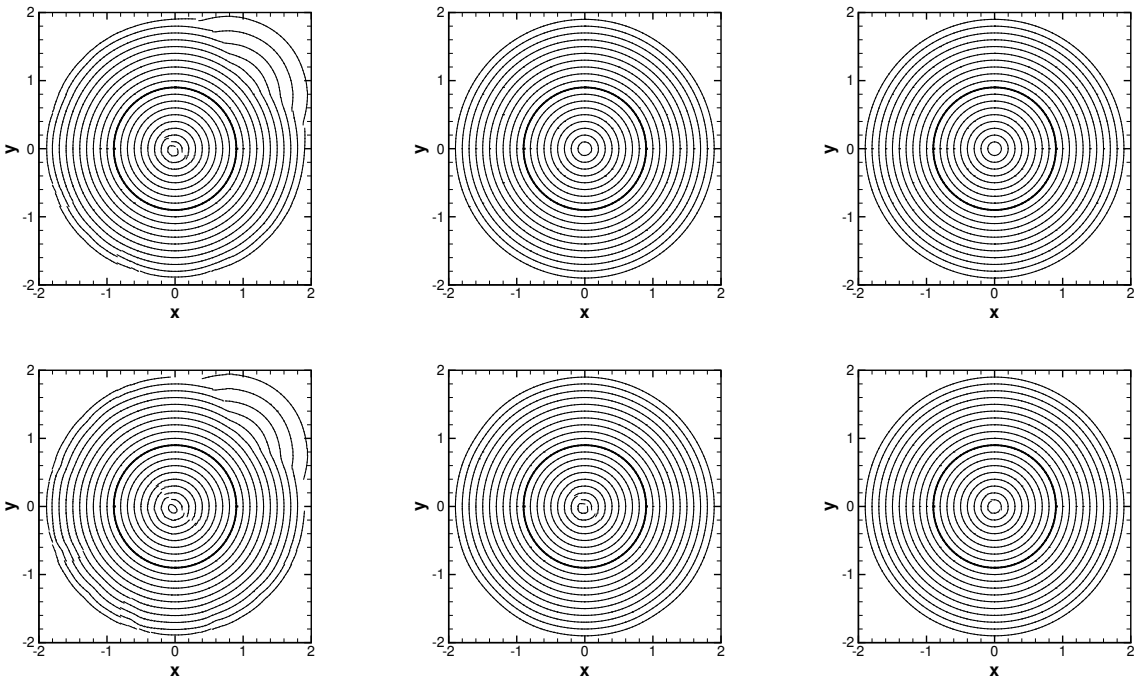


(a) $\beta_{\max} = 1$ (b) $\beta_{\max} = 2$ (c) $\beta_{\max} = 3$

Figure 2.13: $(\phi_h)_x$ at $\tau = 3$ for $g = g_2$ using different $\beta_{\max}$. Contours run from -1 to 1 with interval 0.2. $N = 3$, $\lambda = 100$, $c_\xi = 0.001$, $Q = 50$, and $h = 0.05$.

Reinitialization in 3D

Taking advantage of the template programming in deal.II, our code is dimension independent. In addition, the same parameters as discussed in previous sections can be applied to 3D simulations. Therefore, extension of our method to 3D is almost trivial when contact lines are absent. The only exception is the pseudo time step, which could be smaller because of the stability condition (2.34) in 3D.



(a) $\tau = 1$ (b) $\tau = 2$ (c) $\tau = 3$

Figure 2.14: Evolution of ϕ_h for $g = g_2$ using different $\beta_{\max}$. $\beta_{\max} = 3$ for the top row and $\beta_{\max} = 2$ for the bottom row. Contours run from -0.8 (center) to 1 with interval 0.1. The thick line denotes the interface. $N = 3$, $\lambda = 100$, $c_\xi = 0.001$, $Q = 50$, and $h = 0.05$.

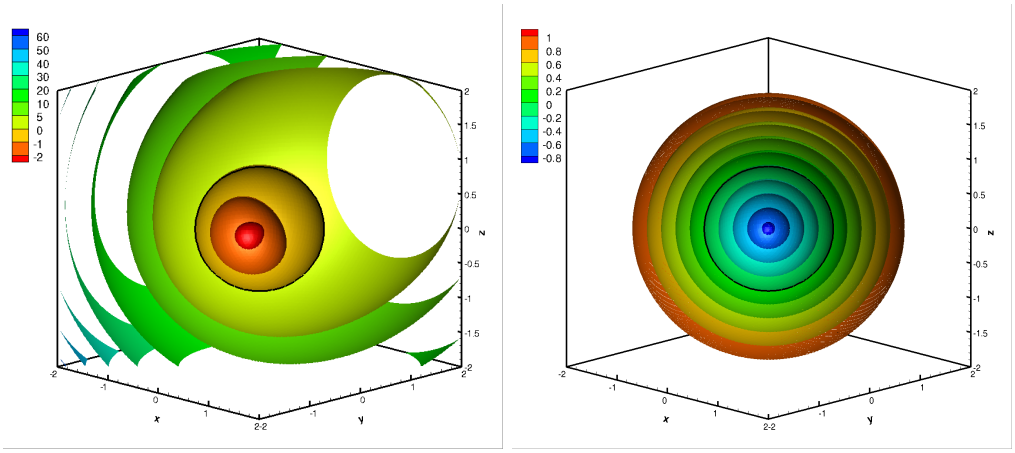
Here we adopt a 3D version of ϕ_0 in Sec. 2.4.1 by setting

$$\phi_0 = g_3(\boldsymbol{x}) \left(\sqrt{x^2 + y^2 + z^2} - r \right), \tag{2.55}$$

where

$$g_3(\boldsymbol{x}) = 0.1 + (x - r)^2 + (y - r)^2 + (z - r)^2 .$$

Figure 2.15 shows the 3D results using the same parameters as in 2D computations. Note that, according to our numerical tests, the pseudo time step for $\beta_{\max} = 3$ is $\Delta\tau = 0.05h$, while for $\beta_{\max} = 2$ is $\Delta\tau = 0.1h$.



(a) $\tau = 0$ (b) $\tau = 1.5$

Figure 2.15: Reinitialization in 3D. $N = 3$, $\beta_{\max} = 3$, $\lambda = 100$, $c_\xi = 0.1$, $c_\xi = 0.01$, and $Q = 5$. The mesh size is $h = 1/16$, which corresponds to 64^3 cells. The thick line denotes the interface.

2.4.2 Elliptic interface

In this example, we reinitialize the elliptic interface to $\tau = 15$ to examine the effect of the WLP in stabilizing the solution. Consider the initial condition

$$\phi_0(x, y) = (x - 0.5)^2 + 6\,(y - 0.5 - 0.5h)^2 - 0.1, \tag{2.56}$$

in a unit cell, where the major axis is placed at the cell centers on purpose such that discontinuities of $\nabla\phi$ lie inside computational cells, rather than on cell edges. The reason is that discontinuities within a cell is more challenging to handle than those on cell edges.

Comparisons of two solutions are given in Fig. 2.17. The only difference is that the WLP method is not applied on the top row, the DG method is used instead to solve for $\nabla\phi_h$ in I_p. At early stages of the simulation, we cannot see the large difference between two solutions. Both of them produce the signed distance function and capture the discontinuities inside computational cells nicely. However, as we continue the calculation, oscillations are getting

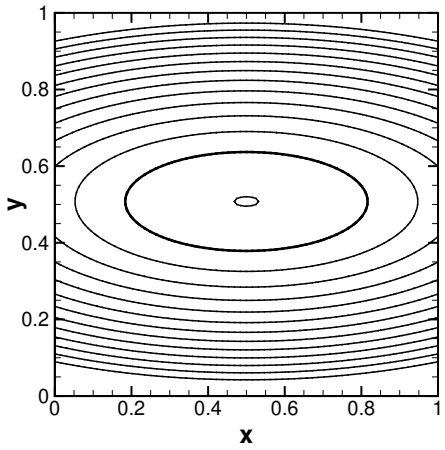


Figure 2.16: Initial condition ϕ_0 for the elliptic interface. The thick line denotes the interface. Contours run from -0.1 to 1.2 with interval 0.1.

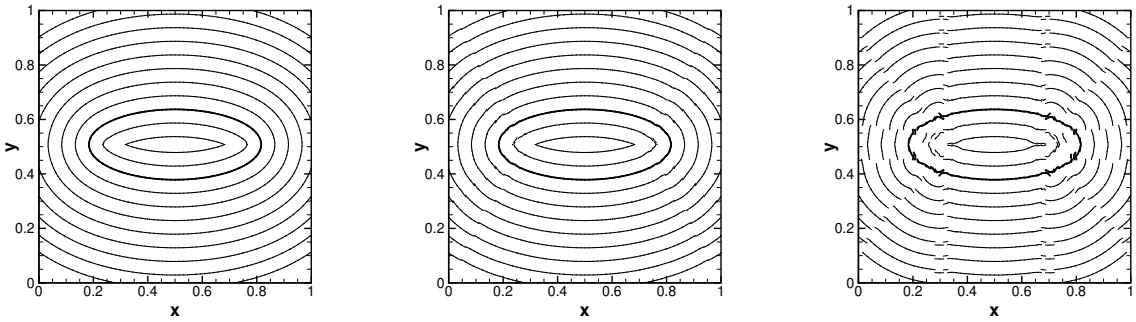
severe and destroy the interface eventually without the WLP. This is because the solution in interface cell are not affected by neighboring cells, oscillations there can grow without bound. The issue does not exist in the WLP, since it is not dependent on τ.

2.4.3 Square interface

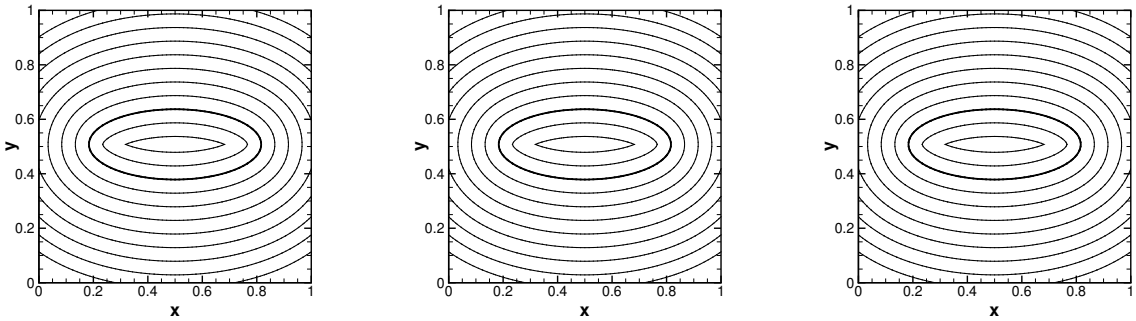
Generally, we do not need to apply limiters. However when the interface has extremely high or singular curvatures, such as the corners of a square interface, it is necessary to consider the limiter. Consider the square interface

$$\phi_0(x, y) = 0.8\,(\max\{|x - 0.5|, |y - 0.5|\} - R_0) \tag{2.57}$$

in a unit square, where $R_0 = 16.7h \approx 0.26$ such that all the interface cells are included in I_p. The sharp corner is rounded after projecting ϕ_0 to V_D^N in interface cells, and the zero level set is broken. This kind of numerical effect is common when non-smooth interfaces



(a) $\tau = 5$ (b) $\tau = 10$ (c) $\tau = 15$



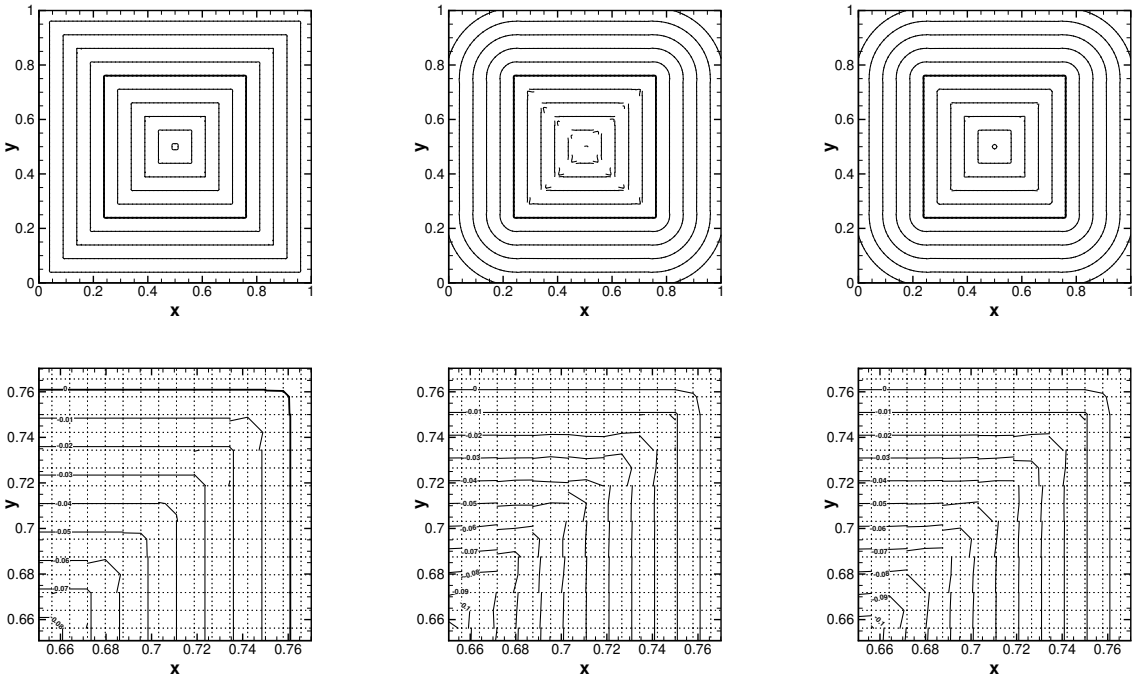
(d) $\tau = 5$ (e) $\tau = 10$ (f) $\tau = 15$

Figure 2.17: Evolution of ϕ_h for the elliptic interface. The interface cells are solved by evolving the PDE (2.3) in the top row, while the interface cells in I_p are fixed by the WLP method in the bottom row. Contours run from -0.1 to 0.4 with interval 0.05. The thick line is $\phi_h = 0$. $N = 3$ and $h = 1/64$.

are approximated by polynomials, which was also reported in [93]. Without the limiter, the oscillations are developed along the diagonals of the square, where characteristics converge, as shown in Fig. 2.18(b). While in Fig. 2.18(c), the oscillations are suppressed to some extend after applying the limiter, thus enhancing the quality of the solution.

As mentioned in Sec. 2.4.3, discontinuities of $\nabla\phi$ located on cell edges are easier to deal with. Here we rotate the square interface 45° clockwise to get a diamond interface

$$\phi_0(x, y) = 0.8\left(|x - 0.5| + |y - 0.5| - \sqrt{2}R_0\right) \tag{2.58}$$



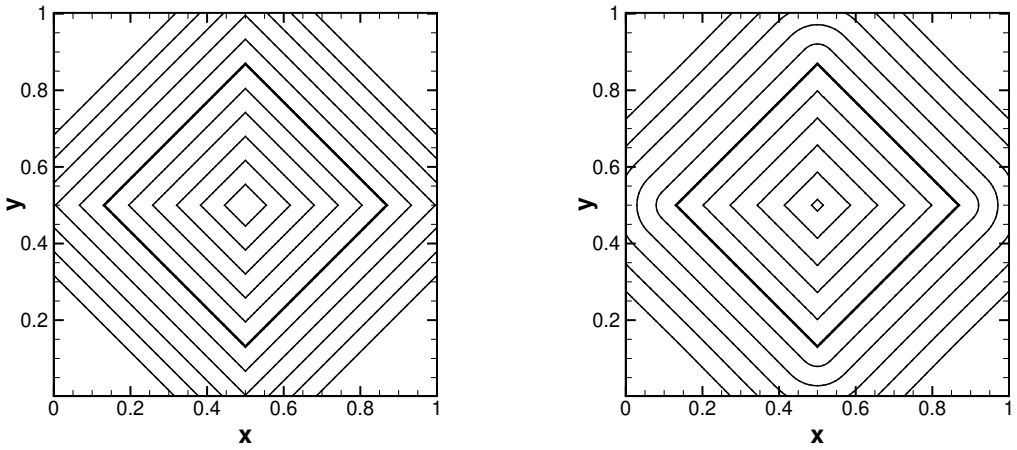
(a) $\tau = 0$ (b) $\tau = 0.4$ without limiter (c) $\tau = 0.4$ with limiter

Figure 2.18: Reinitialization of the square interface without (b) and with (c) limiters. The top row shows the solutions in the whole domain, with contours running from -0.25 to 0.25 with interval 0.05. The bottom row shows the close-up views of ϕ_h near the upper-right corner of the interface. For visualization purposes, each actual computational cell is divided into 2×2 cells demarcated by the dotted grid lines. $N = 3$ and $h = 1/64$.

where all kinks lie on cell edges, see Fig. 2.19(a). In this case, we obtain a perfect singed distance function in Fig. 2.19(b) without applying the limiter.

2.4.4 Interface deformation in a swirling vortex

We combine the reinitialization procedure with the level set equation to test the overall performance of our method in preserving the interface. Following [88, 120, 121], we consider



(a) $\tau = 0$ (b) $\tau = 0.4$

Figure 2.19: Reinitialization of a square interface with all kinks on the cell edges. No limiter is applied. Contour levels run from -0.25 to 0.25 with interval 0.05. $N = 3$ and $h = 1/64$.

a circular interface sheared by the velocity field

$$\mathbf{v}(\boldsymbol{x}, t) = \begin{bmatrix} -\sin^2(\pi x)\sin(2\pi y) \\ \sin^2(\pi y)\sin(2\pi x) \end{bmatrix} \tag{2.59}$$

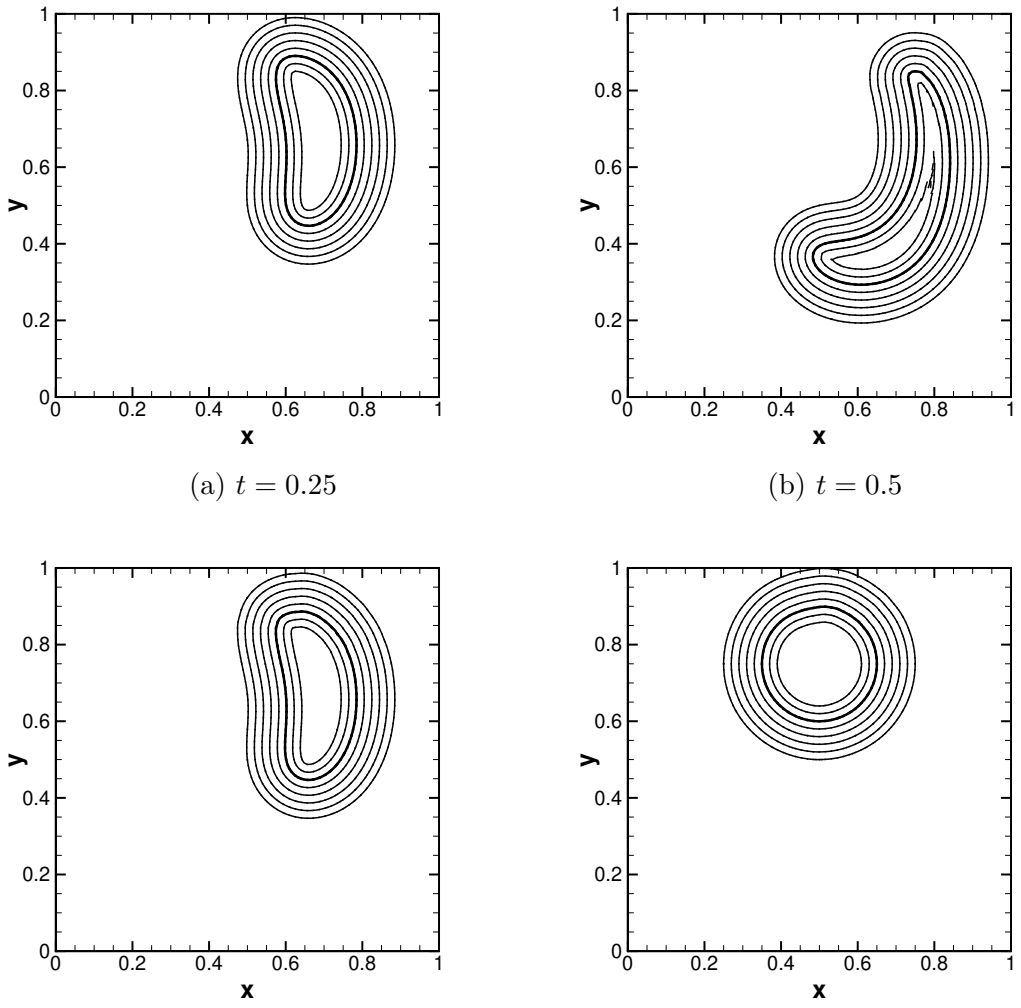
in a computational domain $[0, 1]^2$. Here we reverse the velocity field at $t = 0.5$ such that the interface will return to the initial location at $t = 1$. Note that t is the flow time, which should be distinguished from the pseudo time τ used in the level-set reinitialization. The initial circular interface is centered at $(0.5, 0.75)$ with a radius $r = 0.5$.

The level-set equation

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \boldsymbol{u}) = 0 \tag{2.60}$$

is solved by the DG method [96] with a local LF flux for spatial discretization and the third order TVD RK method for time integration. The time step is chosen to be $\Delta t = 0.1h$ such that the corresponding CFL number is 0.1. Details of the discretization are given in Sec.

3.2.2.

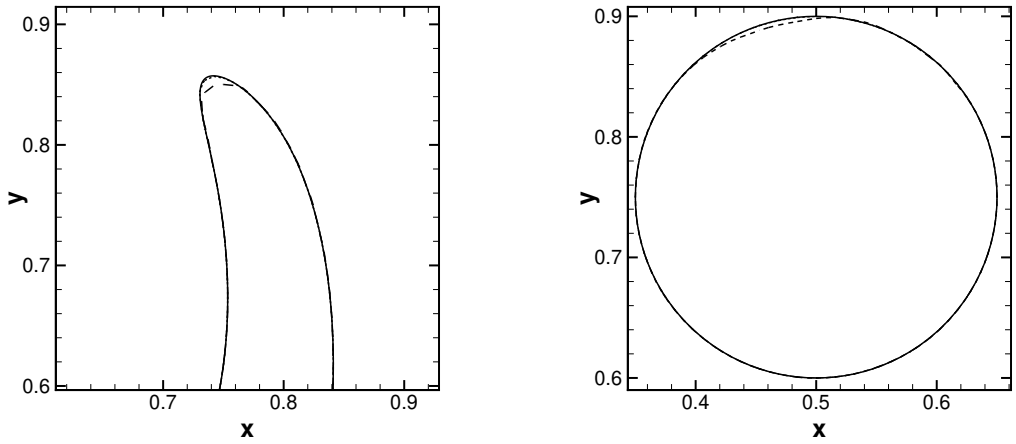


(a) $t = 0.25$ (b) $t = 0.5$

(c) $t = 0.75$ (d) $t = 1$

Figure 2.20: Circular interface sheared by a vortex that is reversed at $t = 0.5$. Reinitialization is performed every time step till a pseudo time $\tau = 0.1$. ϕ_h contours run from -0.04 to 0.1 with interval 0.02. The thick line is the zero level set. $N = 3$ and $h = 1/64$.

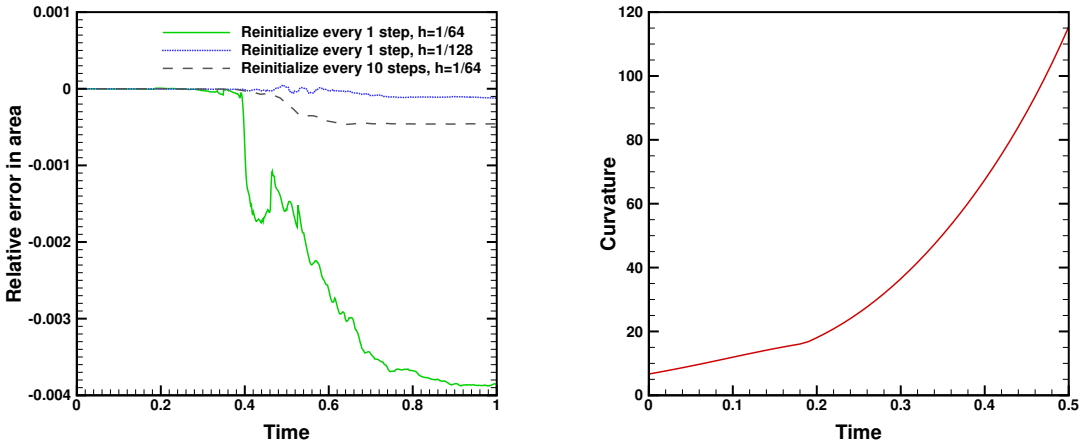
We observe in Fig. 2.20 that the signed distance function is maintained pretty well during the advection of the interface. A further comparison with the exact solutions in Fig. 2.21 reveals that the interface is preserved almost flawlessly (see the dashed line) except for a small portion near the top, which is caused by the error at the high-curvature tip as shown



(a) $t = 0.5$ (b) $t = 1$

Figure 2.21: Comparison of the interface. The dashed and dotted lines represent the numerical solutions obtained by reinitialization every 1 time step and reinitialization every 10 time steps, respectively. The solid lines represent the exact solution, which almost overlap with the dotted lines. $N = 3$ and $h = 1/64$.

in Fig. 2.21(a).



(a) (b)

Figure 2.22: (a) Relative error in area bounded by the interface. (b) Maximum curvature of the exact interface.

We keep track of the mass loss during the evolution of the interface in Fig. 2.22(a). Here

mass is the area bounded by the zero level set. The maximum error when we perform reinitialization every time step is around -0.39%. In the same mesh resolution, our result is comparable to that in [88] computed by the conservative level-set method. However, in terms of interface displacement, our solutions presented in Fig. 2.21 is at least as good as those in a finer mesh with $h = 1/128$ in [88]. Note that formulation of the conservative level-set method can only guarantee the conservation of the integral of ϕ, but not the location of the interface.

In practice it is usually not necessary to perform reinitialization every time step, especially for small time steps. For example, the reinitialization is performed every 10 steps in [108] and even 100 steps in [104]. Our simulation with reinitialization every 10 steps gives a much better result: the relative error is -0.045%, nearly $\frac{1}{10}$ of the original one. The interfaces at $t = 0.5$ and 1 are almost the same as the exact ones as shown in Fig. 2.21. Before the flow reversal, the maximum of the interface curvature increases with time and reaches 32 (i.e., $\frac{1}{2h}$) at $t = 0.28$, as shown in Fig. 2.22(b). The relative errors of area are -7.07×10^{-6} and -5.66×10^{-6} at $t = 0.28125$ for reinitializations every 1 step and 10 steps, respectively. This indicates that our method barely causes any mass loss for a smooth interface with a low curvature ($\lesssim \frac{1}{2h}$). The mass loss mostly occurs during $t \in [0.28, 0.72]$, when curvature is above $\frac{1}{2h}$. This is expected because the polynomial space cannot accurately resolve the curvature radius that is comparable to or smaller than the cell size h. If the curvature is too high, there is also a possibility that the interface cell is not detected by our method, such as Fig. 2.3(e). One way to resolve high curvature is mesh refinement. The results with $h = 1/128$ is included in Fig 2.22(a) for comparison. Although 1280 reinitializations are performed during $t \in [0, 1]$, the relative error is amazingly small: -0.012%. This error can be further reduced if we perform reinitialization every few time steps.

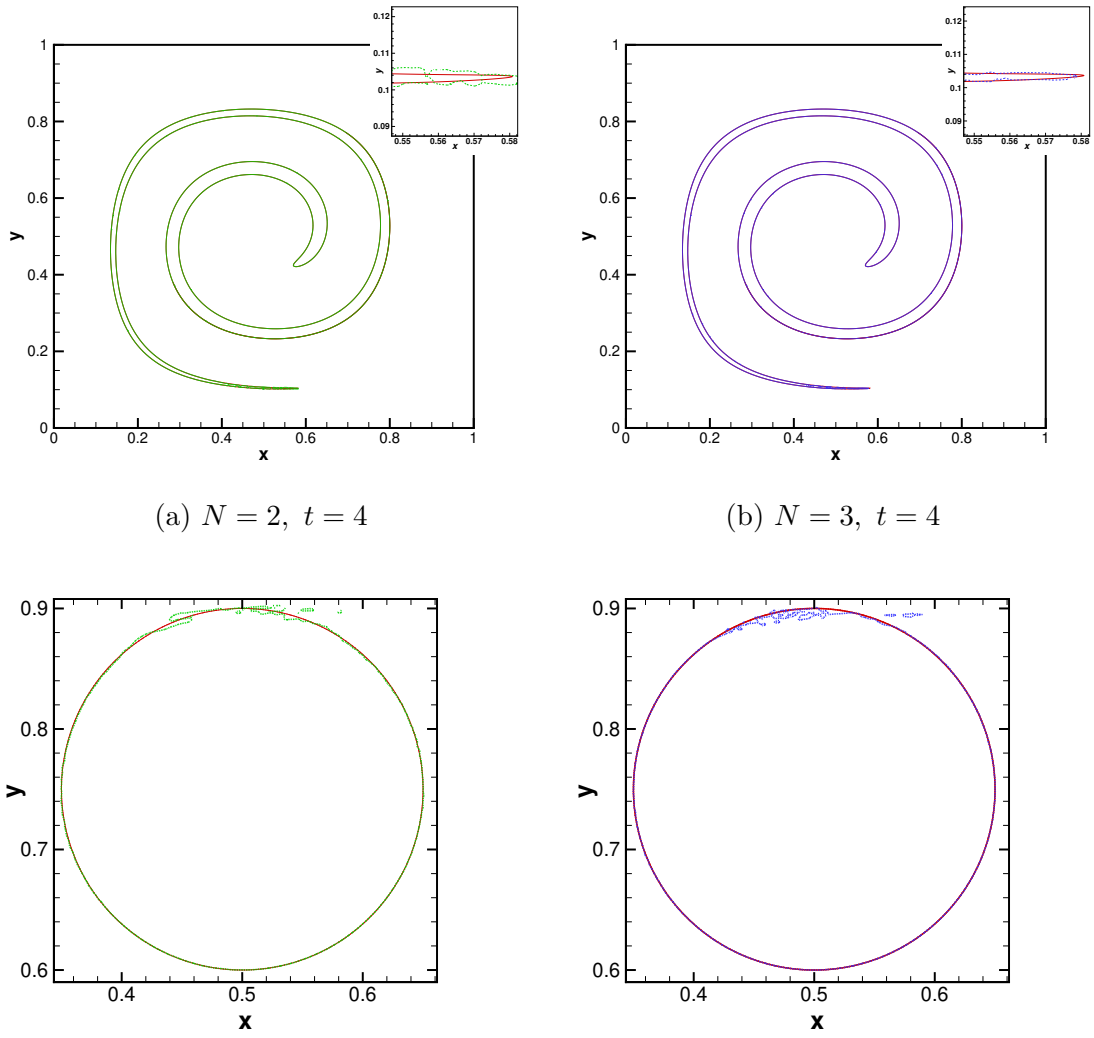
Long-time vortex test

To demonstrate the capability of our method in handling long filaments, we consider a slightly modified velocity field

$$\mathbf{v}(\boldsymbol{x}, t) = \begin{bmatrix} -\sin^2(\pi x)\sin(2\pi y)\cos(\pi t/8) \\ \sin^2(\pi y)\sin(2\pi x)\cos(\pi t/8) \end{bmatrix} \tag{2.61}$$

following [91, 92, 110, 122]. The interface gets fully stretched at $t = 4$ and restores to initial state at $t = 8$. Thanks to the easy implementation of adaptive mesh refinement in the DG framework, we use an adaptive mesh with the finest mesh size $h_{\min} = 1/512$ at the interface. For stability, we choose $\Delta t = 2.5 \times 10^{-4}$ for $N = 2$ and $\Delta t = 10^{-4}$ for $N = 3$. We perform 20 reinitializations within every time unit.

As shown in Fig. 2.23, the circle is mostly recovered at $t = 8$, except for some oscillations at the top. The area loss of $N = 2$ (0.15%) is comparable to those obtained by Gómez *et al.* [122] (0.11%) on a 256^2 main grid and Herrmann [110] (0.28%) with $h = 1/1024$. The third and fifth weighted essentially non-oscillatory (WENO) schemes are used in [122] and [110], respectively. Since Gómez *et al.* divided each cell around the interface into 4^2 subcells, their finest cell size is $h_{\min} = 1/1024$, which is comparable to our $N = 2$ in terms of degrees of freedom. However, our result with $N = 3$ (area loss of 0.29%) shows no improvement over $N = 2$, probably because the higher order method is more prone to numerical oscillations when the solution is non-smooth. This is consistent with the consensus that one should use p-refinement for the smooth part and h-refinement for the singular part of the solution in hp-finite element methods.



(a) $N = 2,\ t = 4$ (b) $N = 3,\ t = 4$

(c) $N = 2,\ t = 8$ (d) $N = 3,\ t = 8$

Figure 2.23: Circular interface under long-time shear. The solid line (red) is the exact solution, while the dashed line is the numerical solution. The insets show the zooms of the tail tip. Relative errors in area are 0.15% and 0.29% for $N = 2$ and $N = 3$, respectively. An adaptive mesh with $h_{\min} = 1/512$ is used and the maximal number of cells (at $t = 4$) is around 25000, which amounts to a 158^2 uniform mesh.

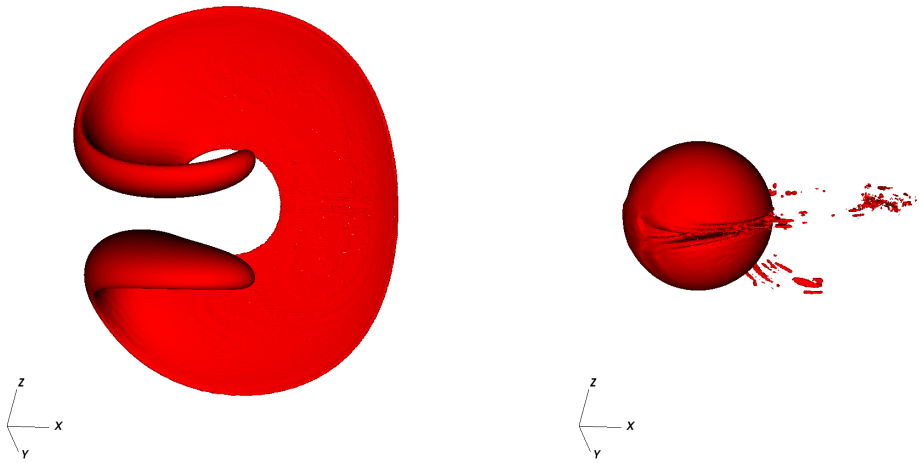
Vortex test in 3D

We consider a sphere with radius $r = 0.15$ centered at $(0.35, 0.35, 0.35)$ in a divergence free velocity field [84, 92, 123, 124]

$$\mathbf{v}(\boldsymbol{x}, t) = \begin{bmatrix} 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos(\pi t/3) \\ -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\cos(\pi t/3) \\ -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)\cos(\pi t/3) \end{bmatrix}. \tag{2.62}$$

The sphere gets fully stretched at $t = 1.5$ and restores to its initial shape at $t = 3$. We use an adaptive mesh with $h_{\min} = 1/256$ and the DG method with $N = 2$. The time step is chosen to be 2.5×10^{-4} and we perform 40 reinitializations within every time unit. The other parameters are the same as those in 2D calculations.

As shown in Fig. 2.24, oscillations appear near the equator of the sphere at $t = 3$. This is likely due to the lack of mesh resolution to describe the thin film: the smallest film thickness at $t = 1.5$ is less than $h_{\min}$. Our mass loss 1.62% is slightly greater than the second order adaptive level-set method by Min and Gibou [124], who reported a volume loss of 0.74% with $h_{\min} = 1/512$, but with a slightly different velocity field. The DG conservative level-set method by Jibben and Herrmann [92] seems to perform much better with a volume loss of 0.25% on a 128^3 uniform mesh with N=2. However, they only performed three reinitializations for the whole simulation while we did 120 reinitializations. This makes the errors not directly comparable. But since we use an adaptive mesh, the maximum number of unknowns is actually much smaller than the 128^3 mesh in [92]. We have to point out that the classical level-set description develops derivative discontinuities at the center of the thin film, while the conservative level set method has a second order diffusion term that regularizes the solution. This might be the reason why the conservative level-set may perform better with thin filaments or films. But the conservative level-set function has a steep transition layer at the interface and thus loses the numerical convenience of the signed distance function. Further discussions between the conservative level-set methods and the classical level-set methods are beyond the scope of this work.



(a) $t = 1.5$ (b) $t = 3$

Figure 2.24: Deformation of a sphere under shear. The relative error in volume is 1.62% at $t = 3$. The maximal number of cells is 460400, which amounts to a 77^3 uniform mesh. $N = 2$, $\beta_{\max} = 3$, $\lambda = 100$, $c_\xi = 0.1$, $c_\xi = 0.01$, $Q = 5$, and $h_{min} = 1/256$.

2.4.5 Rotation of a slotted disk

We consider the standard 2D benchmark: the rigid body rotation of a slotted disk [125] in the unit square $[0, 1]^2$. We use the velocity

$$\mathbf{v}(\boldsymbol{x}, t) = \begin{bmatrix} \pi(0.5 - y)/3.14 \\ \pi(x - 0.5)/3.14 \end{bmatrix}, \tag{2.63}$$

whose period of rotation is 6.28. The initial interface is a circular disk of radius 0.15 centered at $(0.5, 0.75)$ with a slot of width 0.05 and length 0.25 being cut off. Then ϕ_0 is computed according to the closet distance to the interface. We reinitialize the level-set solution every 10 time steps in two uniform meshes with $h = 1/128$ and $1/256$, and the corresponding time steps are $\Delta t = 0.001$ and 0.0005, respectively. Note that the limiter smooths out the sharp corners too much at early time of the simulation, therefore we do not apply limiters here.

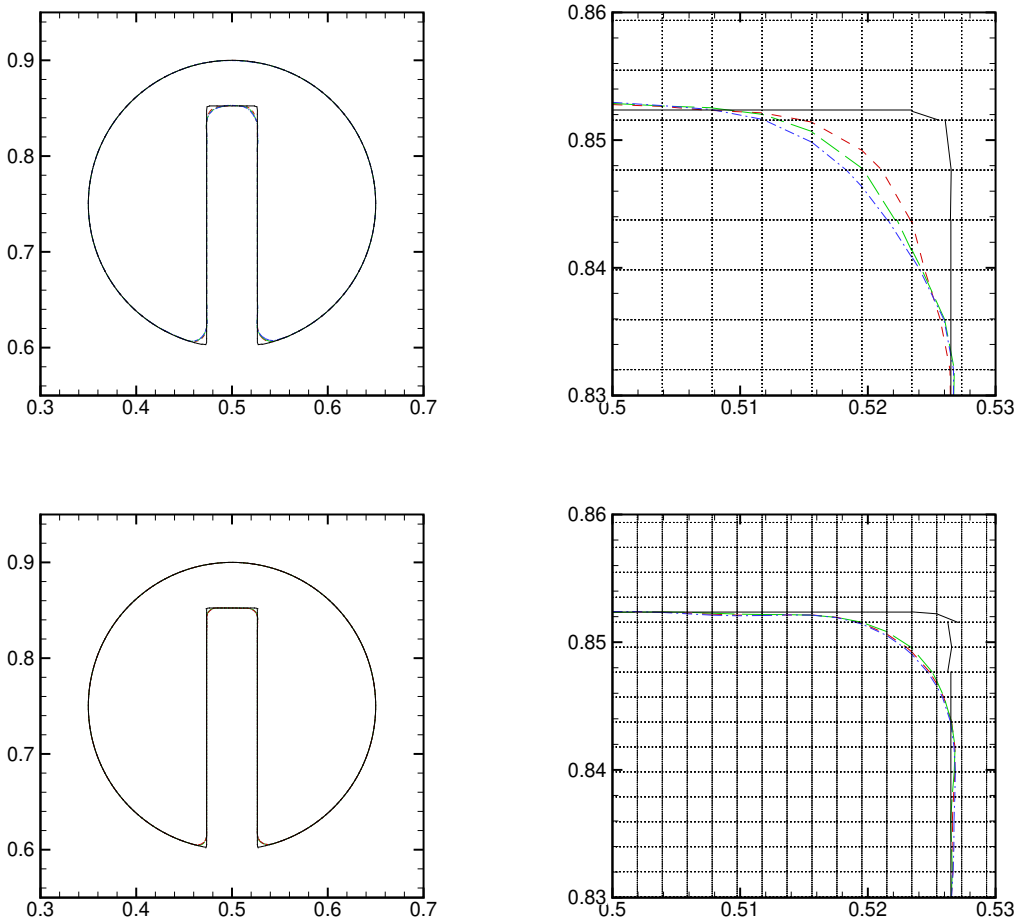


Figure 2.25: Rotation of the slotted disk. The top row shows the results with $h_{\min} = 1/128$, while the bottom row shows the results with $h_{\min} = 1/256$. The solid (black) line represent the initial interface. The dashed (red), long-dashed (green), and dash-dotted (blue) lines represent the interfaces after one, two, and three revolutions. The right column shows the zoom of the upper-right corner of the slot. For visualization purposes, each actual computational cell is divided into 2×2 cells demarcated by the dotted grid lines. $N = 3$. (color online)

The interface shapes are given in Fig. 2.25. In the left column, the interfaces after different full revolutions almost overlap. Close-up views at a sharp corner are shown in the right column. The initial interface here is the zero level set after projecting the exact signed distance function to the DG space. It is thus discontinuous at the corner. After rotation, the

corner is eventually rounded to a smooth curve with curvature radius being approximately $2h$. The relative mass losses after three revolutions, as shown in Figure 2.26, are around 0.1% and 0.06% for $h = 1/128$ and $1/256$, respectively. These are much smaller than 0.86% on a 256^2 mesh and 0.43% on a 512^2 mesh obtained by Hartmann *et al.* [82] using a fifth-order upstream central scheme. It should be noted that our DG method with $N = 3$ has ten degrees of freedom (DOF) in each cell. If we compare the DOF, our 128^2 mesh still has less DOF than the 512^2 finite difference mesh. More importantly, the majority of the mass loss occurs in the initial stage when the sharp corner cannot be well approximated by the DG polynomial space. Once the corner is rounded to such an extent that the curvature can be resolved by the computational mesh, further rotation causes very little mass loss. The mass loss in [82], however, grows almost linearly with the number of revolutions. In comparison, our method preserves interface better, especially in long-time simulations.

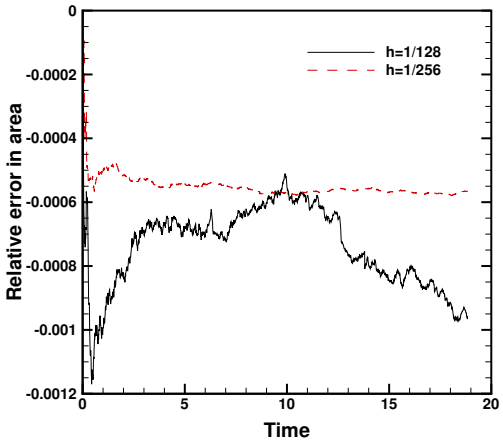


Figure 2.26: Relative error in the area of the slotted disk.

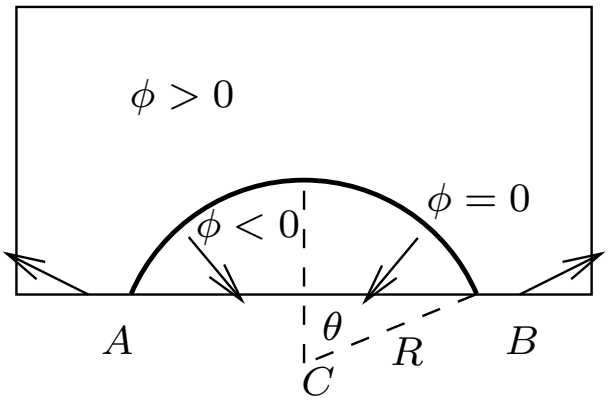


Figure 2.27: Schematic of a drop sitting on the solid surface in a rectangular domain. The arrows denote the directions of the characteristics.

2.4.6 Contact line

Reinitialization in 3D is discussed in Chap. 4, and here we only focus on 2D cases. With contact lines, the level-set reinitialization requires boundary conditions in part of the wall boundary, depending on the directions of the characteristics. In Fig. 2.27 for example, we have

$$S_\eta(\phi_0)\frac{\nabla\phi}{|\nabla\phi|}\cdot\boldsymbol{n}_w > 0$$

on AB, and

$$S_\eta(\phi_0)\frac{\nabla\phi}{|\nabla\phi|}\cdot\boldsymbol{n}_w < 0$$

elsewhere on the wall boundary, and $\boldsymbol{n}_w$ is the outward unit normal. We need characteristics going into the domain outside AB on the wall boundary, where boundary conditions have to be supplied. We refer to this part of the boundary as inflow boundary and denote it by

$$\partial\Omega_{in} = \{\boldsymbol{x}\in\partial\Omega : S(\phi_0)\nabla\phi\cdot\boldsymbol{n}_w < 0\} \tag{2.64}$$

where we impose the Dirichlet conditions for $\nabla\phi$.

As depicted in Fig. 2.27, we consider a drop resting on a solid surface. The initial interface is a part of a circle given by $|\boldsymbol{x} - \boldsymbol{x}_0| = R$ with $\boldsymbol{x}_0 = (0, -2 - R\cos(\theta))^\intercal$. Here R and θ are the radius of the circle and the contact angle, respectively. A and B are the contact lines (points in 2D) where the interface meets the solid wall.

The initial condition is given by

$$\phi_0 = \sqrt{x^2 + (y + 2 + R\cos\theta)^2} - R. \tag{2.65}$$

The computational domain is meshed into 64×32 uniform square cells.

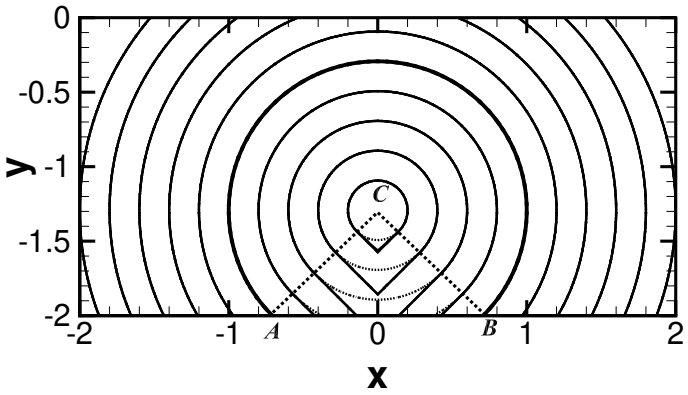


Figure 2.28: Reinitialization of a drop with a contact angle $\theta = 3\pi/4$. The solid lines are ϕ_h contours at $\tau = 1$, while the dotted lines are ϕ_0. Contours run from -0.8 to 1 with increment 0.2. $N = 3$ and $h = 1/16$.

We test two cases with different contact angles $\theta = 3\pi/4$ and $\pi/6$. In the first one we set $R = 1$ and $\theta = 3\pi/4$. The inflow boundary is the portion inside a drop: $\Omega_{in} = [-1, 1] \times 0$. Based on the interface normals at A and B, we impose the boundary condition:

$$\nabla\phi = \begin{cases} \left(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right)^\intercal, & \text{if } -1 \le x \le 0,\ y = 0, \\ \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right)^\intercal, & \text{if } 0 < x \le 1,\ y = 0. \end{cases} \tag{2.66}$$

As shown in Fig. 2.28, the solution ϕ_h at $\tau = 1$ is exactly the same as ϕ_0 outside the triangle ΔABC. But the solution in ΔABC is determined by the boundary condition (2.66). In this region, the contours are straight lines intersecting the boundary at the angle $\theta = 3\pi/4$.

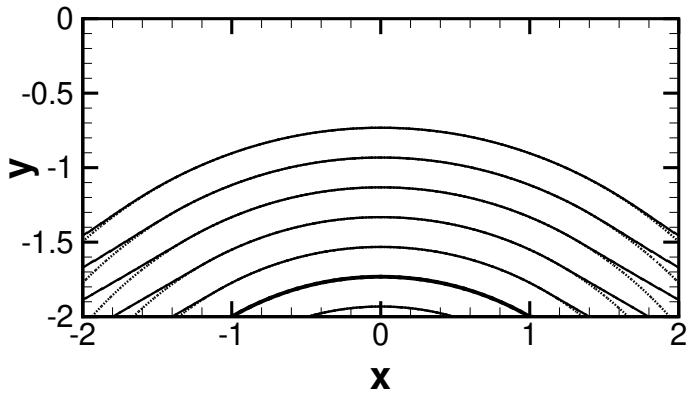


Figure 2.29: Reinitialization of a drop with a contact angle $\theta = \pi/6$. The solid lines are ϕ_h contours at $\tau = 1$, while the dotted lines are ϕ_0. Contours run from -0.2 to 1 with increment 0.2. $N = 3$ and $h = 1/16$.

The numerical method works equally well for the second case with an acute contact angle $\theta = \pi/6$ and $R = 2$. In this case, $\partial\Omega_{in} = ([-2,-1] \cup [1,2]) \times 0$, and the boundary condition for (2.3) is given by

$$\nabla\phi = \begin{cases} \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)^{\mathsf{T}}, & \text{if } -2 \le x \le -1,\ y = 0, \\ \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)^{\mathsf{T}}, & \text{if } 1 \le x \le 2,\ y = 0. \end{cases} \tag{2.67}$$

As shown in Fig. 2.29, the level sets in the region above the inflow boundary are straight lines intersecting the boundary at the angle $\theta = \pi/6$.

It should be noted that the $\nabla\phi$ condition on the inflow boundary is artificial, and the choice is not unique. The bottom line is that the boundary condition should maintain $|\nabla\phi| = 1$ and produce smooth level sets. For example, for circular interfaces in Figs. 2.28 and 2.29,

it is difficult to say whether the solution ϕ_h at pseudo time $\tau = 1$ is better than ϕ_0 with concentric level sets. But in the general case, when the interface shape is arbitrary, imposing $\nabla\phi$ condition based on the contact angle seems the most feasible.

2.5 Summary

In this chapter, we have developed a PDE-based method for level-set reinitialization in the DG framework. In this method, we first anchor $\nabla\phi$ in part of the interface cells by the WLP method, then compute $\nabla\phi$ in all the other cells by the DG method. Lastly, ϕ is recovered to preserve the interface by utilizing the interface location. Our method provides an easy treatment of the boundary condition for level-set reinitialization in 2D. In addition, a hybrid numerical flux is constructed to deal with highly distorted ϕ_0. Though limiting is not necessary in general, a limiter based of $\nabla\phi$ is developed to suppress oscillations in extreme nonsmooth or singular regions. Numerical results have proven that our method is highly accurate and stable in the long term.

Chapter 3

Level-set method for interfacial flows with moving contact lines

In this chapter, we couple the level-set equation with the flow equations to simulate interfacial flows. Ren and E's slip condition is used to relax the contact line singularity. Our method can easily obtain mesh-independent results, and CAH is implemented to automatically capture the motion of the contact line, such as advancing, receding, and pinning. The work in this chapter has been submitted to Journal of Computational Physics for publishcation [126].

3.1 Introduction

Ren and E' slip condition has two parts. The first part is the classic Navier's slip, which is imposed on the entire wall boundary. While the second part accounts for the balance between the friction force at the contact line and the unbalanced Young's stress, and is only valid at the contact line. Therefore, it is a sharp-interface model. This slip condition has gained popularities recently. Ren and E applied their slip condition to the level-set method and investigated contact line dynamics on heterogeneous surfaces [127]. This level-set work was later extended to moving contact lines with insoluble surfactants [128]. Zhang and Ren also investigated the influence of viscoelasticity on contact line dynamics using an immersed boundary method combined with the slip boundary condition [129]. Recently, Zhao and

Ren implemented the slip boundary condition on an interface-fitted mesh to study moving contact line problems [130].

After dealing with the stress singularity at the contact line, we still have another challenging issue, which is the contact angle hysteresis, since most solid surfaces are intrinsically rough or chemically heterogenous. In this case, the contact line stays pinned when the microscopic dynamic contact angle is between a receding contact angle θ_R and an advancing contact angle θ_A. The most popular approach for contact angle hysteresis was developed by Spelt for a level-set method [27]. An intermediate contact angle is obtained such that the contact line is pinned. If this angle is within the hysteresis window, the solution is accepted; otherwise, the solution is abandoned and the contact line is moved with prescribed contact angles. This idea was later extended to different methods, eg., the phase-field method [28], the volume-of-fluid method [30], the Lattice Boltzmann method [131], and the front-tracking method [132]. However, this approach relies on ghost cells outside the boundary to pin the contact line or to impose the contact angle condition, which can be challenging on curved boundaries and unstructured meshes. Recently, a thermodynamically consistent phase-field model was developed for contact angle hysteresis [133]. Since the dynamic contact angle is part of the solution instead of being imposed, this method is easy to implement and automatically captures the pinning, advancing, and receding of the contact line. Motivated by [133], we will show that the slip condition by Ren and E can also be easily modified to capture the contact angle hysteresis.

The rest of this chapter is organized as follows. We first introduce the governing equations and numerical methods in Section 3.2. We then explain how to incorporate contact angle hysteresis in Section 3.3. The numerical results, including code validation and mesh convergence studies, are given in Section 3.4.

3.2 Governing equations

Consider an incompressible system of two immiscible Newtonian fluids on an impermeable solid surface, as shown in Fig. 3.1. We use the level-set method to track the interface implicitly. The interface is represented by the zero level set of a signed distance function ϕ that is evolved by the the level-set equation

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0. \tag{3.1}$$

The regions with $\phi > 0$ and $\phi < 0$ are occupied by fluid 1 and fluid 2, respectively.

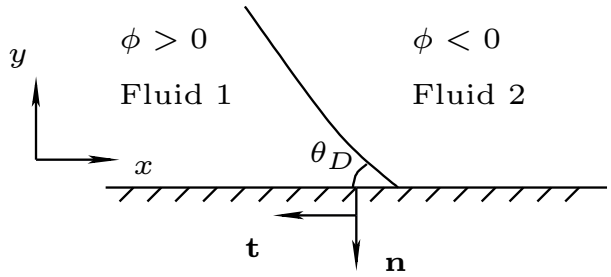


Figure 3.1: Schematic of a moving contact line on a solid substrate. The (microscopic) dynamic contact angle θ_D is defined with respect to fluid 1, which occupies the region with $\phi > 0$.

Following the level-set literature, the two-phase system can be treated as a single fluid with density and viscosity given by

$$\rho(\phi) = H_\epsilon(\phi)\rho_1 + (1 - H_\epsilon(\phi))\rho_2 \tag{3.2}$$

and

$$\mu(\phi) = H_\epsilon(\phi)\mu_1 + (1 - H_\epsilon(\phi))\mu_2, \tag{3.3}$$

where H_ϵ is a smooth Heaviside function defined in (2.50), 2ϵ is the interface thickness, and

the subscripts $_1$ and $_2$ denote fluid 1 and 2, respectively.

For the surface tension term, we adopt the tensor form [134]

$$\boldsymbol{\tau}_\phi = \sigma\delta_\epsilon(\phi)|\nabla\phi|\boldsymbol{T}(\boldsymbol{n}_\phi), \tag{3.4}$$

where σ is the surface tension, $\delta_\epsilon(\phi) = H'_\epsilon(\phi)$ is the smooth Dirac delta function, $\boldsymbol{n}_\phi = \frac{\nabla\phi}{|\nabla\phi|}$ is the unit normal to the interface, and $T(\boldsymbol{n}_\phi) = \boldsymbol{I} - \boldsymbol{n}_\phi \otimes \boldsymbol{n}_\phi$. It should be noted that a term $|\nabla\phi|$ is included in (3.4) to deal with the case where ϕ deviates from a singed distance function.

The incompressible two-phase flow is governed by the momentum equation

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = \nabla\cdot(-p\boldsymbol{I} + \boldsymbol{\tau} + \boldsymbol{\tau}_\phi) + \rho\mathbf{g} \tag{3.5}$$

and the continuity equation

$$\nabla\cdot\boldsymbol{u} = 0, \tag{3.6}$$

where $\boldsymbol{u}$ is the fluid velocity, p is the pressure, $\boldsymbol{\tau} = \mu[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T]$ is the viscous stress, and $\mathbf{g}$ is the gravitational acceleration.

On the solid wall, we impose the no-penetration condition

$$\boldsymbol{n}\cdot(\boldsymbol{u} - \boldsymbol{u}_w) = 0 \tag{3.7}$$

in the normal direction and Ren and E's slip condition in the tangential direction, where $\boldsymbol{u}_w$ denotes the wall velocity and $\boldsymbol{n}$ is the outward pointing unit normal to the wall boundary (see Fig. 3.1). Motivated by the level-set work in [127], the slip condition by Ren and E can

be expressed as

$$\beta(\phi)\boldsymbol{u}_s = -\left[\boldsymbol{n}\cdot\boldsymbol{\tau} + \sigma\delta_\epsilon(\phi)\left(\cos\theta_S - \frac{\nabla\phi\cdot\boldsymbol{n}}{|\nabla\phi|}\right)\nabla\phi\right]\cdot\boldsymbol{T}(\boldsymbol{n}), \tag{3.8}$$

where $\boldsymbol{u}_s = \boldsymbol{u} - \boldsymbol{u}_w$ is the slip velocity, θ_S is the prescribed static contact angle, $\beta(\phi) = \beta_N + \beta_{CL}\delta_\epsilon(\phi)|\nabla\phi\cdot\boldsymbol{T}(\boldsymbol{n})|$, β_N is the friction coefficient between the Newtonian fluids and the solid wall, and β_{CL} is the friction coefficient at the contact line. For any vector $\boldsymbol{a}$, $\boldsymbol{a}\cdot T(\boldsymbol{n})$ gives the tangential component of $\boldsymbol{a}$ in the plane with normal $\boldsymbol{n}$. Thus the right-hand side of (3.8) is a force (per unit area) tangential to the solid wall. This force includes contributions from the viscous stress and the unbalanced Young's stress. It is obvious that $\boldsymbol{u}_s$ given by (3.8) is tangential to the wall since $\boldsymbol{u}_s\cdot\boldsymbol{n} = 0$. At the contact line, $\boldsymbol{u}_s$ is simply the contact line velocity relative to the solid wall. It should be noted that (3.8) has no limitation on spatial dimensions and it reduces to the boundary condition in [127] in 2D.

In the limit of vanishing ϵ, (3.8) recovers the 3D version of the Navier slip condition

$$\beta_N\boldsymbol{u}_s = -(\boldsymbol{n}\cdot\boldsymbol{\tau})\cdot\boldsymbol{T}(\boldsymbol{n}) \tag{3.9}$$

away from the contact line (i.e., $\phi \neq 0$) and

$$\beta_{CL}\boldsymbol{u}_s = -\sigma\left(\cos\theta_S - \cos\theta_D\right)\frac{\nabla\phi\cdot\boldsymbol{T}(\boldsymbol{n})}{|\nabla\phi\cdot\boldsymbol{T}(\boldsymbol{n})|} \tag{3.10}$$

at the contact line (i.e., $\phi = 0$), where θ_D is the microscopic dynamic contact angle and we have used the geometric relation $\cos\theta_D = \boldsymbol{n}_\phi\cdot\boldsymbol{n} = \frac{\nabla\phi\cdot\boldsymbol{n}}{|\nabla\phi|}$. The term $\sigma\left(\cos\theta_S - \cos\theta_D\right)$ is exactly the unbalanced [11] Young's stress. In a 2D flow as shown in Fig. 3.1, these two

equations reduce to the familiar formulations in [11, 127]:

$$\beta_N u_s = \boldsymbol{n} \cdot \boldsymbol{\tau} \cdot \mathbf{t} = \mu \frac{\partial u}{\partial y} \tag{3.11}$$

away from the contact line and

$$\beta_{CL} u_s = \sigma \left(\cos\theta_S - \cos\theta_D \right) \tag{3.12}$$

at the contact line, where u_s is the slip velocity in x-direction, $\mathbf{t}$ is the unit tangent vector to the wall, and u is the x-component of fluid velocity.

The Navier slip condition (3.11) determines a slip length $l_s = \frac{\mu}{\beta_N}$. Similar to viscosity, the phenomenological parameter β_N may take different values in the two fluids, although we will use a single constant in this paper for simplicity. Equation (3.12) agrees with the molecular-kinetic theory at the leading order [135]. It should be noted that β_N and β_{CL} have different dimensions.

A numerical advantage of the slip condition (3.8) is that it does not require us to manually impose the contact angle condition on the geometry of the interface. Instead, both the slip velocity $\boldsymbol{u}_s$ and the dynamic contact angle θ_D are part of the solution. If the contact line is at static equilibrium, both viscous stress $\boldsymbol{\tau}$ and $\boldsymbol{u}_s$ vanish and (3.8) recovers the static contact angle, i.e., $\theta_D = \theta_S$. If the contact line moves, then the relation (3.12) at the contact line predicts a θ_D that differs from θ_S. In particular, $\theta_D > \theta_S$ if the contact line advances and $\theta_D < \theta_S$ is the contact line recedes.

In summary, the governing equations include the level-set equation (3.1) for the interface and the Navier-Stokes equations (3.5) and (3.6) for the flow field, with the latter supplemented by the slip boundary condition (3.8).

3.2.1 Weak form of Navier-Stokes equations

In this subsection, we derive the weak form of the Navier-Stokes equations (3.5) and (3.6) supplemented with the slip condition (3.8).

Assume that the domain boundary $\partial\Omega$ can be partitioned into three parts based on boundary conditions: $\partial\Omega_D$ where the Dirichlet condition $\boldsymbol{u} = \boldsymbol{u}_b$ is imposed, $\partial\Omega_N$ where the traction condition (natural boundary condition) is imposed, and the solid wall $\partial\Omega_w$ where the no-penetration condition and the slip condition (3.8) are imposed. We seek the weak solution $(\boldsymbol{u}, p) \in \mathcal{U} \times \mathcal{P}$, with solution spaces

$$\mathcal{U} = \{\boldsymbol{u} \in H^1(\Omega)^d : \boldsymbol{u} = \boldsymbol{u}_b \text{ on } \partial\Omega_D, \boldsymbol{u} \cdot \boldsymbol{n} = \boldsymbol{u}_w \cdot \boldsymbol{n} \text{ on } \partial\Omega_w\}, \tag{3.13}$$

and

$$\mathcal{P} = L^2(\Omega)^d, \tag{3.14}$$

where d denotes the spatial dimension of the flow. The corresponding test spaces are

$$\mathcal{U}_0 = \{\boldsymbol{u} \in H^1(\Omega)^d : \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial\Omega_D, \boldsymbol{u} \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega_w\}, \tag{3.15}$$

and $\mathcal{P}$, respectively.

Taking the inner product of (3.5) with the test function $\mathbf{v} \in \mathcal{U}_0$ and the inner product of (3.6) with $q \in \mathcal{P}$ in Ω, we obtain the weak form

$$\begin{aligned}\left(\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right), \mathbf{v}\right) &= \left(\boldsymbol{n} \cdot \left(-p\boldsymbol{I} + \boldsymbol{\tau} + \boldsymbol{\tau}_\phi\right), \mathbf{v}\right)_{\partial\Omega} \\ &\quad + (p, \nabla \cdot \mathbf{v}) - (\boldsymbol{\tau} + \boldsymbol{\tau}_\phi, \nabla \mathbf{v}) + (\rho \mathbf{g}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathcal{U}_0\end{aligned} \tag{3.16}$$

and

$$(\nabla \cdot \boldsymbol{u}, q) = 0, \quad \forall q \in \mathcal{P}, \tag{3.17}$$

where $(\cdot,\cdot)$ denotes the inner product in Ω and $(\cdot,\cdot)_{\partial\Omega}$ denotes the inner product on $\partial\Omega$.

The boundary inner product on the right-hand side of (3.16) can be further written as

$$\begin{aligned}
&\left(\boldsymbol{n}\cdot\left(-p\boldsymbol{I}+\boldsymbol{\tau}+\boldsymbol{\tau}_\phi\right),\mathbf{v}\right)_{\partial\Omega}\\
=&\left(\boldsymbol{n}\cdot\left(-p\boldsymbol{I}+\boldsymbol{\tau}+\boldsymbol{\tau}_\phi\right),\mathbf{v}\right)_{\partial\Omega_w}\\
=&\left(-\boldsymbol{n}p+\boldsymbol{n}\cdot\boldsymbol{\tau}+\sigma\delta_\epsilon(\phi)|\nabla\phi|\boldsymbol{n}\cdot(\boldsymbol{I}-\boldsymbol{n}_\phi\otimes\boldsymbol{n}_\phi),\mathbf{v}\right)_{\partial\Omega_w}\\
=&\left(\boldsymbol{n}\cdot\boldsymbol{\tau}-\sigma\delta_\epsilon(\phi)(\boldsymbol{n}\cdot\boldsymbol{n}_\phi)\nabla\phi,\mathbf{v}\right)_{\partial\Omega_w},
\end{aligned} \tag{3.18}$$

where we have used $\mathbf{v}=\mathbf{0}$ on $\partial\Omega_D$ and zero traction $\boldsymbol{n}\cdot(-p\boldsymbol{I}+\boldsymbol{\tau}+\boldsymbol{\tau}_\phi)=\mathbf{0}$ on $\partial\Omega_N$ in the first equality, and $\boldsymbol{n}\cdot\mathbf{v}=0$ on $\partial\Omega_w$ in the third equality. Here we consider zero traction for simplicity and an additional boundary inner product should be considered if the traction on $\partial\Omega_N$ is nonzero. Considering the slip condition (3.8), Eq. (3.16) can be further written as

$$\begin{aligned}
\left(\rho\left(\frac{\partial\boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\nabla\boldsymbol{u}\right),\mathbf{v}\right)+\beta(\phi)\left(\boldsymbol{u}-\boldsymbol{u}_w,\mathbf{v}\right)_{\partial\Omega_w}=&-\left(\sigma\delta_\epsilon(\phi)\cos\theta_S\nabla\phi,\mathbf{v}\right)_{\partial\Omega_w}\\
&+(p,\nabla\cdot\mathbf{v})-(\boldsymbol{\tau}+\boldsymbol{\tau}_\phi,\nabla\mathbf{v})+(\rho\mathbf{g},\mathbf{v}),\quad\forall\mathbf{v}\in\mathcal{U}_0.
\end{aligned} \tag{3.19}$$

The weak solution $(\boldsymbol{u},p)$ can be found by solving (3.19) and (3.17).

3.2.2 Numerical methods

The governing equations are solved by the finite element method on a quadrilateral mesh with hierarchical adaptive mesh refinement based on the open-source deal.II library [118, 136]. Due to different natures of these equations, we solve the level-set equation and the Navier-

Stokes equations separately with the former solved by DG and the latter solved by the mixed finite element method.

We focus on 2D problems in this chapter. Consider a triangulation $\mathcal{T}_h$, consisting of non-overlapping quadrilaterals, of the computational domain Ω. In addition to the discontinuous finite-element space V_D^N (2.10), we define a continuous finite-element space

$$V_C^N = \{\phi \in C^0(\Omega) : \phi \in Q^N(K), \forall K \in \mathcal{T}_h\}, \tag{3.20}$$

where N denotes the polynomial degree. In this work, we take $N = 3$ for ϕ, $N = 2$ for $\boldsymbol{u}$, and $N = 1$ for p. The finite dimensional solution spaces for ϕ_h, $\boldsymbol{u}_h$, and p_h are $\mathcal{F}_h = V_D^3$, $\mathcal{U}_h = (V_C^2)^2 \cap \mathcal{U}$, and $\mathcal{P}_h = V_C^1$, respectively, where we have used the subscript $_h$ to denote finite-dimensional approximations. The test space for $\boldsymbol{u}_h$ is simply $\mathcal{U}_{h,0} = (V_C^2)^2 \cap \mathcal{U}_0$.

Since the flow is incompressible, the level-set equation (3.1) can be written as a conservation law

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\boldsymbol{u}\phi) = 0. \tag{3.21}$$

In each element K, by taking the inner product of (3.21) with the test function $\psi \in \mathcal{F}_h$ and performing integration by parts, we obtain the weak formulation

$$\left(\frac{\partial \phi_h}{\partial t}, \psi\right)_K + \left(\hat{H}(\phi_h^-, \phi_h^+), \psi\right)_{\partial K} - (\boldsymbol{u}_h \phi_h, \nabla \psi)_K = 0, \quad \forall \psi \in \mathcal{F}_h, \tag{3.22}$$

where $\hat{H}(\phi_h^-, \phi_h^+)$ denotes the numerical flux that approximates $\boldsymbol{n} \cdot \boldsymbol{u}_h \phi_h$, ϕ_h^- and ϕ_h^+ are the inside and outside values of ϕ_h on the element boundary ∂K, and $\boldsymbol{n}$ is the outward pointing unit normal to ∂K. We use the local Lax-Friedrichs flux

$$\hat{H}(\phi_h^-, \phi_h^+) = \boldsymbol{n} \cdot \boldsymbol{u}_h \frac{\phi_h^- + \phi_h^+}{2} - \frac{\alpha}{2}(\phi_h^+ - \phi_h^-), \tag{3.23}$$

where $\alpha = \max(|\boldsymbol{n} \cdot \boldsymbol{u}_h|)$ and the maximum is taken over the relevant element edge. The semi-discrete weak form (3.22) is integrated by the third-order TVD RK method to advance ϕ_h^n to ϕ^{n+1}. To decouple $\boldsymbol{u}_h$ from ϕ_h, the $\boldsymbol{u}_h$ values at intermediate time levels between t^n and t^{n+1}, which are required by the TVD RK method, are obtained by explicit extrapolations from $\boldsymbol{u}_h^{n-1}$ and $\boldsymbol{u}_h^n$. Note that we choose the third-order TVD RK for the sake of stability rather than accuracy, and the overall scheme is only second-order accurate in time.

The discontinuous solution ϕ_h cannot be differentiated. We thus map it to a continuous function $\phi_C \in V_C^3$ by least squares before feeding it to flow equations:

$$(\phi_C - \phi_h, \psi) = 0, \quad \forall \psi \in V_C^3. \tag{3.24}$$

The flow equations (3.19) and (3.17) are discretized by the Crank-Nicolson scheme. In each time step, we seek the weak solution $(\boldsymbol{u}_h^{n+1}, p_h^{n+\frac{1}{2}}) \in \mathcal{U}_h \times \mathcal{P}_h$ satisfying the discretized weak form

$$\begin{aligned}
&\left(\rho(\phi_C^{n+\frac{1}{2}})\left(\tfrac{\boldsymbol{u}_h^{n+1}-\boldsymbol{u}_h^n}{\Delta t} + \boldsymbol{u}_h^* \cdot \nabla \boldsymbol{u}_h^{n+\frac{1}{2}} + \tfrac{1}{2}(\nabla \cdot \boldsymbol{u}_h^*)\boldsymbol{u}_h^{n+\frac{1}{2}}\right), \mathbf{v}\right) \\
&\qquad + \beta(\phi_C^{n+\frac{1}{2}})\left(\boldsymbol{u}_h^{n+\frac{1}{2}} - \boldsymbol{u}_w, \mathbf{v}\right)_{\partial\Omega_w} = -\left(\sigma\delta_\epsilon(\phi_C^{n+\frac{1}{2}})\cos\theta_S \nabla\phi_C^{n+\frac{1}{2}}, \mathbf{v}\right)_{\partial\Omega_w} \\
&\qquad\qquad + \left(p_h^{n+\frac{1}{2}}, \nabla \cdot \mathbf{v}\right) - \left(\mu(\phi_C^{n+\frac{1}{2}})(\nabla \boldsymbol{u}_h^{n+\frac{1}{2}} + (\nabla \boldsymbol{u}_h^{n+\frac{1}{2}})^T), \nabla \mathbf{v}\right) \\
&\quad - \left(\sigma\delta_\epsilon(\phi_C^{n+\frac{1}{2}})\left|\nabla\phi_C^{n+\frac{1}{2}}\right|(\boldsymbol{I} - \boldsymbol{n}_\phi^{n+\frac{1}{2}} \otimes \boldsymbol{n}_\phi^{n+\frac{1}{2}}), \nabla \mathbf{v}\right) + (\rho(\phi_C^{n+\frac{1}{2}})\mathbf{g}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathcal{U}_{h,0}
\end{aligned} \tag{3.25}$$

and

$$-(\nabla \cdot \boldsymbol{u}_h^{n+1}, q) = 0, \quad \forall q \in \mathcal{P}_h, \tag{3.26}$$

where $\boldsymbol{u}_h^{n+\frac{1}{2}} = \frac{\boldsymbol{u}_h^n + \boldsymbol{u}_h^{n+1}}{2}$, $\phi_C^{n+\frac{1}{2}} = \frac{\phi_C^n + \phi_C^{n+1}}{2}$, $\boldsymbol{n}_\phi^{n+\frac{1}{2}} = \frac{\nabla\phi_C^{n+\frac{1}{2}}}{\left|\nabla\phi_C^{n+\frac{1}{2}}\right|}$, and $\boldsymbol{u}_h^*$ is an explicit approximation of $\boldsymbol{u}^{n+\frac{1}{2}}$ by a linear extrapolation from $\boldsymbol{u}_h^n$ and $\boldsymbol{u}_h^{n-1}$. Here we have adopted the

skew-symmetric form for the convection term and the scheme is unconditionally stable if the density is a constant.

Thanks to the explicit approximation $\boldsymbol{u}_h^*$, Eqs. (3.25) and (3.26) lead to a linear saddle point problem:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} U \\ P \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}, \tag{3.27}$$

where U and P are solution vectors for $\boldsymbol{u}_h^{n+1}$ and $p_h^{n+\frac{1}{2}}$, respectively. The asymmetric square block A comes from (3.25) excluding the pressure term while the non-square block B comes from (3.26). The size of the resulting matrix is relatively small in 2D, and we solve (3.27) by the direct sparse linear solver UMFPACK [137].

In each time step, the solution procedure can be summarized as follows:

(i) Check the mesh and the interface. If necessary, perform local refinement and coarsening such that the interface region is covered by the finest mesh and bulk region is covered by the coarsest mesh. Transfer data from the old mesh to the new mesh if the mesh is altered.

(ii) Based on $\boldsymbol{u}_h^n$, $\boldsymbol{u}_h^{n-1}$, and ϕ_h^n, solve (3.22) to obtain ϕ_h^{n+1}.

(iii) Reinitialize ϕ_h^{n+1} to a signed distance function if necessary.

(iv) Map ϕ_h^{n+1} to ϕ_C^{n+1} in the continuous finite space by solving (3.24).

(v) Based on $\boldsymbol{u}_h^n$, $\boldsymbol{u}_h^{n-1}$, ϕ_C^n, and ϕ_C^{n+1}, solve the linear system of (3.25) and (3.26) to obtain $\boldsymbol{u}_h^{n+1}$ and $p_h^{n+\frac{1}{2}}$.

3.3 Contact angle hysteresis

With the contact angle hysteresis, the contact line remains pinned for a range of contact angles, the minimum of which is referred to as the receding contact angle θ_R and the maximum is referred to as the advancing contact angle θ_A. The contact line advances if $\theta_D > \theta_A$, recedes if $\theta_D < \theta_R$, and is pinned if $\theta_R \leq \theta_D \leq \theta_A$.

When the contact line moves, we can still use the method in the previous section with θ_S replaced by θ_A for the advancing contact line and by θ_R for the receding contact line. Special treatment is needed when the contact line is pinned. In this case, the no-slip condition $\boldsymbol{u}_s = \mathbf{0}$ is required at the contact line. To make this condition consistent with (3.8), we adopt the formula

$$\beta(\phi)\boldsymbol{u}_s = -(\boldsymbol{n}\cdot\boldsymbol{\tau})\cdot\boldsymbol{T}(\boldsymbol{n}), \tag{3.28}$$

which recovers the no-slip condition at $\phi = 0$ while still maintaining the Navier slip condition away from the contact line. Thus it provides a smooth transition between the pinned and moving contact lines. With (3.28) in place of the slip boundary condition by Ren and E, the weak form of the momentum equation (3.19) is updated to

$$\left(\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right), \mathbf{v}\right) + \beta(\phi)\left(\boldsymbol{u} - \boldsymbol{u}_w, \mathbf{v}\right)_{\partial\Omega_w} = -\left(\sigma\delta_\epsilon(\phi)\frac{\nabla\phi\cdot\boldsymbol{n}}{|\nabla\phi|}\nabla\phi, \mathbf{v}\right)_{\partial\Omega_w} + (p, \nabla\cdot\mathbf{v}) - (\boldsymbol{\tau} + \boldsymbol{\tau}_\phi, \nabla\mathbf{v}) + (\rho\mathbf{g}, \mathbf{v}), \quad \forall\mathbf{v}\in\mathcal{U}_0. \tag{3.29}$$

when the contact line is pinned. It should be noted that the only difference between this equation and (3.19) is the first term on the right-hand side: $\cos\theta_S$ in the inner product on $\partial\Omega_w$ is now replaced by $\cos\theta_D = \frac{\nabla\phi\cdot\boldsymbol{n}}{|\nabla\phi|}$.

The relation between θ_D and (θ_R, θ_A) can be inferred from ϕ. Motivated by the phase-field

method for hysteresis in [133], we define

$$F(\theta) = \int_e \delta_\epsilon(\phi) \left(\cos\theta - \frac{\nabla\phi \cdot \boldsymbol{n}}{|\nabla\phi|} \right) \mathrm{dS}, \tag{3.30}$$

where e is an element edge (element face in 3D) in the neighborhood of the contact line on $\partial\Omega$. Obviously, $\theta_D > \theta$ if $F(\theta) > 0$ and $\theta_D < \theta$ if $F(\theta) < 0$. Then the status (advancing, receding, or pinned) of the contact line can be determined from the signs of $F(\theta_A)$ and $F(\theta_R)$.

In the first term on the right-hand side of the discretized weak form (3.25), we need to perform integration on the boundary edges on $\partial\Omega_w$. The contact angle hysteresis can be incorporated with a little modification to this boundary integral. For each element edge e on $\partial\Omega_w$, we first evaluate $F(\theta_A)$ and $F(\theta_D)$ with $\phi = \phi_C^{n+\frac{1}{2}}$ in (3.30), and then proceed as follows with boundary inner product in (3.25):

- If $F(\theta_A) > 0$, then $\theta_D > \theta_A$ and the contact line advances. Set $\theta_S = \theta_A$.
- If $F(\theta_R) < 0$, then $\theta_D < \theta_R$ and the contact line recedes. Set $\theta_S = \theta_R$.
- Otherwise, $\theta_A \geq \theta_D \geq \theta_R$ and the contact line is pinned. Set $\cos\theta_S = \frac{\nabla\phi_C^{n+\frac{1}{2}} \cdot \boldsymbol{n}}{\left|\nabla\phi_C^{n+\frac{1}{2}}\right|}$, i.e., $\theta_S = \theta_D$.

Since $\theta_A > \theta_D$, we have $\cos\theta_A < \cos\theta_D$ and the third case corresponds to $F(\theta_R) \geq 0 \geq F(\theta_A)$. These operations are performed on all boundary edges on $\partial\Omega_w$; but only the boundary integral on the edges in the contact line region, i.e., where $\delta_\epsilon(\phi)$ is non-zero, is affected. All the other operations remain the same as those in Section 3.2.2. Thus, the contact angle hysteresis can be easily included in the formulation for moving contact line problems.

In our method, no matter whether the contact line is pinned or not, the dynamic angle θ_D is computed from the momentum equation and thus the momentum balance is automatically

satisfied. There is no need to use any special technique as in [30] to determine θ_D that satisfies local momentum balance when the contact line is pinned.

It should be noted that β_{CL} for the pinned contact line essentially plays a role of penalty parameter to enforce the no-slip condition. Thus, in order to achieve a good pinning performance, we may need to choose a large enough β_{CL} for the pinned contact line. In other words, we need a large contact line friction to resist contact line motion and thus pin the contact line.

3.4 Numerical results and discussions

We consider seven test cases. We first validate the interfacial flow part of our code by computing a bubble rising problem. In the second case, we examine the capability of our method to capture topological change in the simulation of the pinch-off of a pendent drop. In the test case of an advancing interface in plane Poiseuille flow, we systematically investigate the use of the slip condition (3.8) in moving contact line problems including mesh convergence and parameter justification. In the test case of drop spreading, we come up with a computational strategy for predictive simulations. In the test case of a pinned drop in plane Poiseuille flow, we validate the capability of our method in pinning the contact lines. In the test case of advancing and receding interfaces in plane Poiseuille flow, we demonstrate that our method correctly captures the transition between pinning and moving. Finally, in the test case of sliding drop on an inclined wall, we further demonstrate the capability of our method in capturing hysteresis. All the parameters are dimensionless except for the bubble rising problem.

3.4.1 Bubble rising

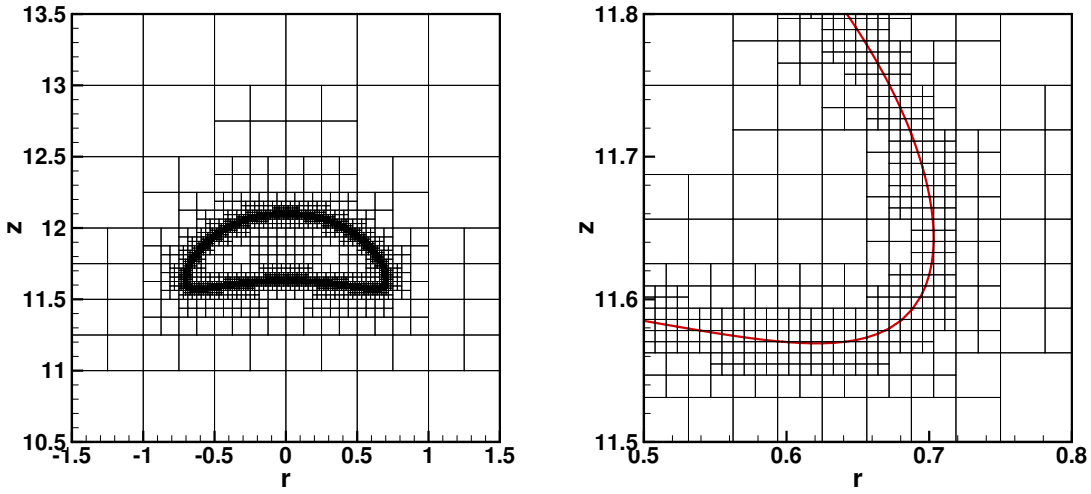
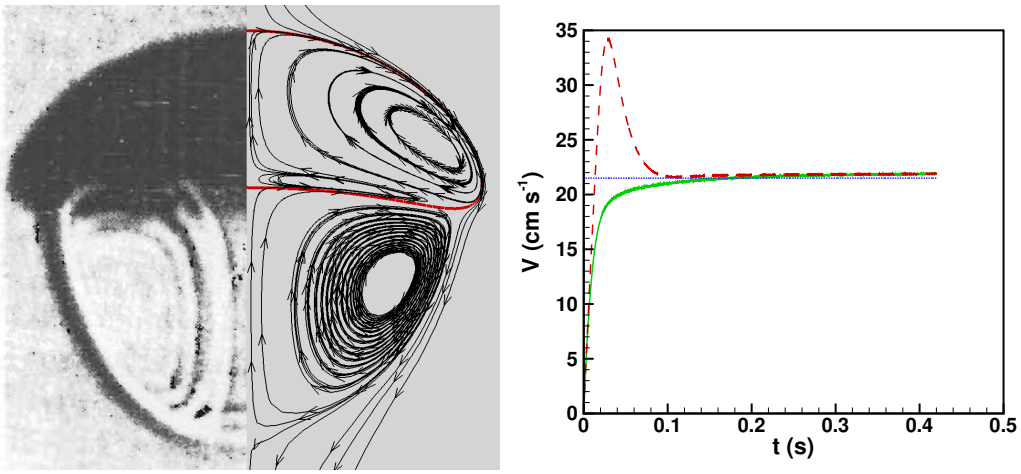


Figure 3.2: Illustration of the adaptive mesh refinement. The left panel shows the mesh around the bubble while the right panel shows a close-up view at the rim of the bubble. The thick solid (red) line in the right panel denotes the interface, i.e., the $\phi = 0$ level set.

Hnat and Buckmaster conducted experiments with spherical-cap air bubbles rising in incompressible liquids to study the steady-state shapes and terminal velocities [1], wherein the results were reproduced numerically by many others [34, 138, 139, 140, 141] for code validation. In this numerical test, we will use the experiment of Fig. 1a in [1] with the following parameters: liquid density $\rho_l = 0.8755$ g cm^{-3}, gas density $\rho_g = 0.001$ g cm^{-3}, liquid viscosity $\mu_l = 1.18$ P, gas viscosity $\mu_g = 0.01$ P, surface tension $\sigma = 32.2$ dyn cm^{-1}, gravitation acceleration $g = 980$ cm s^{-2}, and bubble radius $R_0 = 0.61$ cm. Due to axisymmetry, we only compute the right half of the meridian plane. In the r-z plane, the computational domain is a rectangle of $(0, 8R_0) \times (0, 30R_0)$ and the initially spherical bubble is released from $(0, 5R_0)$. We use an adaptive mesh with minimum mesh size $h_{\min} = \frac{R_0}{64}$ at the interface and maximum mesh size $h_{\max} = R_0$ far away from the bubble, as shown in Fig. 3.2. The half-width of the interface is taken to be $\epsilon = 1.5h_{\min}$.

Our numerical results are displayed in Fig. 3.3. The bubble shape and the wake structure at the steady state are in good agreement with the experiment, as shown in Fig. (3.3a). We also keep track of the instantaneous velocities at the top and the bottom of the bubble, which are shown in Fig. (3.3b). The steady-state velocity in our numerical test, which is 21.89 cm s^{-1}, is slightly larger than 21.5 cm s^{-1} reported by the experiments. We note that a similar terminal speed, 21.90 cm s^{-1}, was obtained in [139].



(a) (b)

Figure 3.3: The steady-state bubble shape (a) and the instantaneous velocities at the top and the bottom of the bubble (b). The left half of (a) is the experimental image adapted from [1].

3.4.2 Pinch-off of a pendant drop

In this example, we show that our method is able to capture topological changes. Following [140, 142, 143], we consider a hemispherical drop placed at the nozzle of a capillary tube, which grows with inflow fluid, and eventually leads to the pinch-off of a drop. The inflow velocity profile on the upper boundary is given as:

$$\boldsymbol{u} = \begin{cases} -2V\left(1-\left(\frac{r}{R_0}\right)^2\right)\mathbf{e}_z, & \text{if } r < R_0 \\ \mathbf{0}, & \text{otherwise,} \end{cases} \tag{3.31}$$

where R_0 is the radius of the tube, and V is the average velocity in the capillary tube, which is set to $V = 0.0172$. Meanwhile, we impose zero stress conditions on the lower boundary, symmetric and slip boundary conditions on the left and right boundaries, respectively. Due to axisymmetry of the flow, we just need to compute half of the meridian plane in a computational domain $[0, 4R_0]\times[0, 10R_0]$ in the r-z plane. In order to match the dimensionless groups in [143], we use $R_0 = 1$, $\rho_1 = 1$, $\rho_2 = 0.5$, $\mu_1 = \mu_2 = 0.178$, $\sigma = 1$, $g = 0.930$, and $V = 0.0172$ such that Bond number $Bo = \frac{(\rho_1-\rho_0)gR_0^2}{\sigma} = 0.465$, Capillary number $\mathrm{Ca} = \frac{\mu_1 V}{\sigma} = 3.05\times10^{-3}$, Weber number $\mathrm{We} = \frac{\rho_1 V^2 R_0}{\sigma} = 2.95\times10^{-4}$, and ratios of the density and viscosity are $\frac{\rho_1}{\rho_2} = 2$ and $\frac{\mu_1}{\mu_2} = 1$, respectively. Besides, we define a dimensionless time $t^* = \frac{tV}{R_0}$ to present our results.

Since the flow is axisymmetric, we only compute half of the meridian plane. The computational domain is $[0, 4R_0]\times[0, 10R_0]$ in the r-z plane. We take $h_{\min} = \frac{1}{64}R_0$ near the interface and $h_{\max} = \frac{1}{2}R_0$ away from the interface.

We display a sequence of snapshots depicting the evolution of the drop in Fig. 3.4. At about $t^* = 3.3495$, pinch-off takes place with several secondary drops detached from the primary drop. The radius of the primary drop is around $1.33R_0$, which is close to $1.30R_0$ extracted from Fig. 2 of [143]. We present the contours of ϕ_h near the instant of pinch-off in Fig. 3.5 as well. Complicated as the topological change is, our method can reproduce a signed distance function near the interface while taking care of the process of pinch-off. After the pinch-off, a long filament is generated on top of the primary drop. The subsequent retraction of the filament creates capillary waves, which eventually breaks the long filament into shorter ones, see Fig. 3.5(c,d). Note that in Fig. 3.5(d), the radius of the filament is $0.1h_{\min}$, while the

radius of its bulk at the lower end is $h_{\min}$.

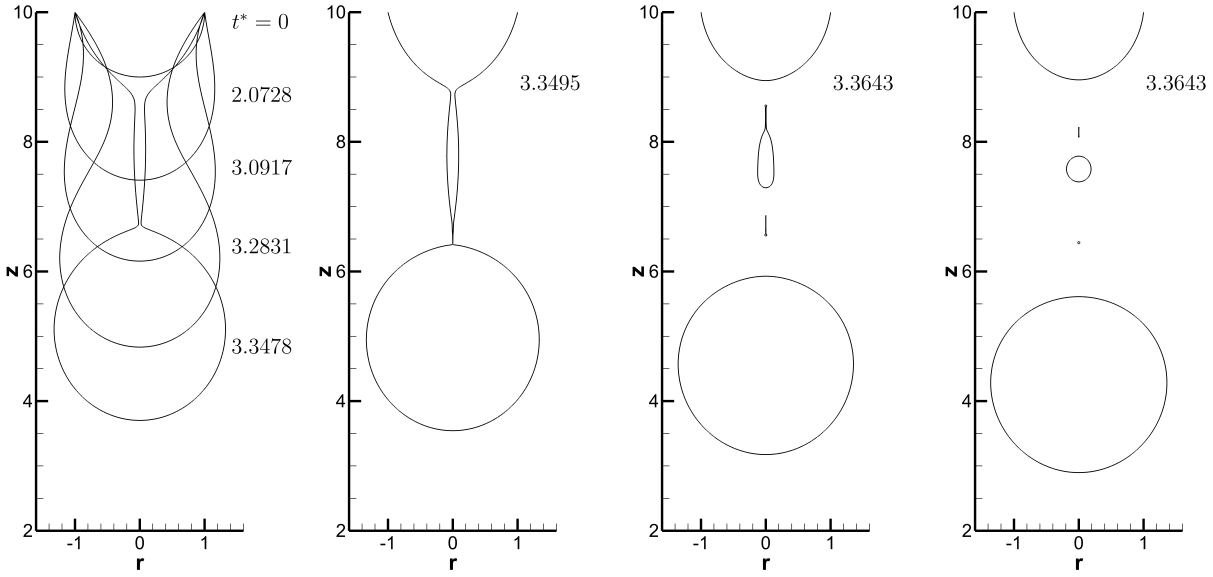
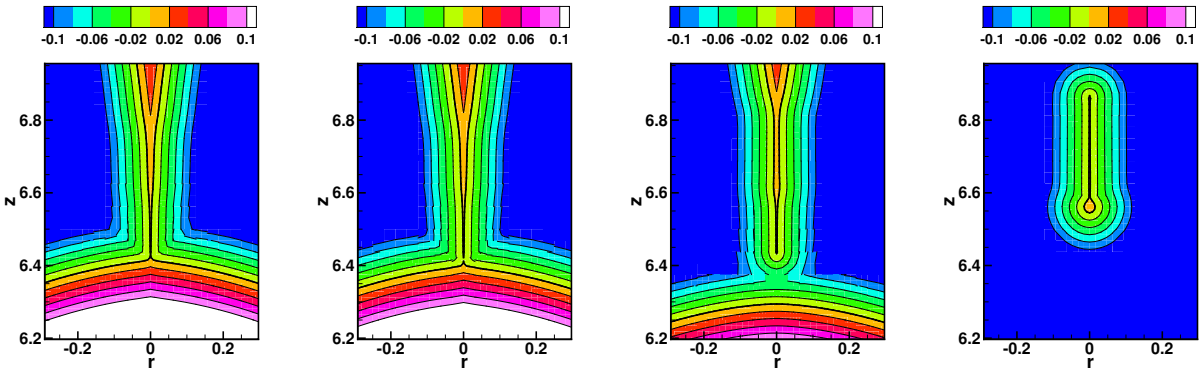


Figure 3.4: Snapshots of the pinch-off process of a pendant drop.



(a) $t^* = 3.3495$ (b) $t^* = 3.3500$ (c) $t^* = 3.3529$ (d) $t^* = 3.3643$

Figure 3.5: ϕ contours near the instant of pinch-off. The thick line denotes the interface.

We have to point out that the successful capture of the filament much thinner than the cell size in this test case is fortuitous: the shock wave of ϕ inside the filament lies exactly on the axis of symmetry. In the general case, the WLP method may have trouble in capturing sub-cell structures for the following two reasons. First, if both sides of the filament pass through

the same cell, similar to Fig. 2.3(f), our method fails to identify that cell as an interface cell. Of course, this can be resolved if we consider all the possibilities of interface cells. But the second reason is more substantial: the polynomial approximation suffers from large errors in case of shock waves or other sub-cell structures inside the cell. This limitation applies to all methods based on high-degree polynomial approximation. In interfacial flows, mesh refinement is probably the only way to go because the flow field also needs to be resolved.

3.4.3 Advancing interface in plane Poiseuille flow

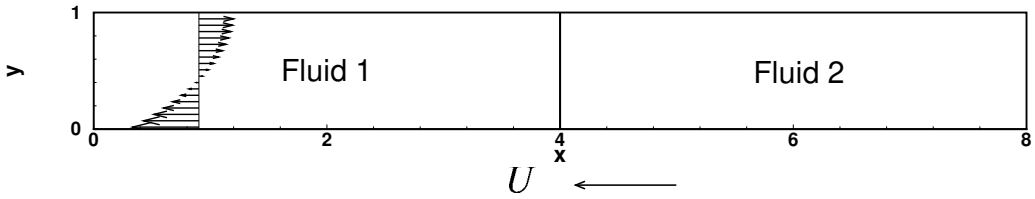


Figure 3.6: Schematic of an advancing interface in plane Poiseuille flow.

We consider the steady plane Poiseuille flow of two immiscible fluids with identical viscosity μ. Inertia is neglected. The frame is fixed to the interface, i.e., the walls are moving horizontally with constant speed U while the interface is stationary. In the following, all numbers are made dimensionless by channel half width W, surface tension σ, and fluid viscosity μ, unless otherwise specified. Under this normalization, we have $W = \sigma = \mu = 1$. Since the flow is symmetric, we only compute the lower half of the channel. The computational domain is a rectangle of $(0, 8) \times (0, 1)$ in the x-y plane, with $y = 0$ being the moving wall and $y = 1$ being the axis of symmetry, as shown in Fig. 3.6. We run the simulation with an initially flat interface at $x = 4$ until a steady state is achieved. The capillary number is defined as $Ca = \mu U/\sigma$. We first investigate mesh convergence of our method and then analyze the variables that affect contact line dynamics.

Mesh convergence

Mesh convergence is crucial to all predictive numerical simulations. This task gets more challenging with a moving contact line: the slip length has to be well resolved to produce mesh-independent results [15, 27, 40]. There are three microscopic length scales: mesh size $h_{\min}$, (half) thickness of the interface ϵ, and slip length l_s. Here, only l_s is physically relevant and the other two are numerical. Our goal is to find a way to generate results that are independent of $h_{\min}$ and ϵ.

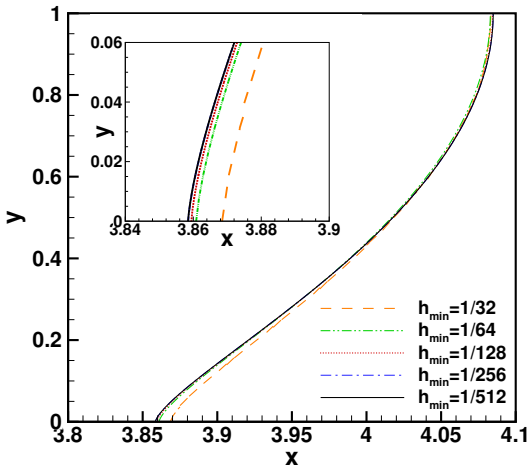
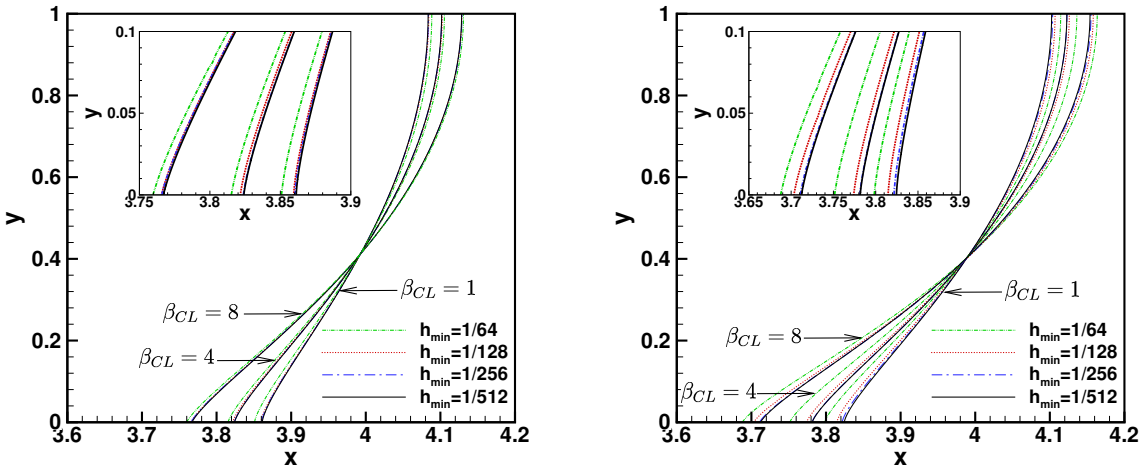


Figure 3.7: Mesh convergence for a fixed interface thickness. The inset is a close-up view at the contact line. $\epsilon = \frac{1.5}{128}$, $\beta_N = 100$ (such that $l_s = 0.01$), $\beta_{CL} = 1$, $Ca = 0.03$, $\theta_S = 90°$.

These lengths define two independent dimensionless groups, eg., $\frac{h_{\min}}{\epsilon}$ and $\frac{\epsilon}{l_s}$. We first investigate $\frac{h_{\min}}{\epsilon}$, i.e., how to choose mesh size to achieve mesh convergence for a given interfacial thickness. We fix the interfacial thickness at $\epsilon = \frac{1.5}{128}$, and conduct simulations with $h_{\min} = \frac{1}{32}, \frac{1}{64}, \cdots, \frac{1}{512}$, which corresponds to $\frac{\epsilon}{h_{\min}} = 0.375, 0.75, \cdots, 6$. The slip length is taken to be $l_s = 0.01$, which is well resolved by ϵ as explained later. We can easily see the convergence in the steady-state interface profile, as shown in Fig. 3.7, as $h_{\min}$ reduces. The curves of $h_{\min} = \frac{1}{256}$ and $\frac{1}{512}$ overlap; the errors are negligible even for $h_{\min} = 1/128$ and

$h_{\min} = 1/64$. The result gets unsatisfactory when the mesh is coarsened to $h_{\min} = \frac{1}{32}$. We thus come to the first criterion for mesh convergence:

$$\frac{h_{\min}}{\epsilon} \leq \frac{4}{3}. \tag{3.32}$$



(a) $l_s = 0.01$ ($\beta_N = 100$) (b) $l_s = 0.005$ ($\beta_N = 200$)

Figure 3.8: Mesh convergence at fixed l_s and $\frac{h_{\min}}{\epsilon}$. The insets are close-up views at the contact line. $\epsilon = 1.5h_{\min}$, $Ca = 0.03$, $\theta_S = 90°$.

We next investigate $\frac{\epsilon}{l_s}$, i.e., the sharp-interface limit with respect to ϵ when l_s is fixed. We keep $\frac{h_{\min}}{\epsilon} = \frac{1}{1.5}$ fixed while refining mesh, such that ϵ is always well resolved. We test $\beta_N = 100$ and 200, which correspond to $l_s = 0.01$ and 0.005, respectively, as illustrated in Fig. 3.8. The detailed errors for $\beta_N = 100$ are given in Table 3.1. Curves with different β_{CL} show that this parameter does not affect mesh convergence. Convergence is achieved if $h_{\min} \leq \frac{1}{128}$ for $l_s = 0.01$ and $h_{\min} \leq \frac{1}{256}$ for $l_s = 0.005$. It is tempting to conclude a convergence criterion based on $\frac{h_{\min}}{l_s}$. However, the curve of $h_{\min} = \frac{1}{64}$ and $\epsilon = \frac{1.5}{128}$ in Fig. 3.7 shows much better convergence than that of $h_{\min} = \frac{1}{64}$ and $\epsilon = \frac{1.5}{64}$ in Fig. 3.8(a), which

suggests $\frac{\epsilon}{l_s}$ to be a better choice. We thus have the second criterion for convergence:

$$\frac{\epsilon}{l_s} \leq \frac{150}{128}. \tag{3.33}$$

	$\beta_{CL} = 1$		$\beta_{CL} = 4$		$\beta_{CL} = 8$	
$h_{\min}$	H	E_r	H	E_r	H	E_r
1/64	0.2373	6.76%	0.2893	4.26%	0.3444	2.96%
1/128	0.2249	1.17%	0.2806	1.13%	0.3364	0.56%
1/256	0.2235	0.54%	0.2778	0.11%	0.3356	0.34%
1/512	0.2223	0	0.2775	0	0.3345	0

Table 3.1: Relative errors in the height of the spherical cap shaped interface. $\epsilon = 1.5h_{\min}, \theta_S = 90°\beta_N = 100, Ca = 0.03$. The height H is the distance in the x direction measured from the contact line to the apex of the interface. E_r is the relative error in H, where we have used the solution at $h_{\min} = 1/512$ as the reference.

In summary, we should choose the mesh size and the interface thickness according to $h_{\min} \lesssim \epsilon \lesssim l_s$ to obtain numerical results that are independent of $h_{\min}$ and ϵ. This looks very similar to the criterion for the sharp-interface limit in the phase-field method [144]. The advantage of the level-set method is that it does not require a lot of mesh cells across the narrow-band interface. In the rest of this paper, we conservatively choose $\epsilon = 1.5h_{\min}$ as suggested in [42] and make sure $l_s \geq 1.28h_{\min}$.

Contact line dynamics

The contact line dynamics in the slip condition (3.8) is controlled by β_N, β_{CL}, and θ_S. The effect of θ_S is well understood, but it is still unclear how β_N and β_{CL} quantitatively affect contact line dynamics. Here we try to answer this question by comparing with the well-established Cox theory [7].

According to the matched asymptotic analysis by Cox, to the leading order, the apparent

contact angle θ_{app} and the static contact angle θ_S are connected by

$$g(\theta_{app}) = g(\theta_S) + \mathrm{Ca}\ln(L/L_s) \tag{3.34}$$

due to viscous bending of the interface. Here L_s is the slip length that characterizes the inner region of the contact line and L is characteristic length of the macroscopic flow. Note that this L_s is a "physical" scale associated with the contact line and may be different from the l_s. To distinguish these two length scales, we refer to L_s as the effective slip length and l_s as the numerical slip length hereafter. For the plane Poiseuille flow, we can simply take $L = W$. The function g is given by

$$g(\theta) = \int_0^\theta \frac{d\theta}{f(\theta)} \tag{3.35}$$

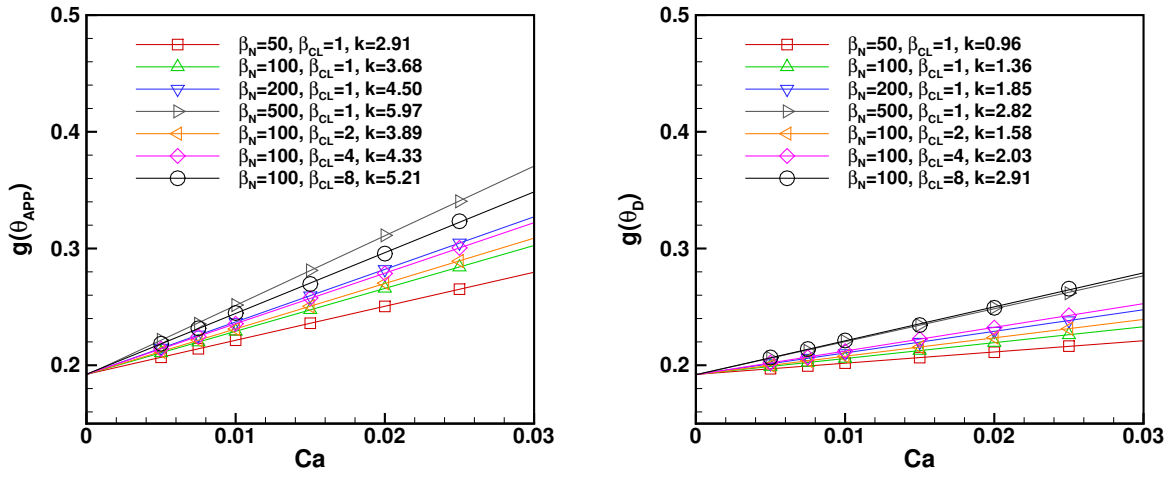
where

$$f(\theta, r_\mu) = \frac{2\sin\theta\left\{r_\mu^2(\theta^2-\sin^2\theta)+2r_\mu\left[\theta(\pi-\theta)+\sin^2\theta\right]+\left[(\pi-\theta)^2-\sin^2\theta\right]\right\}}{r_\mu(\theta^2-\sin^2\theta)[(\pi-\theta)+\sin\theta\cos\theta]+\left[(\pi-\theta)^2-\sin^2\theta\right](\theta-\sin\theta\cos\theta)} \tag{3.36}$$

and r_μ is the viscosity ratio between the receding and advancing fluids.

We first study the influence of Ca on θ_{app} and θ_D. Here θ_{app} is determined by fitting a circle to the deformed interface [144, 145] and θ_D is directly measured from the $\phi = 0$ level curve at the contact line. The results for various combinations of (β_N, β_{CL}) with fixed $\theta_S = 90°$ are shown in Fig. 3.9. Figure 3.9(a) confirms that $g(\theta_{app})$ is linear in Ca, in accordance with the Cox theory (3.34). Not surprisingly, $g(\theta_D)$ is also a linear function of Ca, consistent with the numerical observation in [11] as well as the molecular-kinetic theory [9]. It should be noted that $g(\theta_S) = 0.1921$, which is exactly the $y-$intercept of both $g(\theta_{app})$ and $g(\theta_D)$.

This linear behavior is also observed for different θ_S, as shown in Fig. 3.10. According to (3.34), the slope of the $g(\theta_{app})$–Ca curve is only dependent on L_s and is independent of



(a) (b)

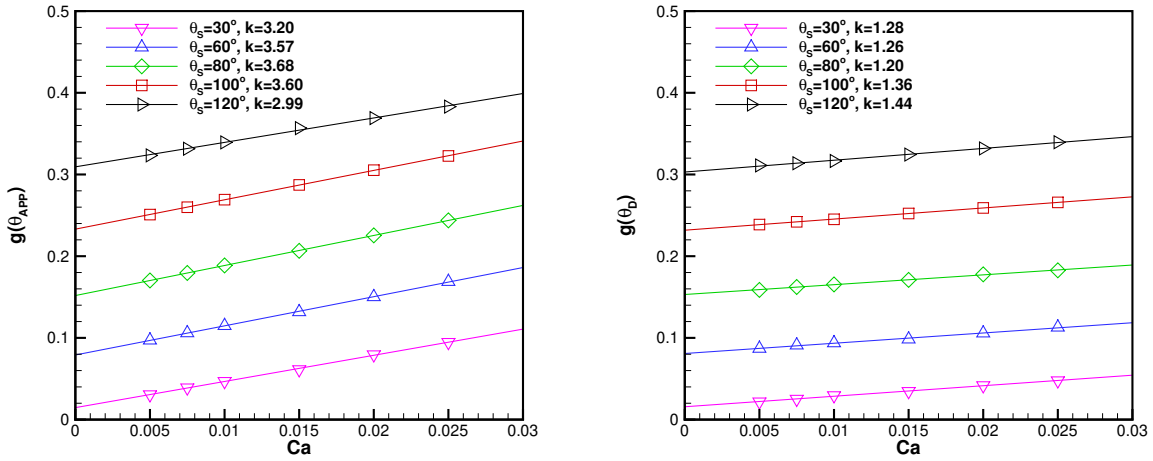
Figure 3.9: Dependence of $g(\theta_{app})$ and $g(\theta_D)$ on Ca for different (β_N, β_{CL}). $\theta_S = 90°$. The k values are the slopes of the linear fits.

θ_S. This is confirmed by Fig. 3.10(a) with acceptable errors: the slopes have an average of 3.4 and a standard deviation of 0.3. The largest deviation from the average is observed for $\theta_S = 120°$, possibly because θ_{app} is too close to 180° and the circle fitting is more prone to numerical errors due to a larger interface deformation. Similar linear behavior is also observed for $g(\theta_D)$, as shown in Fig. 3.10(b), with an average slope of 1.31 and a standard deviation of 0.09.

The results in Figs. 3.9 and 3.10 suggest that both θ_{app} and θ_D can fit in the relation

$$g(\theta) = g(\theta_S) + Ca\, k, \tag{3.37}$$

where k is dependent on both β_N and β_{CL} but independent of θ_S. Comparing with (3.34), we can see that β_N and β_{CL} work together to determine the effective slip length L_s that controls the dynamics of θ_{app}. For example, in Fig. 3.9(a), the slope $k = 2.91$ for $(\beta_N, \beta_{CL}) = (50, 1)$ correspond to $\frac{L_s}{W} = 0.055$, and the slope $k = 5.21$ for $(\beta_N, \beta_{CL}) = (100, 8)$ corresponds to



(a) (b)

Figure 3.10: Dependence of $g(\theta_{app})$ and $g(\theta_D)$ on Ca for different θ_S. $\beta_N = 100$, $\beta_{CL} = 1$.

$\frac{L_s}{W} = 0.0055$.

Since both θ_{app} and θ_D satisfy (3.37), it follows that

$$g(\theta_{app}) = g(\theta_D) + Ca\,\tilde{k}, \tag{3.38}$$

for some constant $\tilde{k}$, which is easily confirmed by numerical results. It is, however, surprising to note that this $\tilde{k}$ is independent of β_{CL}, as shown in Fig. 3.11. That is, β_{CL} has nothing to do with viscous bending. The data points for $\beta_{CL} = 1$ indicate that $\tilde{k} \approx 0.52 \ln\left(\frac{W}{l_s}\right)$, where $l_s = \frac{\mu}{\beta_N}$. We do not have any good explanation on the prefactor for now and will leave further investigation for future work.

In summary, β_N and β_{CL} both affect the deviation of θ_D from θ_S, while only β_N controls the viscous effect that bends the interface from θ_D at the wall to θ_{app} at the macroscopic scale. It is the cooperation of β_N and β_{CL} that determines the effective slip length L_s.

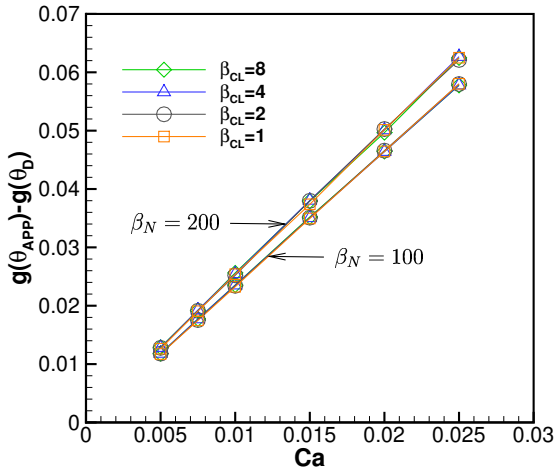


Figure 3.11: $g(\theta_{app}) - g(\theta_D)$ as a function of Ca. $\theta_S = 90°$.

3.4.4 Drop spreading and computational strategy

In this section, we simulate two cases of drop spreading with different initial contact angles. Inertia is again neglected. The computational setup is illustrated in Fig. 3.12. The flow is axisymmetric and we only compute the right half of the median plane.

Comparison with the Cox theory

In the first test case, we consider the spreading of an initially a hemisphere drop with radius $R_0 = 0.5$. We take $\theta_S = 60°$. The spreading radius will be directly compared with the theoretical results in [146].

If we assume the drop to be a spherical cap, which is a reasonable approximation for $Ca \ll 1$, the spreading radius a can be written as a function of θ_{app} based on volume conservation:

$$a = \left(\frac{3V_d}{\pi}\right)^{1/3} \frac{\sin\theta_{app}}{(2 - 3\cos\theta_{app} + \cos^3\theta_{app})^{1/3}}, \tag{3.39}$$

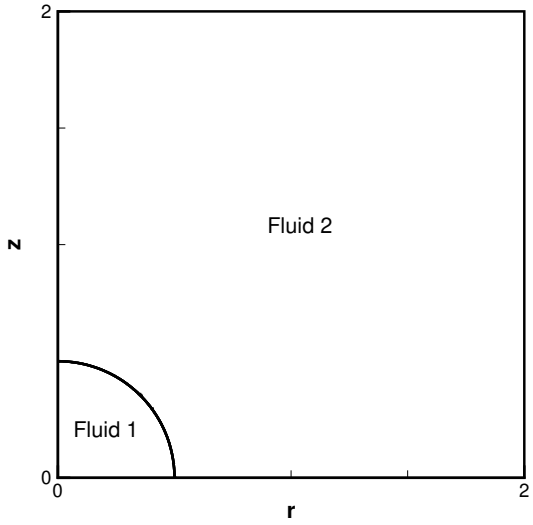


Figure 3.12: Computational setup for drop spreading simulations. The computation domain is a square of $4R_0 \times 4R_0$ with the wall located at $z = 0$.

where V_d is the volume of the drop. The final spreading radius a can be predicted based on the static contact angle θ_S and the initial contact angle θ_0 [146]:

$$\frac{a_f}{a_0} = \left[\frac{(2 - 3\cos\theta_0 + \cos^3\theta_0)\sin^3\theta_S}{(2 - 3\cos\theta_S + \cos^3\theta_S)\sin^3\theta_0}\right]^{1/3}, \tag{3.40}$$

where a_0 is the initial spreading radius. For the hemispherical drop considered here, we have $a_0 = R_0$ and $\theta_0 = 90°$ and the equation above gives $\frac{a_f}{R_0} = 1.276186$ for $\theta_S = 60°$, which is confirmed by our numerical results in Figs. 3.14.

Mesh convergence at transient times

The mesh convergence results for this transient problem are shown in Fig 3.13. The different spreading curves converge as the mesh refines. The final spreading radii are given in Table 3.2, which indicates a first order convergence. This is expected due to the smooth Dirac

delta function δ_ϵ in the formulation.

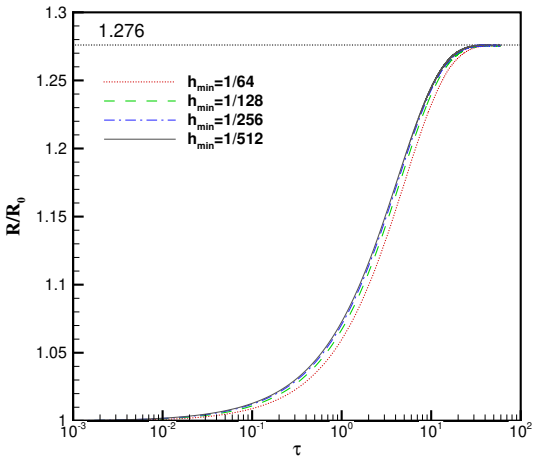


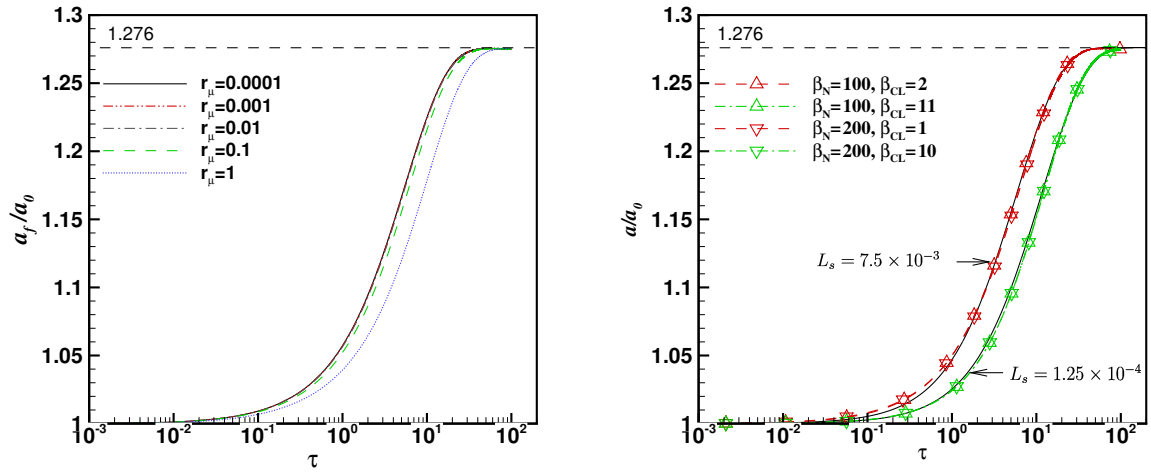
Figure 3.13: Study of mesh dependence with $\epsilon = 1.5h_{\min}, \beta_N = 100$.

$h_{\min}$	a_f/R_0	Relative error	Convergence rate
1/64	1.275312	6.85E-04	-
1/128	1.275458	5.70E-04	0.26
1/256	1.275830	2.79E-04	1.03
1/512	1.276010	1.38E-04	1.02

Table 3.2: Maximal spreading radius at different levels of mesh refinement. $\theta_S = 60°, r_\mu = 0.01, \beta_N = 100, \beta_{CL} = 1, \epsilon = 1.5h_{\min}$.

It was reported in [146] that the influence of gas viscosity is negligible for $r_\mu \leq 0.01$. We verify this by computing spreading with $r_\mu = 1, 0.1, \cdots, 0.0001$, and the numerical results are given in Fig. 3.14(a). It is obvious that the spreading curves with $r_\mu \leq 0.01$ are indistinguishable. In the following simulations will simply use $r_\mu = 0.01$ for gas-liquid systems.

Based on the Cox theory, Wörner *et al.* [146] derived analytical equations to predict the $a \sim t$ curves of spherical-cap shaped drops. The effective slip L_s of the moving contact line can thus be determined by matching the theoretical and the actual spreading curves. The spreading curves with different (β_N, β_{CL}) values are given in Fig. 3.14(b). First of all,



(a) Effect of viscosity ratio.

(b) Comparison with theory.

Figure 3.14: Spreading of a hemispherical drop with $\theta_S = 60°$. In (a), we vary the viscosity ratio r_μ while keeping $\beta_N = 100$ and $\beta_{CL} = 1$ fixed. In (b), we vary β_N and β_{CL} while keeping $r_\mu = 0.01$ fixed. The solid lines in (b) are theoretical curves based on the Cox theory with L_s labeled in the plot and $L = a_0$. The finest mesh size is set to $h_{\min} = \frac{1}{128}$ and $\frac{1}{256}$ for $\beta_N = 100$ and 200, respectively.

the theoretical curves match the numerical ones very well if a proper L_s is chosen. Thus our level-set method with the slip condition (3.8) agrees with the Cox theory. Meanwhile, the same L_s can be reproduced by multiple choices of (β_N, β_{CL}). This indicates that β_{CL} can be used to compensate β_N in controlling the effective slip L_s, and further motivates the following computational strategy.

Computational strategy

We propose the following computational strategy for mesh-independent and predictive numerical simulations of moving contact line problems. First, choose an affordable mesh size $h_{\min}$. Then, determine interface thickness $\epsilon \gtrsim h_{\min}$ and numerical slip $l_s \gtrsim \epsilon$ such that the results are mesh-independent. The friction coefficient β_N can be computed from l_s. Finally, adjust β_{CL} to produce the desired L_s. A larger β_{CL} is required to produce a smaller L_s.

This strategy is similar to that proposed by Yue and Feng for the phase-field method [54]. Unfortunately, we have not obtained a qualitative formula similar to that in the phase-field method to guide the choice of β_{CL} yet. This will be an important part of our future work. For now, we leave β_{CL} as a fitting parameter that needs to be calibrated based on experiments or other results, similar to that in [48]. Once calibrated, our method will be able to predict the correct contact line dynamics for a wide spectrum of contact line velocities and flow geometries, as long as the two fluids and the solid surface remain the same.

Comparison with experiment

In this test case, we compare our results with the experimental data on drop spreading by Zosel [2]. This also serves as an example on the usage of our computational strategy.

In the experiments, solutions of polyisobutylene in decaline with a range of concentrations were tested. It was observed that all experimental data fall onto a master curve if the dimensionless spreading radius $\frac{a}{R_0}$ is plotted against the dimensionless spreading time $\frac{t\sigma}{\mu_1 R_0}$. We choose the data points for pure polyisobutylene, which have the widest coverage on the whole master curve. Although θ_S was reported to be about 58 to 60° in the experiments, the final spreading radius, which is about $\frac{a_f}{R_0} = 1.69$, suggests a smaller angle $\theta_S = 54°$ based on (3.39). We thus take $\theta_S = 54°$ in our simulations. The drop is initially spherical with a radius $R_0 = 0.5$ and a center at $(0, 0.48)$, such that the inner rim of the narrow-band interface just touches the wall. We set the viscosities to $\mu_1 = 1$ and $\mu_2 = 0.01$.

Following the proposed computational strategy, we first set $h_{\min} = \frac{1}{128}$ and $\epsilon = 1.5h_{\min}$. Then we pick $\beta_N = 100$ such that $l_s = \frac{\mu_1}{\beta} = 0.01$ can be resolved by $h_{\min}$ and ϵ. Finally, we tune β_{CL} to match experimental data. The spreading curves with different β_{CL} are given in Fig. 3.15. The curves with $\beta_{CL} = 0.5$ and $\beta_{CL} = 1$ match the experimental curve the best.

Once we identify the (β_N, β_{CL}) pair for the liquid-gas-solid system, it can be used to predict the contact line dynamics under other flow conditions.

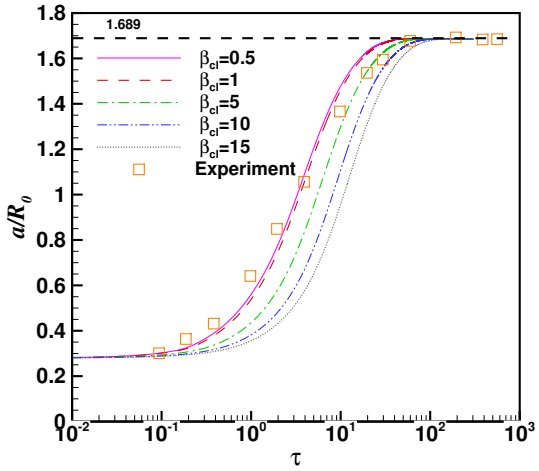


Figure 3.15: Comparison of drop spreading with experiment. The symbols are the experimental data for the spreading of polyisobutylene on polytetrafluoroethylene [2]. $\beta_N = 100$,$h_{\min} = 1/128$, $\epsilon = 1.5h_{\min}$, $\theta_S = 54°$.

It should be noted that our numerical curves cannot match the experimental data exactly: the experimental data demonstrate a lower slope in the semi-log plot. The same trend was also observed in other numerical simulations [147, 148, 149]. This consistent discrepancy is probably because of the constant-coefficient assumptions in the contact line models. Maybe the friction coefficients β_N and β_{CL} (or the slip length) should be functions of the contact line velocity. Further investigation is beyond the scope of this paper.

3.4.5 Pinned drop in Poiseuille flow

We test the capability of our method in pinning contact lines by comparing with the boundary integral method [3]. A cylindrical-cap droplet, with an area of 0.5, is initially placed on a solid surface with contact angle $\theta = \pi/3$, as shown in Fig (3.16). The droplet is sheared by a

pressure-driven flow with contact lines pinned. The capillary number is subcritical such that the drop eventually achieves a steady deformation. The flow is inertialess with $\mu_1 = \mu_2 = 1$ and $\sigma = 1$. We impose a large hysteresis window $[1°, 179°]$ such that the contact lines are pinned on the wall. The computational domain is a rectangle of dimensions $(0, 8) \times (0, 2)$. On the left boundary $x = 0$, we impose the inflow condition $\boldsymbol{u} = [u, 0]^T$ with

$$u = \frac{3}{2}\bar{V}\left(1 - (1 - y)^2\right), \tag{3.41}$$

where $\bar{V}$ is the average velocity in the channel. Following [3], we define the capillary number as $Ca = \frac{\mu_1 E h}{\sigma}$, where $E = 3\bar{V}$ is the shear rate at the wall (noting that the channel half height is 1) and $h = 0.4511$ is the initial height of the drop. We take $\beta_N = 100$, $\beta_{CL} = 100$, $h_{\min} = \frac{1}{64}$, and $\epsilon = 1.5h_{\min}$ in our simulations. To achieve desirable pinning performance, a large enough β_{CL} needs to be used when the contact line is pinned.

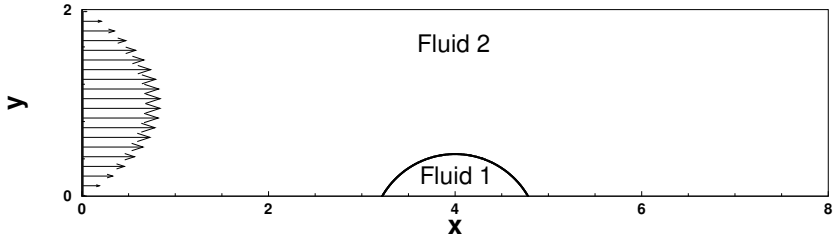


Figure 3.16: Schematic of a drop under shear in plane Poiseuille flow.

The steady-state drop shapes are given in Fig. 3.17. Our results match those of the boundary integral method almost perfectly. Since we pin the contact line through the fluid velocity, it is very difficult to achieving exact pinning due to numerical errors in computing the flow field and advecting the level-set function. Thus the contact line may shift away from its original position. At the leading (left) edge of the drop, as shown in Fig. 3.18(a), the contact line is blown downstream as Ca increases. This trend is probably related to the finite thickness of the numerical interface, which may cause a large error when the interface is almost parallel

to the wall, i.e., when the dynamic contact angle is close to 0° or 180°. This error is however acceptable compared to the computational mesh: the maximum displacement of the contact line is around $\frac{h_{\min}}{4}$ at $Ca = 0.15$. At the trailing (right) edge of the drop, the contact line is nicely pinned because the dynamic contact angle is close to 90°

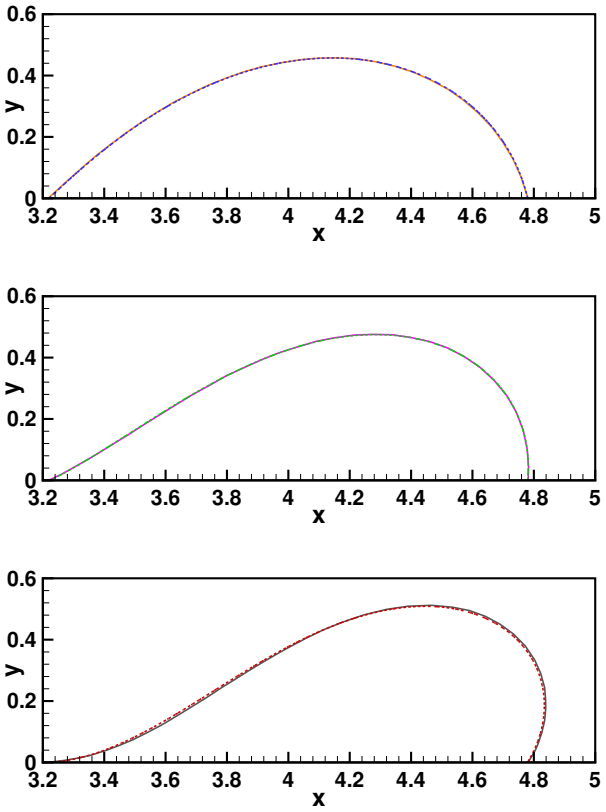
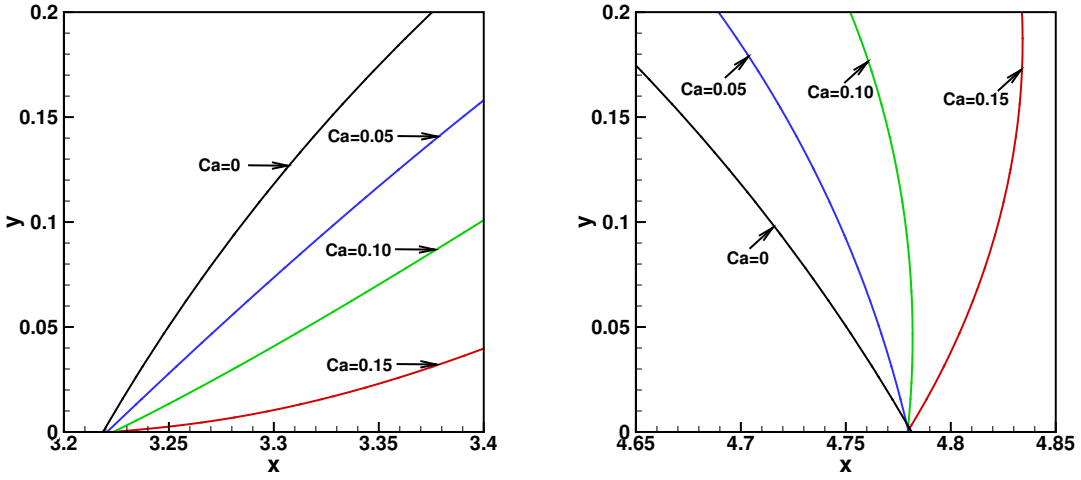


Figure 3.17: Comparison with the boundary-integral results of Schleizer and Bonnecaze [3]. From top to bottom, Ca=0.05, 0.10, and 0.15. The solid lines represent the boundary-integral results, while the dashed lines indicate our level-set results.

3.4.6 Advancing and receding contact lines in a channel

In this test case, we consider two immiscible fluids separated by two interfaces in a plane Poiseuille flow. Due to symmetry, we only compute the lower half of the channel, which is a rectangular domain $[0, 8] \times [0, 1]$, as shown in Fig. 3.19. The channel is long enough such



(a) Leading edge (b) Trailing edge

Figure 3.18: Zoomed views of the steady-state interface in the vicinity of the contact lines. $Ca = 0$ denotes the undeformed interface.

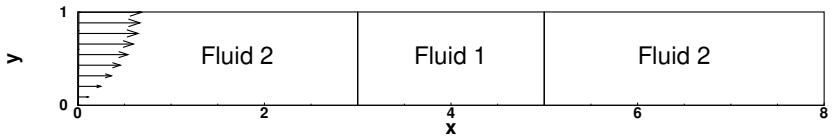


Figure 3.19: Schematic of advancing and receding contact lines in a channel

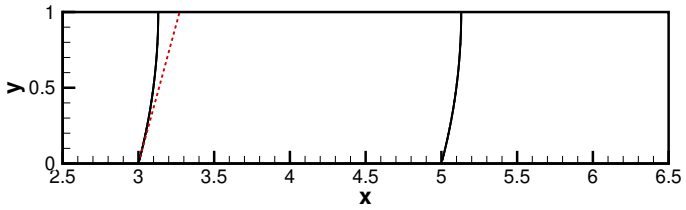
that the inflow and outflow conditions are not affected by interface deformation. The two interfaces are initially vertical and located at $x = 3$ and 5, respectively. Under flow, the contact line to the left eventually recedes with respect to fluid 1 while the other contact line advances; we henceforth refer to these two contact lines as receding and advancing contact lines, respectively, even when they are pinned. The system is inertialess and the two components have the same viscosity $\mu = 1$. The surface tension is set to $\sigma = 1$. The prescribed advancing and receding contact angles are $\theta_R = 75°$ and $\theta_A = 135°$, respectively. We take the following friction coefficients: $\beta_N = 100$, $\beta_{CL} = 1$ for moving contact lines, and $\beta_{CL} = 100$ for pinned contact lines. On the left boundary $x = 0$, we impose the same parabolic velocity profile as (3.41). We take $\bar{V} = 0.01$ such that $Ca = \frac{\mu \bar{V}}{\sigma} = 0.01$ is small

enough and the deformed interfaces remain almost circular. For convenience we define a normalized time $t^* = \frac{t\bar{V}}{H}$, where $H = 1$ is the half height of the channel.

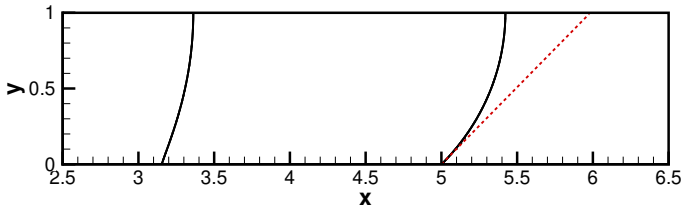
Typical interface shapes are given in Fig. 3.20 and the zooms at the contact line are shown in Fig. 3.21. The receding contact line remains pinned until about $t^* = 0.0876$ and the advancing one remains pinned until about $t^* = 0.2906$. For a circular interface with a pinned contact line, we can find the following relation between dynamic angle θ_D and normalized time t^* based on mass conservation [133]: $t^* = \frac{1}{2}\frac{H}{\bar{V}}\left(\frac{\delta}{\sin^2\delta} - \cot\delta\right)$, where $\delta = |\frac{\pi}{2} - \theta_D|$. This gives $t^* = 0.0881$ for the receding contact line to achieve $\theta_D = 75^\circ$ and $t^* = 0.2854$ for the advancing contact line to achieve $\theta_D = 135^\circ$. Our numerical depinning times agree with these theoretical predictions very well.

3.4.7 2D sliding drop

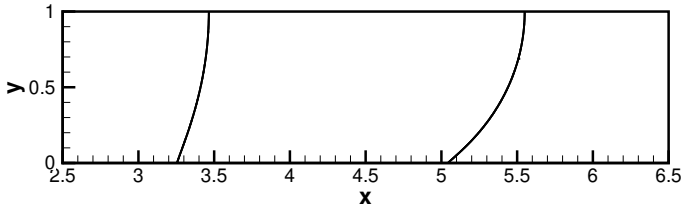
This test case is adapted from [30]. We consider the deformation of a drop on a wall which is slowly inclined, as shown in Fig. 3.22. The drop is initially semicircular with radius $R_0 = 0.5$, and the computational domain is a rectangle of 5×1. The finest mesh is set to $h_{\min} = \frac{1}{128}$. We choose the following fluid properties: $\rho_1 = 1$, $\rho_2 = 0.01$, $\mu_1 = 1$, $\mu_2 = 0.1$, and $\sigma = 1$. Since $\rho_2 \ll \rho_1$, we define the Bond number as $Bo = \rho_1 g R_0^2/\sigma$, according to which the magnitude of gravitational acceleration g is adjusted to achieve different Bo. The wall is initially horizontal and slowly inclined until the drop starts to slide. Each inclination angle α is maintained for a period of the greater of the inertia-capillary time $\sqrt{\rho_1 R_0^3/\sigma}$ and the visco-capillary time $\mu_1 R/\sigma$, which is 0.5 for our chosen parameters, to allow enough time for the drop to deform. The increment of slop angle varies depending on whether the inclination angle is near critical. In the simulations, instead of rotating the computational domain, we rotate the gravitational acceleration $\boldsymbol{g} = -g(\sin\alpha, \cos\alpha)$. For the contact line,



(a) $t^* = 0.0876$



(b) $t^* = 0.2906$

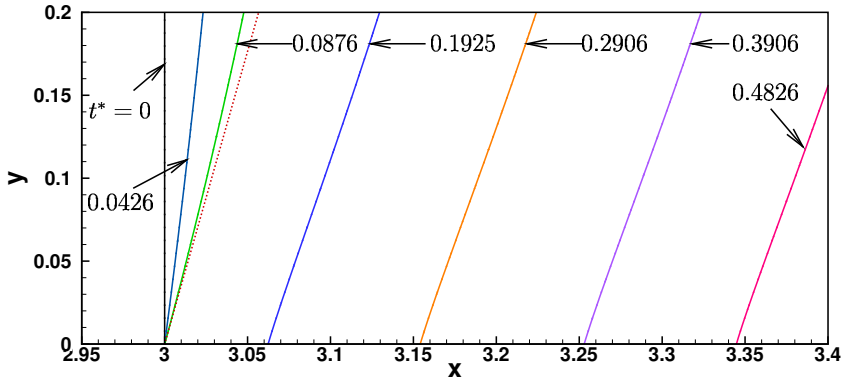


(c) $t^* = 0.3926$

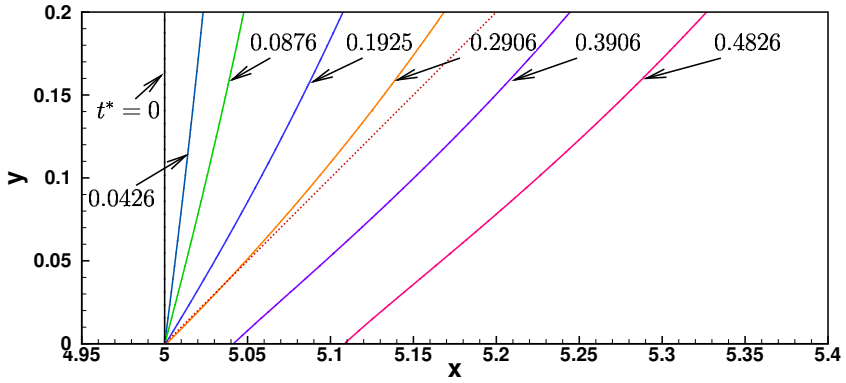
Figure 3.20: Shapes of advancing and receding interfaces in a channel. The red dotted lines correspond to the receding angle $\theta_R = 75^\circ$ and the advancing angle $\theta_A = 135^\circ$, respectively.

we take $\beta_N = 100$, $\beta_{CL} = 1$ at the moving contact line, and $\beta_{CL} = 100$ at the pinned contact line.

Theoretically, based on a force balance along the wall, one can derive the critical inclination



(a) Receding interface



(b) Advancing interface

Figure 3.21: Evolution of the interfaces in the vicinity of the contact lines.

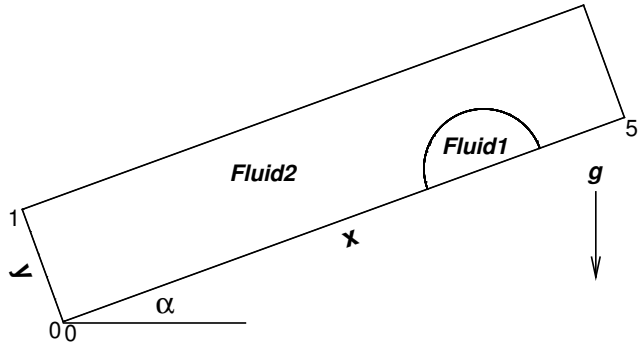


Figure 3.22: Schematic of a drop on an inclined wall.

angle α_c when the drop starts to slide [150, 151]:

$$\frac{1}{2}\pi R^2 \rho_1 g \sin\alpha_c = \sigma(\cos\theta_R - \cos\theta_A), \tag{3.42}$$

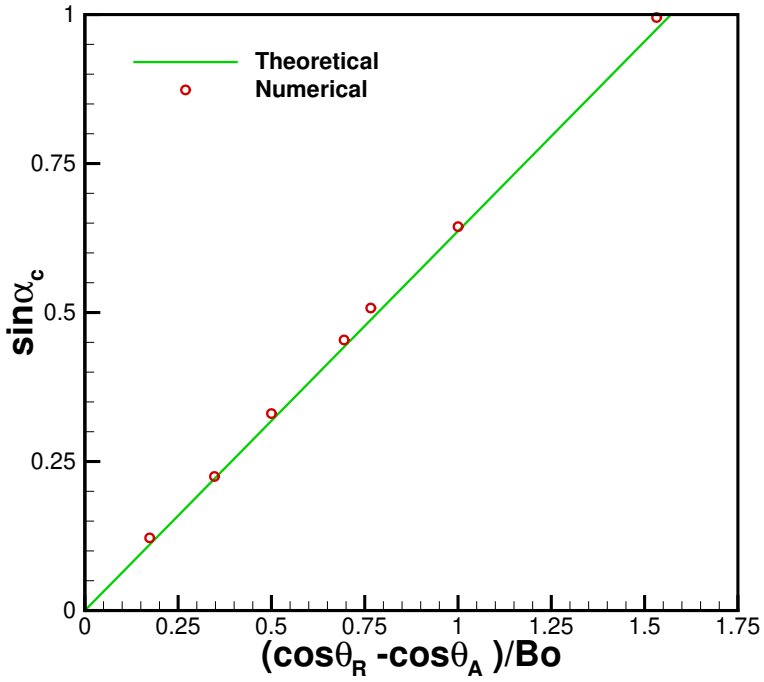


Figure 3.23: Critical inclination angle α_c versus $(\cos\theta_R - \cos\theta_A)/\mathrm{Bo}$.

which can be rewritten as

$$\sin\alpha_c = \frac{2}{\pi \mathrm{Bo}} \left(\cos\theta_R - \cos\theta_A\right). \tag{3.43}$$

By choosing different hysteresis angles and Bond numbers, we can compute the corresponding critical inclination angles α_c and compare against the theoretical relation (3.43). Here, we test Bo = 0.5, 1.0 and 2.0 with different hysteresis angles $(\theta_R, \theta_A) = (80°, 100°), (60°, 120°)$, and $(40°, 140°)$, which are the same as in [30]. The critical inclination angles are summarized in Fig. 3.23, which indicates a very good agreement with the theoretical relation (3.43). Quantitatively, the agreement is better than that obtained by the volume-of-fluid method in [30]. In terms of computational mesh, our finest mesh size is slightly smaller, but the total number of cells, typically around 2000, is much less than the 500×100 uniform mesh in [30].

The critical drop shapes are displayed in Fig. 3.24. For small Bond numbers, when the hysteresis is sufficiently large, the drop is pinned on the wall even when the inclination angle achieves 90°. For example, at $Bo = 0.5$, the drops are pinned on the wall with identical shapes for $(\theta_R, \theta_A) = (60°, 120°)$ and $(40°, 140°)$. In these two cases, α_c does not exist.

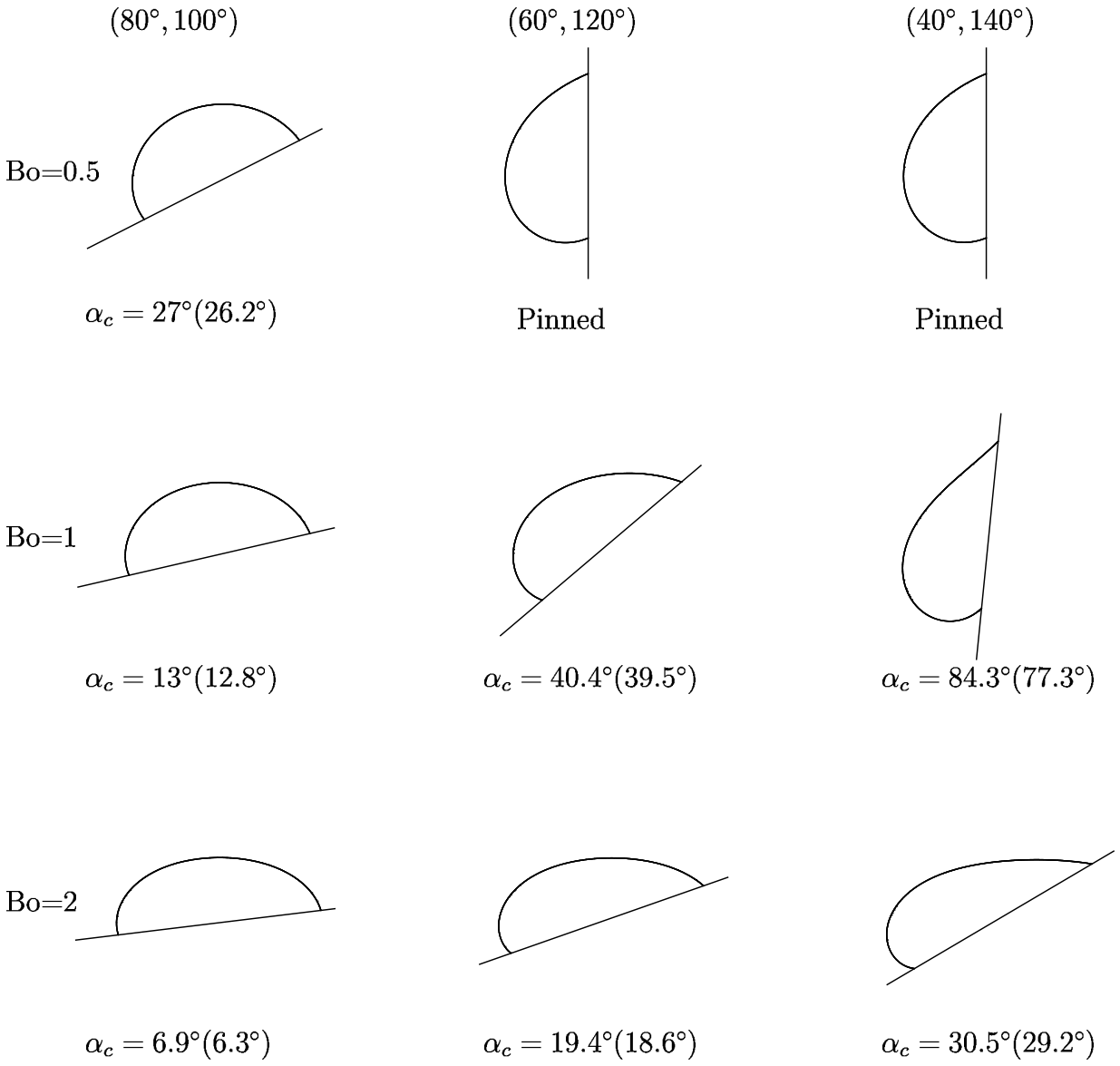
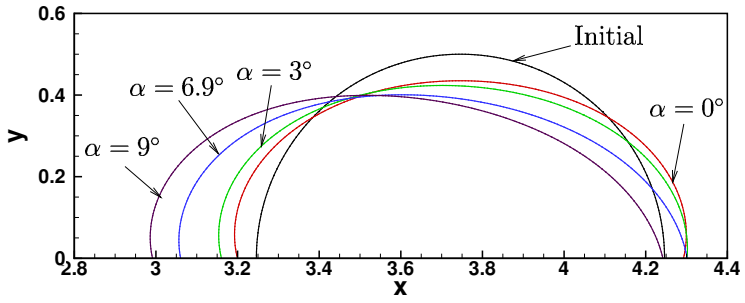
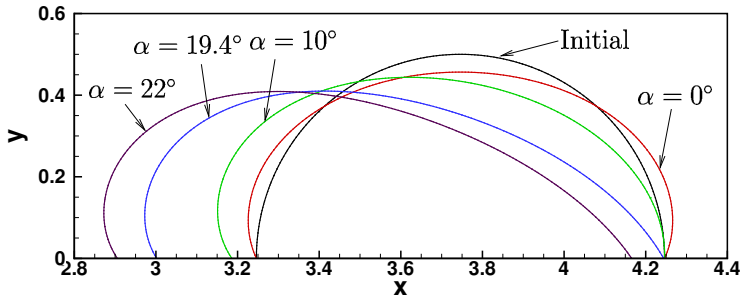


Figure 3.24: Drop shapes at critical inclination angles. Bo is fixed in each row and (θ_R, θ_A) is fixed in each column. The theoretical values of α_c are given in the parentheses.

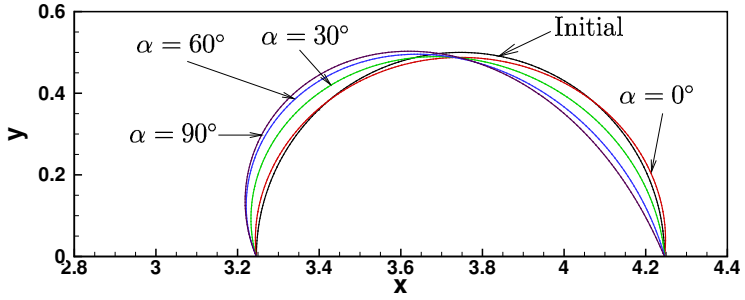
Drop shape evolutions for selected parameters are given in Fig. 3.25. Each curve denotes the



(a) $Bo = 2$, $\theta_A = 100°$, $\theta_R = 80°$.



(b) $Bo = 2$, $\theta_A = 120°$, $\theta_R = 60°$.



(c) $Bo = 0.5$, $\theta_A = 120°$, $\theta_R = 60°$.

Figure 3.25: Evolutions of drop shapes.

interface obtained with the inclination angle α being fixed for a time span of 0.5. It can be viewed approximately as the steady drop shape for the given α. In (a), gravity is dominant and θ_A is close to the initial contact angle of 90°. At $\alpha = 0°$, the drop spreads due to gravity.

As α increases, the dynamic contact angle at the advancing contact line (left) increases and the one at the receding contact line (right) decreases. The advancing contact line moves first with the receding one pinned at $\alpha = 3°$. At $\alpha = 6.9°$, the receding contact line starts to move at a very low speed and this angle is recorded as the critical inclination angle. In (b), the hysteresis window is increased to $(60°, 120°)$. θ_A is big enough to inhibit the initial drop spreading at $\alpha = 0°$: the drop flattens under gravity, but the contact lines remain pinned. The later dynamics is similar to that in (a). At a sufficiently small Bo, gravity is insufficient to overcome the contact angle hysteresis, as shown in (c). As a result, the drop deforms with contact lines pinned.

3.5 Summary

In this chapter, we have developed a level-set method in the finite-element framework to simulate interfacial flows with moving contact lines. The level-set function is reinitialized by the method in Chap. 2. The contact line singularity is relaxed by Ren and E's slip condition that has two friction coefficients: β_N and β_{CL}. A series of simulations have been conducted to study systematically the relations among β_N, β_{CL}, and the Cox theory. It is shown that only β_N affects the mesh convergence, and different combinations of β_N and β_{CL} can recover the slip length in the Cox theory. Therefore, a computational strategy is proposed to reproduce the dynamics of moving contact lines that is in good agreement with the Cox theory. With a simple modification, the contact angle hysteresis is incorporated into our method. Numerical simulations in 2D have illustrated the accuracy and capability of our method.

Chapter 4

Three-dimensional simulations of moving contact lines

In this chapter, we extend the level-set reinitialization to 3D moving contact line problems. The extension is not trivial, because the contact angle may vary along the contact line. Therefore, different from the simple treatment in 2D, an extension equation needs to be solved to obtain the boundary condition on the inflow boundary $\partial\Omega_{in}$ (see (2.64)) in 3D.

4.1 Introduction

Different approaches have been proposed to supply boundary conditions on the wall boundary in 3D. The first approach is to extend contact angles on the wall boundary, then construct necessary boundary conditions for ϕ based on the extended contact angles [152, 153]. However, the extension to unstructured mesh and curved boundary is limited by the use of ghost cells. The second approach is to solve a reinitialization problem on the wall boundary [106]. However, the extension to 3D is unclear. Another approach is to solve a relaxation equation in the first layer of cells along the inflow boundary such that $\frac{\nabla\phi}{|\nabla\phi|}\cdot\boldsymbol{n}_w$ is fixed [154]. But this approach is dependent on the quality of ϕ_0 away from the interface. In this work, we propose to propagate $\boldsymbol{n}_w\cdot\nabla\phi$ by solving an extension equation, then reconstruct the boundary condition for $\nabla\phi$.

This chapter is organized as follows. We describe the extension equation and the corresponding finite-element discretization in Section 4.2. Then, we show the numerical results in 3D in Section 4.3, including reinitialization of the ellipsoidal interfaces and evolution of a drop on an inclined wall.

4.2 Boundary conditions for $\nabla\phi$

Although our reinitialization method provides an easy way to impose boundary conditions in 2D, it is not that straightforward in 3D. We can still use Dirichlet conditions, but an extension equation has to be solved to supply the boundary condition on $\partial\Omega_{in}$. In light of the method in [152], we propose to solve the following equation to extend the scalar field $d_w = \boldsymbol{n}_w \cdot \nabla\phi$ from the contact line to the entire wall boundary:

$$\frac{\partial d_w}{\partial \tau} + S(\phi_0)\left(\nabla_s \phi \cdot \nabla_s d_w\right) = 0 \tag{4.1}$$

with initial condition $d_w(\boldsymbol{x}, 0) = \nabla\phi_0 \cdot \boldsymbol{n}_w$. Here $\nabla_s = \nabla - \boldsymbol{n}_w \cdot \nabla$ is the surface gradient. This equation is solved concurrently with the HJ equation (2.1). At each pseudo time step, we reassemble $\nabla\phi$ on the solid wall by $\frac{d_w \boldsymbol{n}_w + \nabla_s \phi}{|d_w \boldsymbol{n}_w + \nabla_s \phi|}$.

We rewrite the extension equation (4.1) into

$$\frac{\partial d_w}{\partial \tau} + \nabla_s \cdot (\mathbf{v}_w d_w) - d_w \nabla_s \cdot \mathbf{v}_w = 0 \tag{4.2}$$

where $\mathbf{v}_w = S(\phi_0)\nabla\phi$, and we have used the formulas $\nabla\phi \cdot \nabla_s d = \nabla_s \phi \cdot \nabla_s d$ and $\nabla_s \cdot (\mathbf{v}_w d_w) = d_w \nabla_s \cdot \mathbf{v}_w + \nabla_s d_w \cdot \mathbf{v}_w$. Since the boundary condition is only imposed on the inflow boundary

$\partial\Omega_{in}$, we define the discontinuous finite-element space

$$V_w^N = \{w : w \in P^N(F), \forall F \in \mathcal{T}_h \cap \partial\Omega_{in}\}. \tag{4.3}$$

Multiplying (4.2) by the test function $\tilde{w} \in V_w^N$ and integrating by parts, we obtain the following weak form

$$\left\langle \frac{\partial d_w}{\partial \tau}, \tilde{w} \right\rangle + \left\langle \hat{H}_w(d_w^-, d_w^+), \tilde{w} \right\rangle_{\partial F} - \langle \mathbf{v}_w d_w, \nabla_s \tilde{w} \rangle - \langle d_w \nabla_s \cdot \mathbf{v}, \tilde{w} \rangle = 0, \tag{4.4}$$

where $\langle \cdot, \cdot \rangle$ denotes the integral on the face element $F \in \mathcal{T}_h \cap \partial\Omega_{in}$ on the wall boundary, and $\langle \cdot, \cdot \rangle_{\partial F}$ denotes the integral over the edges of element F. Here $\hat{H}_W(d_w^-, d_w^+)$ is the numerical flux that approximates $\mathbf{v} \cdot \boldsymbol{n}_e d_w$, d_w^- and d_w^+ are the inside and outside values of d_w on the element boundary ∂F, and $\boldsymbol{n}_e$ is the outward pointing unit normal to ∂F. We use the local Lax-Friedrichs flux

$$\hat{H}_w(d_w^-, d_w^+) = \mathbf{v}_w \cdot \boldsymbol{n}_e \frac{d_w^- + d_w^+}{2} - \frac{\alpha_w}{2}\left(d_w^+ - d_w^-\right) \tag{4.5}$$

where $\alpha_w = \max\left(|\mathbf{v}_w \cdot \boldsymbol{n}_e|\right)$ and the maximum is taken over the relevant element edge. The time integration uses the third-order TVD RK, which is the same as that used for the HJ equation in Chap. 2.

At each pseudo time step, we first compute d_w on the inflow boundary $\partial\Omega_{in}$, where we need to impose boundary condition. We then solve the HJ equation with the wall boundary condition

$$\nabla\phi = \frac{d_w \boldsymbol{n}_w + \nabla_s \phi}{|d_w \boldsymbol{n}_w + \nabla_s \phi|}. \tag{4.6}$$

4.3 Numerical results and discussions

In this section, we first illustrate the reinitilization of ellipsoidal interfaces to show that our method is accurate and stable. Then we simulate the evolution of a drop on an inclined wall and present some preliminary results. Parameters for level-set reinitialization are identical to those in Sec. 2.4.1.

4.3.1 Reinitialization of ellipsoidal interfaces

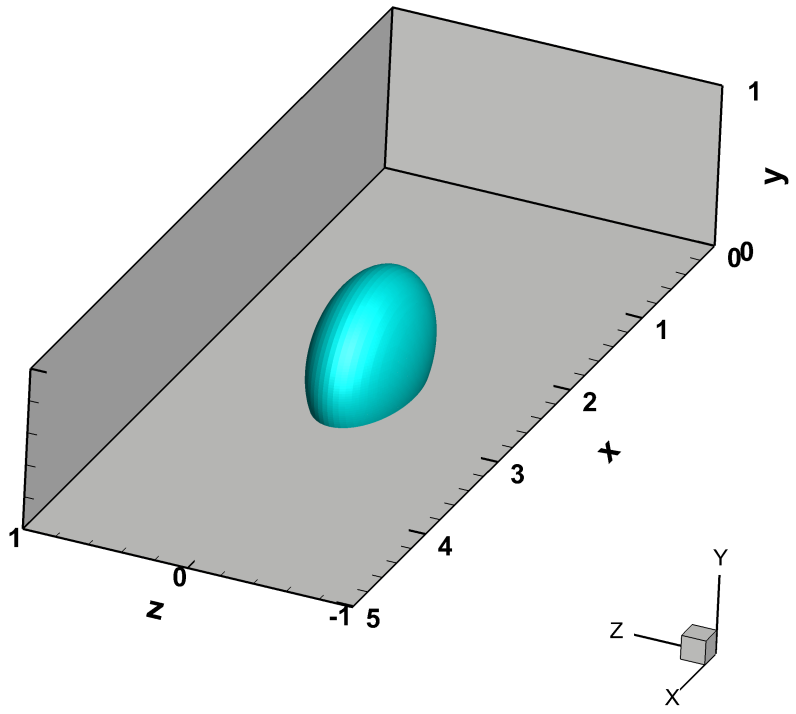
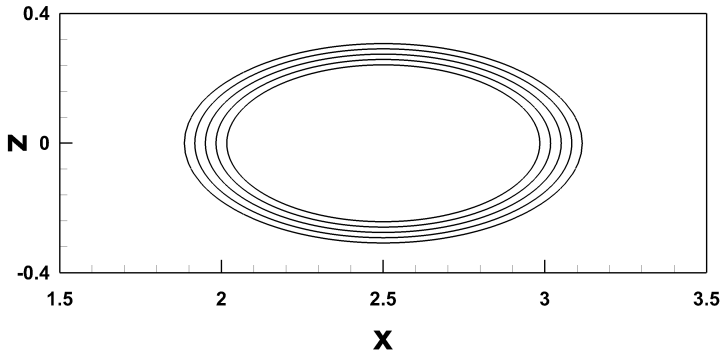
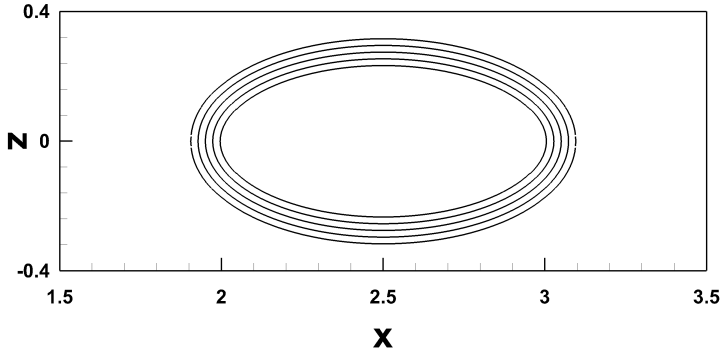


Figure 4.1: Schematic of an ellipsoidal interface cut by the wall boundary.

In this example, we reinitialize ellipsoidal interfaces, and the schematic is given in Fig. 4.1.



(a) ϕ_0.



(b) ϕ_h at $\tau = 15$.

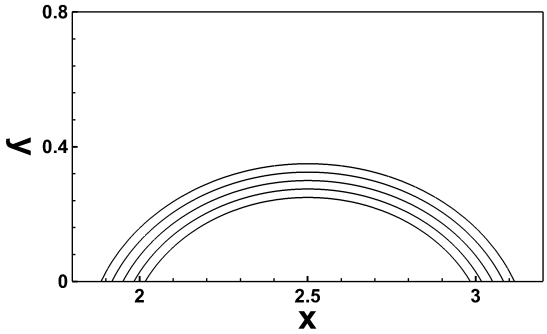
Figure 4.2: ϕ contours on the wall boundary ($y = 0$) with $c_y = -0.2$. Contours run from -0.04 to 0.04 with interval 0.02.

Consider the initial condition

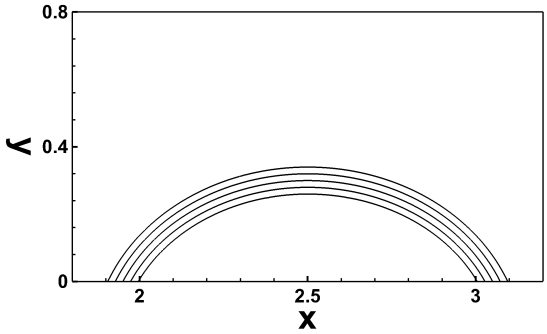
$$\phi_0(x, y, z) = 0.8 * \left(0.5 - \frac{(x - 0.25)^2}{1.2^2} - (y - c_y)^2 - \frac{z^2}{0.6^2} \right), \tag{4.7}$$

in the computational domain $[0, 5] \times [0, 1] \times [-1, 1]$ with $h_{\min} = 1/64$. For other parameters, we take $N = 3$, $\beta_{\max} = 3$, $\lambda = 100$, $\xi = c_\xi h$ with $c_\xi = 0.1$, $\xi_1 = c_{\xi_1} c_\xi$ with $c_{\xi_1} = 0.01$, and $Q = 5$. Here we test two cases by taking $c_y = -0.2$ and 0.2, respectively. Similar to the tests of the elliptic interface in Sec. 2.4.2, we run the reinitializaiton to $\tau = 15$ to test the long-term stability.

When $c_y = -0.2$, the zero level set meets the wall boundary at acute angles, and thus the



(a) ϕ_0.

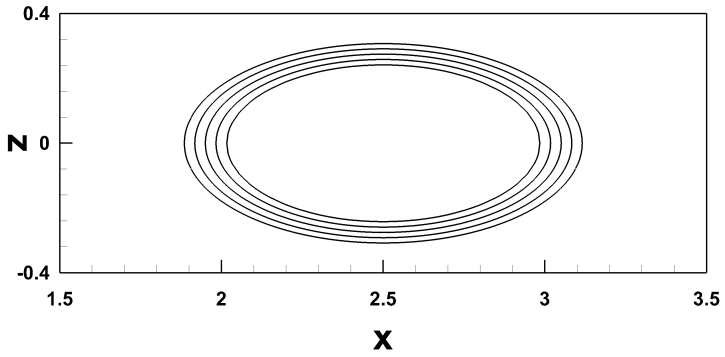


(b) ϕ_h at $\tau = 15$.

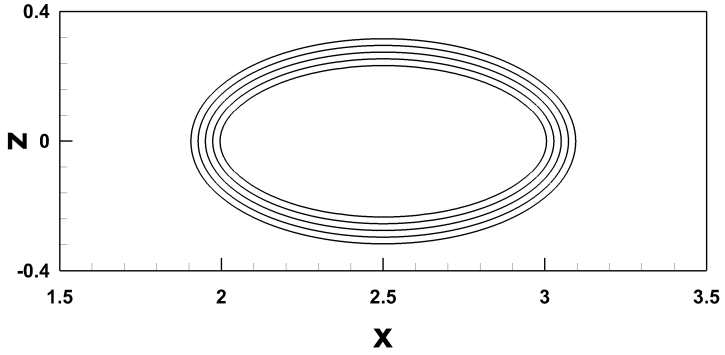
Figure 4.3: ϕ contours at $z = 0$ with $c_y = -0.2$. Contours run from -0.04 to 0.04 with interval 0.02.

inflow boundary is outside the zero level set on the wall boundary. The comparison of the initial condition ϕ_0 and the reinitilized level-set function ϕ_h are given in Figs. 4.2 and 4.3. Note that we only focus on the region near the interface, since an adaptive mesh is used. In interfacial flow calculations, we take $\epsilon = 1.5h_{\min}$; thus ϕ ranges from -0.024 to 0.024 in the interfacial region. We plot 5 contour levels ranging from -0.04 to 0.04 with interval 0.02. From the contour plots in both figures, our results are in good agreement with the signed distance function.

When $c_y = 0.2$, the inflow boundary is inside the zero level set on the wall boundary. In this case, we have to pay extra attention to the stability of the solutions. Except for discontinuities of $\nabla\phi$ as mentioned in Sec. 2.4.2, we will have to take care of the regions



(a) ϕ_0.



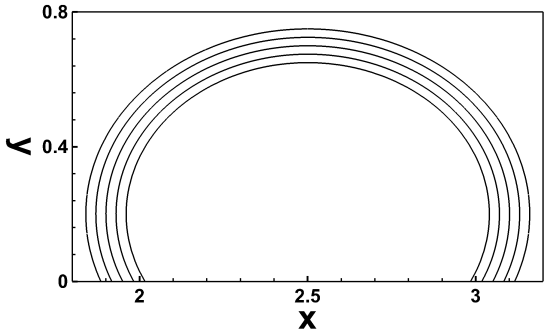
(b) ϕ_h at $\tau = 15$.

Figure 4.4: ϕ contours on the wall boundary ($y = 0$) with $c_y = 0.2$. Contours run from -0.04 to 0.04 with interval 0.02.

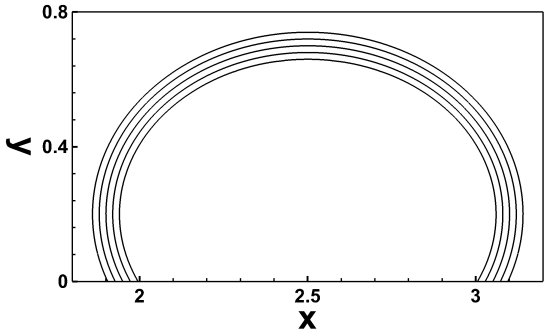
where the characteristics of the extension equation (4.2) intersect as well. Figures. 4.4 and 4.5 show that our method is very stable, and the solution in the troubled region have no effect on the solution near the interface. Therefore, a perfect signed distance function is reproduced.

4.3.2 3D sliding drop

In this section, we show some preliminary results in 3D simulations using the level-set method including the reinitialization of the sliding drop at a late stage of the simulation. For polynomial degrees, we take $N = 3$ for ϕ, $N = 2$ for $\boldsymbol{u}$, and $N = 1$ for p. The time step is set



(a) ϕ_0.



(b) ϕ_h at $\tau = 15$.

Figure 4.5: ϕ contours at $z = 0$ with $c_y = 0.2$. Contours run from -0.04 to 0.04 with interval 0.02.

to $\Delta t = 0.00125$ and level-set function is reinitialized every 5 time steps. We decouple the velocity $\boldsymbol{u}$ and the pressure p by the first order projection method [155], and the discretized systems are both solved by the conjugate gradient method with SSOR preconditioner.

Following [133], we simulate the evolution of a drop on the inclined wall with a fixed inclination angle $\alpha = 60°$. The drop is initially a sphere centered at $(3.75, 0, 0)$ with a radius $R_0 = 0.5$. Due to the symmetry in z-direction, we only compute half of the domain: $[0, 5] \times [0, 1] \times [0, 1]$. The finest mesh is set to $h_{\min} = 1/64$. Correspondingly, we choose $\beta_N = 50$ to increase the numerical slip to $l_s = \mu_1/\beta_N = 0.02$. For contact angles, we take $(\theta_R, \theta_A) = (60°, 120°)$. The parameters of the fluid properties are taken as $\rho_1 = 2, \rho_2 = 0.02, \mu_1 = 1, \mu_2 = 0.01$, and $\sigma = 1$. The gravitational acceleration g is com-

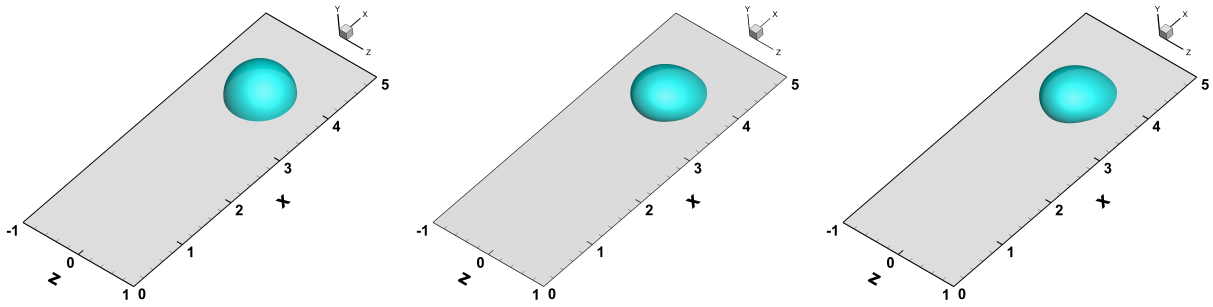
puted based on the Bond number $Bo = \rho_1 g R_0^2/\sigma$, where we have ignored the contribution of ρ_2 since it is much samller than ρ_1. Similar to the 2D case, the critical parameters when the drop starts to slide satisfy [150]

$$\frac{1}{2}\pi R_0^2 \rho_1 V_{drop} g \sin\alpha = w_{drop}\sigma(\cos\theta_R - \cos\theta_A), \tag{4.8}$$

where V_{drop} and w_{drop} are the volume and the width of the drop, respectively. For an initially hemispherical drop, $V_{drop} = w_{drop}$ and the critical Bond number is given by

$$Bo_c = \frac{3\left(\cos(\theta_R) - \cos(\theta_A)\right)}{\pi \sin(\alpha)}. \tag{4.9}$$

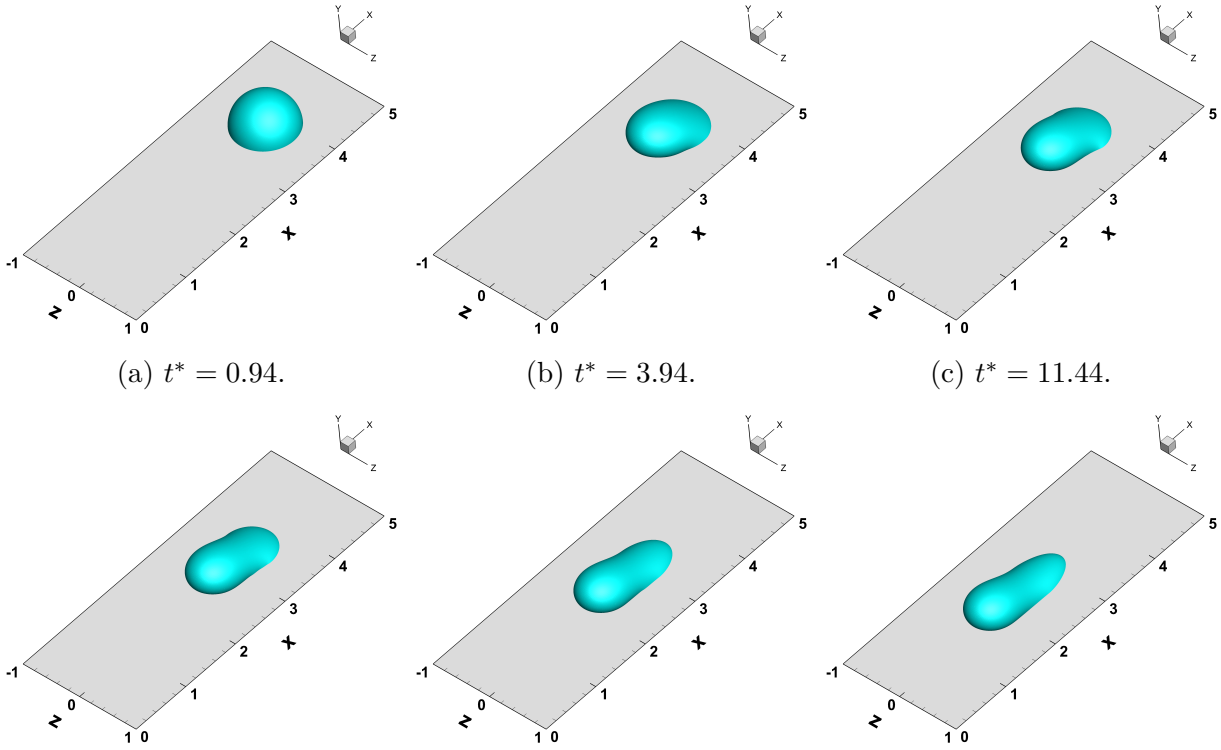
In this test case, $\alpha = 60°$, and thus $Bo_c = 1.1027$. We test two Bond numbers, Bo=1 and 2, by taking $g = 2$ and 4, respectively. In the first case, the Bond number is slightly smaller than the critical value, thus the drop should be pinned on the wall. The Bond number in the second case is almost twice the critical value, and we thus expect the drop to slide on the wall with large deformation.



(a) $t^* = 0$. (b) $t^* = 4.69$. (c) $t^* = 24.69$.

Figure 4.6: Evolution of the drop on the inclined wall with Bo=1.

Sequential snapshots showing the evolution of the drops are displayed in Figs. 4.6 and 4.7. Note that $t^* = t/\sqrt{\rho_1 R_0^3/\sigma}$ is the dimensionless time. Our method successfully captures the pinning and depinning of the contact lines. Side views of the drops given in Fig. 4.8



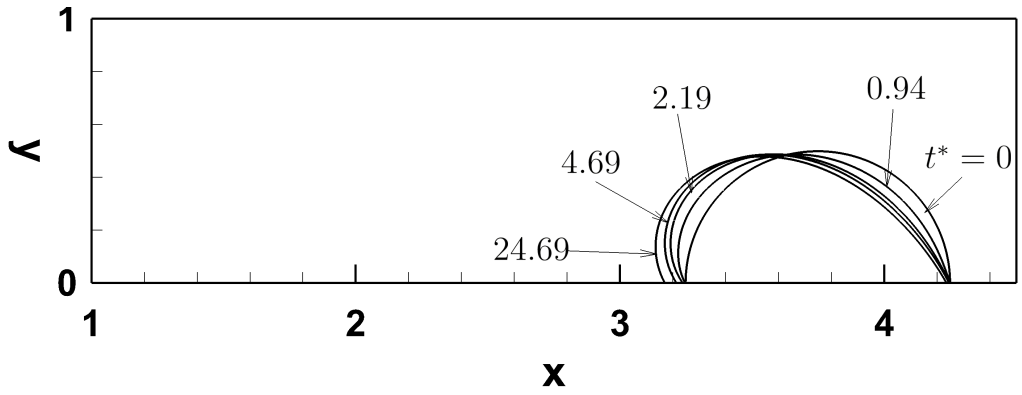
(a) $t^* = 0.94$. (b) $t^* = 3.94$. (c) $t^* = 11.44$.

(d) $t^* = 18.94$. (e) $t^* = 28.94$. (f) $t^* = 39.94$.

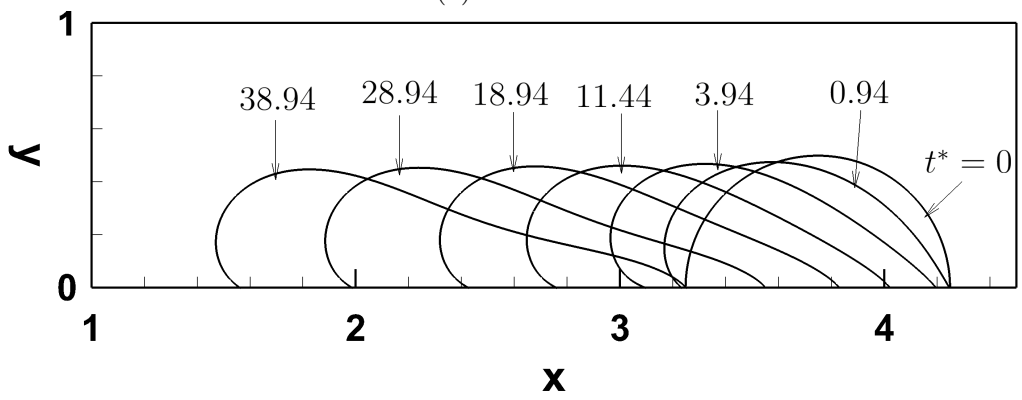
Figure 4.7: Evolution of the drop on the inclined wall with Bo=2.

clearly show the pinning of the contact line in the early stages of the simulations for both cases. Note that, due to numerical errors, the rear portion of the contact line with $Bo = 1$ is not pinned strictly, but the displacement is less than $h_{\min}$. In addition, we illustrate the evolution of the contact lines in Fig.4.9. At later stages of the sliding drop in Fig. 4.9b, we observe that two straight lines, almost parallel to each other, are formed to bridge the advancing and receding ends. This matches the theory in [150, 156] and the simulations of the phase-field method in [133]. Theoretically, these two sides should be parallel to the sliding directions when the sliding motion achieves steady state.

In the end, we examine the reinitialization of the sliding drop with a large advancing contact angle and a small receding contact angle. The contour plot of ϕ_h after reinitialization at $t^* = 38.94$ for $Bo = 2$ is given in Fig. 4.10. Note that the pseudo time for reinitialization



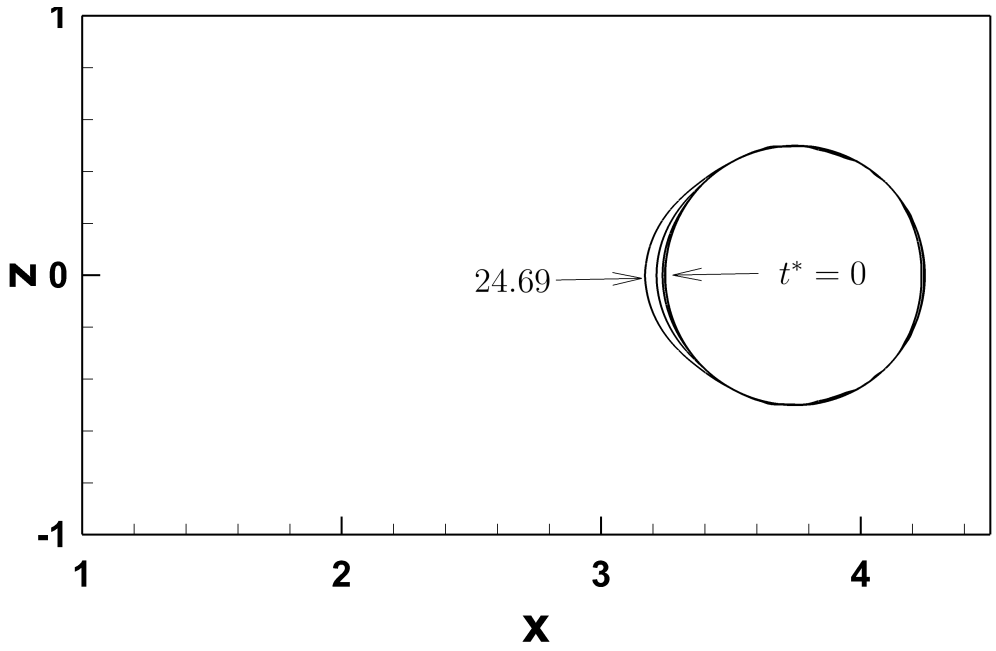
(a) $Bo = 1$.



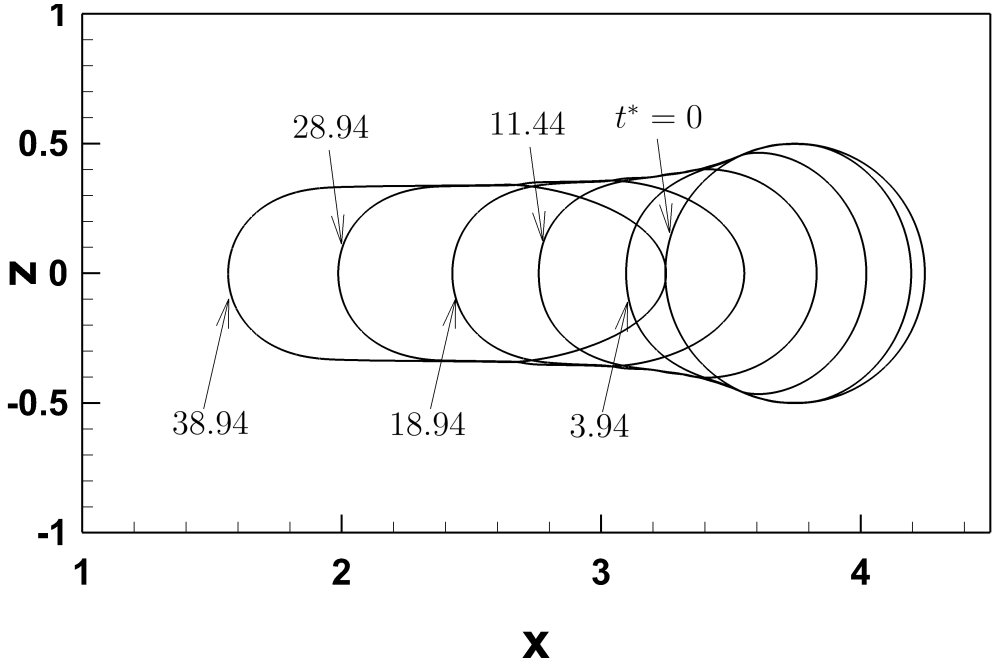
(b) $Bo = 2$.

Figure 4.8: Side views of the drops at different time instants.

is equal to 2ϵ, which is around 0.047 here. Therefore we expect ϕ_h to be a signed distance function within the range of $-0.047 < \phi_h < 0.047$. However, due to the smooth sign function, the numerical characteristic speed is actually smaller than 1. We thus plot the 5 contour levels ranging from -0.04 to 0.04 equidistantly. Away from the wall boundary, the signed distance function is reproduced nicely as illustrated in Fig. 4.10a. In Fig. 4.10b, the uneven distribution of the contour lines on the wall boundary is caused by the variation of the dynamic contact angle θ_D along the contact line. Assume that the interval between contours is D_c, then the distance between each contour line cut by the wall boundary, as shown in Fig. 4.10b, is $D_w = |D_c/\cos(\theta_D)|$. Therefore the contour lines at the advancing and receding ends look sparser.

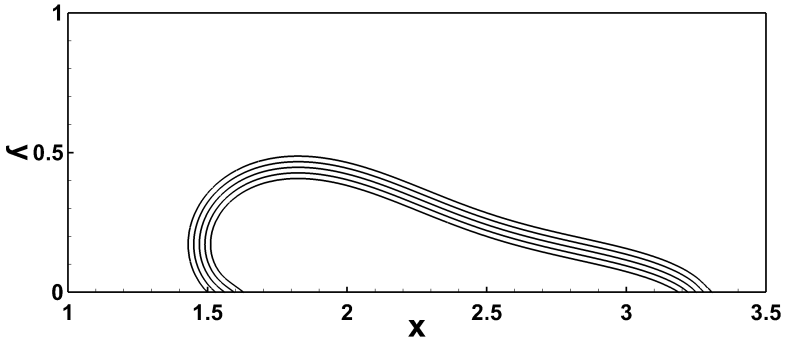


(a) $Bo = 1$.

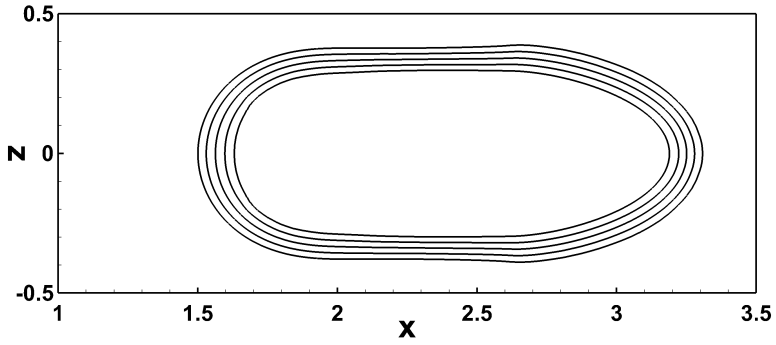


(b) $Bo = 2$.

Figure 4.9: Time evolution of the contact lines.



(a) Side view, $z = 0$.



(b) Top view, $y = 0$.

Figure 4.10: Cross sections of the reinitialized level-set function at $t^* = 38.94$, $Bo = 2$. Contour levels run from 0.04 to 0.04 with interval 0.02.

4.4 Summary

In this chapter, we have extended the level-set method to 3D moving contact line problems. An extension equation is solved on the wall boundary to propagate $\boldsymbol{n}_w \cdot \nabla\phi$ from the contact line to the entire inflow boundary $\partial\Omega_{in}$, where a boundary condition for $\nabla\phi$ is constructed for the level-set reinitialization. Numerical results have shown the stability and potential of our method.

Chapter 5

Conclusions

5.1 Summary

The major contribution of this work is to develop a numerical method that models moving contact lines accurately. In our method, the contact line is evolved by the flow field instead of ad-hoc treatments, and contact angle hysteresis is implemented in a simple and efficient way.

We first develop a robust high-order level-set reinitialization method that preserves the zero level set. The reinitialization method has the following novel features:

(i) For Nth degree piecewise polynomial space, both the weighted local projection and the discontinuous Galerkin method can achieve the optimal Nth order convergence in $\nabla\phi$ for smooth solutions. The convergence order of ϕ is at least $N+1$ in the interface cells and N in the whole domain.

(ii) The penalty flux is necessary to produce smooth solutions, especially when the initial ϕ is highly distorted.

(iii) The numerical method is robust in most cases and handles the discontinuities in $\nabla\phi$ with ease. But in the extreme case with singularities on the interface, the second-derive limiter needs to be applied.

(iv) Since we compute $\nabla\phi$ instead of ϕ in the discontinuous Galerkin method, the boundary condition for the Hamilton-Jacobi equation is easy to set up for contact line problems.

(v) The mass loss is almost zero if the highest interface curvature can be resolved by the computation cell. Mesh refinement is suggested where the interface curvature exceeds $\frac{1}{2h}$. In the moving interface problems, mass conservation can be further improved if we preform reinitialization only once every few time steps.

This method can be easily extended to other types of unstructured meshes and complex geometry.

We then couple the level-set method with the finite-element flow solver for moving contact line problems. The flow equations are solved by a continuous finite element method while the level-set equation is solved by a discontinuous Galerkin method on an adaptive quadrilateral (hexahedral mesh in 3D). After a careful validation against the Cox theory, we come up with a computational strategy for practical contact line simulations. Furthermore, our method can be easily modified to accommodate contact angle hysteresis. The main results can be summarized as follows.

(i) The mesh convergence can be achieved as long as the mesh size $h_{\min}$ is able to resolve the interfacial thickness ϵ and the numerical slip $l_s = \frac{\mu}{\beta_N}$: $h_{\min} \lesssim \epsilon \lesssim l_s$. For a sharp-interface method that does not use the continuum surface force method to apply surface tension, we expect this criterion to reduce to $h_{\min} \lesssim l_s$.

(ii) In additional to β_N, the contact line friction β_{CL} also affects the effective slip length. In particular, a single slip length in the Cox theory can be reproduced by different combinations of (β_N, β_{CL}). In practical computations, we suggest to prescribe β_N based on the mesh convergence requirement and then use β_{CL} as the only fitting parameter to achieve desired slip. By using this strategy, we have obtained a reasonable agreement

with the drop spreading experiment. Since our method does not rely on external models to impose the contact angle condition, it is easy to implement numerically.

(iii) In our method, the dynamic contact angle can be readily obtained from the level-set function, based on which we can determine whether the contact line is pinned, advances, or recedes. Meanwhile, the weak form for the pinned contact line only differs a little from that for the moving contact line. All these properties make it easy to incorporate contact angle hysteresis. More importantly, our method demonstrates very good accuracy in capturing hysteresis.

Lastly, we propose a new method to a construct boundary condition for $\nabla\phi$ on the wall boundary in 3D. The construction requires solving an extension equation of a scalar field, which can be obtained directly from the level-set function. Preliminary results in 3D are presented, showing that our method is able to capture contact angle hysteresis automatically in 3D, as well as modeling the physics of moving contact line accurately.

Our method works pretty well for most test cases except for the simulation of drop impact. We found that the wall boundary is not wetted and an air film is kept between the drop and the wall during the whole impact process, as illustrated in Fig. 5.1. Although this is possible on some superhydrophobic surface, it is unphysical in most cases. The reason for this lies in the no-penetration condition on the solid wall: as the interface gets close to the wall, the normal velocity drops to zero and thus the interface comes to rest. Physically, the van der Waals force has to be considered to rupture this film, which is beyond the scope of this work. Numerically, this thin film can be ruptured by numerical noise or the shift of interface during the reinitialization process. But our reinitialization method preserves the interfce so well that it can never reach the wall.

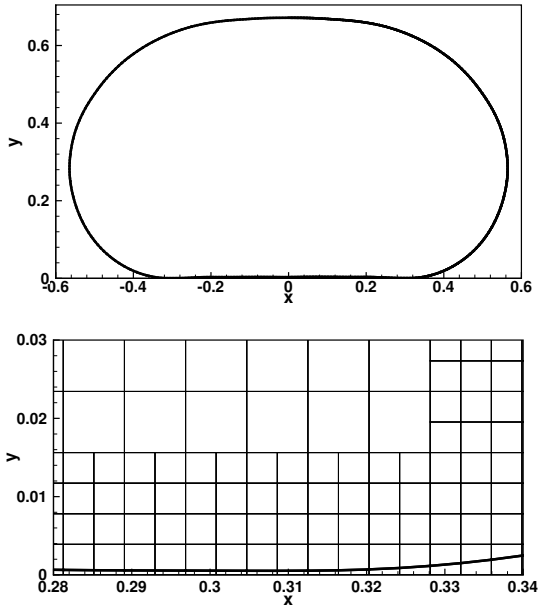


Figure 5.1: Drop impact on a solid wall by the level-set method. A thin air film is formed underneath the drop and never ruptures due to numerical artifacts.

5.2 Future directions

- **Practical guidance of β_{CL} and β_N.** In the current work, we treat β_{CL} as a fitting parameter that is tuned according to experiments or other results. To make the method more practical, it is necessary to provide a qualitative formula that guides the choice of β_{CL} and β_N.

- **Contact line dynamics in viscoelastic fluids**. Our method can also be extend to the contact line dynamics in viscoelastic fluids. The Oldroyd-B model for dilute polymer solutions can be used. The governing equation of the polymer stress, namely the upper convected Maxwell constitutive equation, can be solved by the discontinuous Galerkin method.

Bibliography

[1] J. Hnat and J. Buckmaster, Spherical cap bubbles and skirt formation *Phys. Fluids*, vol. 19, no. 2, pp. 182–194, 1976.

[2] A. Zosel, Studies of the wetting kinetics of liquid drops on solid surfaces *Colloid Polym. Sci.*, vol. 271, no. 7, pp. 680–687, 1993.

[3] A. D. Schleizer and R. T. Bonnecaze, Displacement of a two-dimensional immiscible droplet adhering to a wall in shear and pressure-driven flows *J. Fluid Mech.*, vol. 383, p. 29–54, 1999.

[4] T. Young, Iii. an essay on the cohesion of fluids *Philos. Transactions Royal Soc. Lond.*, vol. 95, pp. 65–87, 1805.

[5] C. Huh and L. E. Scriven, Hydrodynamic model of steady movement of a solid/liquid/fluid contact line *J. Colloid Interface Sci.*, vol. 35, pp. 85–101, 1971.

[6] O. V. Voinov, Hydrodynamics of wetting *J. Fluid Mech.*, vol. 11, no. 5, pp. 714–721, 1976. Translated from Izvestiya Akademii Nauk SSSR, Mekhanika Zhidkosti i Gaza, No. 5, pp. 76–84, September–October, 1976.

[7] R. G. Cox, The dynamics of the spreading of liquids on a solid surface. Part 1. Viscous flow *J. Fluid Mech.*, vol. 168, pp. 169–194, 1986.

[8] M.-Y. Zhou and P. Sheng, Dynamics of immiscible-fluid displacement in a capillary tube *Phys. Rev. Lett.*, vol. 64, pp. 882–885, 1990.

[9] T. Blake and J. Haynes, Kinetics of liquid/liquid displacement *J. Colloid Interface Sci.*, vol. 30, no. 3, pp. 421 – 423, 1969.

[10] T. Qian, X.-P. Wang, and P. Sheng, Molecular scale contact line hydrodynamics of immiscible flows *Phys. Rev. E*, vol. 68, p. 016306, 2003.

[11] W. Ren and W. E, Boundary conditions for the moving contact line problem *Phys. Fluids*, vol. 19, p. 022101, 2007.

[12] D. Jacqmin, Contact-line dynamics of a diffuse fluid interface *J. Fluid Mech.*, vol. 402, pp. 57–88, 2000.

[13] S. Zahedi, K. Gustavsson, and G. Kreiss, A conservative level set method for contact line dynamics *J. Comput. Phys.*, vol. 228, no. 17, pp. 6361 – 6375, 2009.

[14] Y. Sui, H. Ding, and P. D. Spelt, Numerical simulations of flows with moving contact lines *Annu. Rev. Fluid Mech.*, vol. 46, no. 1, pp. 97–119, 2014.

[15] S. Afkhami, S. Zaleski, and M. Bussmann, A mesh-dependent model for applying dynamic contact angles to vof simulations *J. Comput. Phys.*, vol. 228, no. 15, pp. 5370–5389, 2009.

[16] Y. Sui and P. D. Spelt, An efficient computational model for macroscale simulations of moving contact lines *J. Comput. Phys.*, vol. 242, pp. 37 – 52, 2013.

[17] D. Legendre and M. Maglio, Comparison between numerical models for the simulation of moving contact lines *Comput. & Fluids*, vol. 113, pp. 2 – 13, 2015. Small scale simulation of multiphase flows.

[18] S. Afkhami, J. Buongiorno, A. Guion, S. Popinet, Y. Saade, R. Scardovelli, and S. Zaleski, Transition in a numerical model of contact line dynamics and forced dewetting *J. Comput. Phys.*, vol. 374, pp. 1061 – 1093, 2018.

[19] Y. Yamamoto, T. Ito, T. Wakimoto, and K. Katoh, Numerical simulations of spontaneous capillary rises with very low capillary numbers using a front-tracking method

combined with generalized Navier boundary condition *Int. J. Multiph. Flow*, vol. 51, pp. 22 – 32, 2013.

[20] Y. Yamamoto, K. Tokieda, T. Wakimoto, T. Ito, and K. Katoh, Modeling of the dynamic wetting behavior in a capillary tube considering the macroscopic-microscopic contact angle relation and generalized Navier boundary condition *Int. J. Multiph. Flow*, vol. 59, pp. 106 – 112, 2014.

[21] G. Alberti and A. DeSimone, Wetting of rough surfaces: a homogenization approach *Proc. Royal Soc. Lond. A: Math. Phys. Eng. Sci.*, vol. 461, no. 2053, pp. 79–97, 2005.

[22] G. Whyman, E. Bormashenko, and T. Stein, The rigorous derivation of Young, Cassie-Baxter and Wenzel equations and the analysis of the contact angle hysteresis phenomenon *Chem. Phys. Lett.*, vol. 450, no. 4, pp. 355 – 359, 2008.

[23] S. Vedantam and M. V. Panchagnula, Constitutive modeling of contact angle hysteresis *J. Colloid Interface Sci.*, vol. 321, no. 2, pp. 393 – 400, 2008.

[24] B. K. Cheng, B. Naccarato, K. J. Kim, and A. Kumar, Theoretical consideration of contact angle hysteresis using surface-energy-minimization methods *Int. J. Heat Mass Transf.*, vol. 102, pp. 154 – 161, 2016.

[25] P. Rahimi and C. A. Ward, Contact angle hysteresis on smooth and homogenous surfaces in gravitational fields *Microgravity - Sci. Technol.*, vol. 16, pp. 231–235, Mar 2005.

[26] L. Makkonen, A thermodynamic model of contact angle hysteresis *The J. Chem. Phys.*, vol. 147, no. 6, p. 064703, 2017.

[27] P. D. Spelt, A level-set approach for simulations of flows with multiple moving contact lines with hysteresis *J. Comput. Phys.*, vol. 207, no. 2, pp. 389 – 404, 2005.

[28] H. Ding and P. D. M. Spelt, Onset of motion of a three-dimensional droplet on a wall in shear flow at moderate reynolds numbers *J. Fluid Mech.*, vol. 599, pp. 341–362, 2008.

[29] H. Ding, M. N. H. Gilani, and P. D. M. Spelt, Sliding, pinch-off and detachment of a droplet on a wall in shear flow *J. Fluid Mech.*, vol. 644, pp. 217–244, 2010.

[30] J.-B. Dupont and D. Legendre, Numerical simulation of static and sliding drop with contact angle hysteresis *J. Comput. Phys.*, vol. 229, no. 7, pp. 2453 – 2478, 2010.

[31] M. Maglio and D. Legendre, Numerical simulation of sliding drops on an inclined solid surface in *Computational and Experimental Fluid Mechanics with Applications to Physics, Engineering and the Environment* (L. D. G. Sigalotti, J. Klapp, and E. Sira, eds.), (Cham), pp. 47–69, Springer International Publishing, 2014.

[32] N. Linder, A. Criscione, I. V. Roisman, H. Marschall, and C. Tropea, 3D computation of an incipient motion of a sessile drop on a rigid surface with contact angle hysteresis *Theor. Comput. Fluid Dyn.*, vol. 29, pp. 373–390, Dec 2015.

[33] H. H. Hu, Direct simulation of flows of solid-liquid mixtures *Int. J. Multiph. Flow*, vol. 22, no. 2, pp. 335–352, 1996.

[34] P. Yue, J. J. Feng, C. A. Bertelo, and H. H. Hu, An arbitrary Lagrangian-Eulerian method for simulating bubble growth in polymer foaming *J. Comput. Phys.*, vol. 226, pp. 2229–2249, 2007.

[35] J.-F. Gerbeau and T. Lelièvre, Generalized Navier boundary condition and geometric conservation law for surface tension *Comput. Methods Appl. Mech. Eng.*, vol. 198, no. 5, pp. 644 – 656, 2009.

[36] S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows *J. Comput. Phys.*, vol. 100, no. 1, pp. 25 – 37, 1992.

[37] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan, A front-tracking method for the computations of multiphase flow *J. Comput. Phys.*, vol. 169, no. 2, pp. 708 – 759, 2001.

[38] H. Huang, D. Liang, and B. Wetton, Computation of a moving drop/bubble on a solid surface using a front-tracking method *Commun. Math. Sci.*, vol. 2, pp. 535–552, 12 2004.

[39] C. Hirt and B. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries *J. Comput. Phys.*, vol. 39, no. 1, pp. 201 – 225, 1981.

[40] M. Renardy, Y. Renardy, and J. Li, Numerical simulation of moving contact line problems using a volume-of-fluid method *J. Comput. Phys.*, vol. 171, pp. 243–263, 2001.

[41] S. Osher and J. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations *J. Comput. Phys.*, vol. 79, pp. 12–49, 1988.

[42] M. Sussman, P. Smereka, and S. Osher, A level set approach for computing solutions to incompressible two-phase flow *J. Comput. Phys.*, vol. 114, pp. 146–159, 1994.

[43] D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modeling *J. Comput. Phys.*, vol. 155, no. 1, pp. 96 – 127, 1999.

[44] M. Rudman, Volume-tracking methods for interfacial flow calculations *Int. J. for Numer. Methods Fluids*, vol. 24, no. 7, pp. 671–691, 1997.

[45] Y. Renardy and M. Renardy, Prost: A parabolic reconstruction of surface tension for the volume-of-fluid method *J. Comput. Phys.*, vol. 183, no. 2, pp. 400 – 421, 2002.

[46] I. Ginzburg and G. Wittum, Two-phase flows on interface refined grids modeled with vof, staggered finite volumes, and spline interpolants *J. Comput. Phys.*, vol. 166, no. 2, pp. 302–335, 2001.

[47] A. M. Boelens and J. J. de Pablo, Generalised Navier boundary condition for a volume of fluid approach using a finite-volume method *Phys. Fluids*, vol. 31, no. 2, p. 021203, 2019.

[48] H. S. H. Mohand, H. Hoang, G. Galliero, and D. Legendre, On the use of a friction model in a volume of fluid solver for the simulation of dynamic contact lines *J. Comput. Phys.*, vol. 393, pp. 29 – 45, 2019.

[49] D. Adalsteinsson and J. Sethian, A fast level set method for propagating interfaces *J. Comput. Phys.*, vol. 118(2), pp. 269–277, 1995.

[50] T. Hou, Z. Li, S. Osher, and H. Zhao, A hybrid method for moving interface problems with application to the heleshaw flow *J. Comput. Phys.*, vol. 134(2), pp. 236–252, 1997.

[51] W. Mulder, S. Osher, and J. Sethian, Computing interface motion in compressible gas dynamics *J. Comput. Phys.*, vol. 100(2), pp. 209–228, 1992.

[52] F. Gibou, R. Fedkiw, and S. Osher, A review of level-set methods and some recent applications *J. Comput. Phys.*, 2017.

[53] W. Ren and W. E, Derivation of continuum models for the moving contact line problem based on thermodynamic principles *Commun. Math. Sci.*, vol. 9, pp. 597–606, 2011.

[54] P. Yue and J. J. Feng, Wall energy relaxation in the Cahn-Hilliard model for moving contact lines *Phys. Fluids*, vol. 23, p. 012106, 2011.

[55] Y. Di and X.-P. Wang, Precursor simulations in spreading using a multi-mesh adaptive finite element method *J. Comput. Phys.*, vol. 228, no. 5, pp. 1380–1390, 2009.

[56] Q. Zhang, T.-Z. Qian, and X.-P. Wang, Phase field simulation of a droplet impacting a solid surface *Phys. Fluids*, vol. 28, no. 2, p. 022103, 2016.

[57] F. Bai, X. He, X. Yang, R. Zhou, and C. Wang, Three dimensional phase-field investigation of droplet formation in microfluidic flow focusing devices with experimental validation *Int. J. Multiph. Flow*, vol. 93, pp. 130–141, 2017.

[58] L. Luo, X.-P. Wang, and X.-C. Cai, An efficient finite element method for simulation of droplet spreading on a topologically rough surface *J. Comput. Phys.*, vol. 349, pp. 233 – 252, 2017.

[59] T. Omori and T. Kajishima, Apparent and microscopic dynamic contact angles in confined flows *Phys. Fluids*, vol. 29, no. 11, p. 112107, 2017.

[60] Q. He, R. Glowinski, and X.-P. Wang, A least-squares/finite element method for the numerical solution of the Navier–Stokes-Cahn–Hilliard system modeling the motion of the contact line *J. Comput. Phys.*, vol. 230, no. 12, pp. 4991–5009, 2011.

[61] M. Gao and X.-P. Wang, A gradient stable scheme for a phase field model for the moving contact line problem *J. Comput. Phys.*, vol. 231, no. 4, pp. 1372 – 1386, 2012.

[62] M. Gao and X.-P. Wang, An efficient scheme for a phase field model for the moving contact line problem with variable density and viscosity *J. Comput. Phys.*, vol. 272, pp. 704 – 718, 2014.

[63] H. Yu and X. Yang, Numerical approximations for a phase-field moving contact line model with variable densities and viscosities *J. Comput. Phys.*, vol. 334, pp. 665 – 686, 2017.

[64] J. Zhang and P. Yue, A high-order and interface-preserving discontinuous Galerkin method for level-set reinitialization *J. Comput. Phys.*, vol. 378, pp. 634–664, 2019.

[65] J. Sethian, A fast marching level set method for monotonically advancing fronts *Proc. Natl. Acad. Sci.*, vol. 93 (4), pp. 1591–1595, 1996.

[66] D. L. Chopp, Some improvements of the fast marching method *SIAM J. on Sci. Comput.*, vol. 23, no. 1, pp. 230–244, 2001.

[67] H. Zhao, A fast sweeping method for Eikonal equations *Math. Comp.*, vol. 74, pp. 603–627, 2005.

[68] Y.-T. Zhang, H.-K. Zhao, and J. Qian, High order fast sweeping methods for static Hamilton–Jacobi equations *J. Sci. Comput.*, vol. 29, no. 1, pp. 25–56, 2006.

[69] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, An adaptive level set approach for incompressible two-phase flows *J. Comput. Phys.*, vol. 148, no. 1, pp. 81 – 124, 1999.

[70] S. Pillapakkam and P. Singh, A level-set method for computing solutions to viscoelastic two-phase flow *J. Comput. Phys.*, vol. 174, no. 2, pp. 552 – 578, 2001.

[71] X. Zheng, J. Lowengrub, A. Anderson, and V. Cristini, Adaptive unstructured volume remeshing – ii: Application to two- and three-dimensional level-set simulations of multiphase flow *J. Comput. Phys.*, vol. 208, no. 2, pp. 626 – 650, 2005.

[72] J.-J. Xu, Z. Li, J. Lowengrub, and H. Zhao, A level-set method for interfacial flows with surfactant *J. Comput. Phys.*, vol. 212, no. 2, pp. 590 – 616, 2006.

[73] C. Li, C. Xu, C. Gui, and M. Fox, Level set evolution without re-initialization: a new variational formulation. in ieee computer society conference on computer vision

and pattern recognition *IEEE Comput. Soc. Conf. on Comput. Vis. Pattern Recognit.*, vol. 1, pp. 430–436, 2005.

[74] C. Basting and D. Kuzmin, A minimization-based finite element formulation for interface-preserving level set reinitialization *Comput.*, vol. 95(1), pp. 13–25, 2012.

[75] T. Utz, F. Kummer, and M. Oberlack, Interface-preserving level-set reinitialization for DG-FEM *Int. J. Num. Meth. Fluid*, vol. 84, pp. 183–198, 2017.

[76] M. Sussman, E. Fatemi, P. Smereka, and S. Osher, An improved level set method for incompressible two-phase flows *Comput. Fluids*, vol. 27, no. 5, pp. 663 – 680, 1998.

[77] M. Sussman and E. Fatemi, An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow *SIAM J. Sci. Comput.*, vol. 20 4 , pp. 1165–1191, 1999.

[78] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, A PDE-based fast local level set method *J. Comput. Phys.*, vol. 155, no. 2, pp. 410 – 438, 1999.

[79] G. Russo and P. Smereka, A remark on computing distance functions *J. Comput. Phys.*, vol. 163, no. 1, pp. 51 – 67, 2000.

[80] C. Min, On reinitializing level set functions *J. computational physics*, vol. 229, no. 8, pp. 2764–2772, 2010.

[81] A. du Chéné, C. Min, and F. Gibou, Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes *J. Sci. Comput.*, vol. 35, no. 2-3, pp. 114–131, 2008.

[82] D. Hartmann, M. Meinke, and W. Schroder, Differential equation based constrained reinitialization for level set methods *J. Comput. Phys.*, vol. 227(14), pp. 6821–6845, 2008.

[83] D. Hartmann, M. Meinke, and W. Schroder, The constrained reinitialization equation for level set methods *J. Comput. Phys.*, vol. 229, no. 5, pp. 1514 – 1535, 2010.

[84] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, A hybrid particle level set method for improved interface capturing *J. Comput. Phys.*, vol. 183, no. 1, pp. 83 – 116, 2002.

[85] M. Sussman and E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows *J. Comput. Phys.*, vol. 162, no. 2, pp. 301 – 337, 2000.

[86] S. P. van der Pijl, A. Segal, C. Vuik, and P. Wesseling, A mass-conserving level-set method for modelling of multi-phase flows *Int. J. for Numer. Methods Fluids*, vol. 47, no. 4, pp. 339–361, 2005.

[87] X. Yang, A. J. James, J. Lowengrub, X. Zheng, and V. Cristini, An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids *J. Comput. Phys.*, vol. 217, no. 2, pp. 364 – 394, 2006.

[88] E. Olsson and G. Kreiss, A conservative level set method for two phase flow *J. Comput. Phys.*, vol. 210(1), pp. 225–246, 2005.

[89] E. Olsson, G. Kreiss, and S. Zahedi, A conservative level set method for two phase flow ii *J. Comput. Phys.*, vol. 225, no. 1, pp. 785 – 807, 2007.

[90] T. Biben and C. Misbah, Tumbling of vesicles under shear flow within an advected-field approach *Phys. Rev. E*, vol. 67, p. 031908, 2003.

[91] M. Owkes and O. Desjardins, A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows *J. Comput. Phys.*, vol. 249, pp. 275 – 302, 2013.

[92] Z. Jibben and M. Herrmann, An arbitrary-order Runge–Kutta discontinuous Galerkin approach to reinitialization for banded conservative level sets *J. Comput. Phys.*, vol. 349, pp. 453 – 473, 2017.

[93] R. Saye, High-order methods for computing distances to implicitly defined surfaces *Commun. Appl. Math. Comput. Sci.*, vol. 9, no. 1, pp. 107–141, 2014.

[94] B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws. II: General framework *Math. Comput.*, vol. 52, pp. 411–435, 1989.

[95] B. Cockburn, S. Y. Lin, and C.-W. Shu, Tvb Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems *J. Comput. Phys.*, vol. 84, pp. 90–113, 1989.

[96] B. Cockburn, S. Hou, and C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: the multidimensional case *Math. Comput.*, vol. 54, pp. 545–581, 1990.

[97] Y.-T. Zhang and C.-W. Shu, High-order WENO schemes for Hamilton–Jacobi equations on triangular meshes *SIAM J. on Sci. Comput.*, vol. 24, no. 3, pp. 1005–1030, 2003.

[98] D. Levy, S. Nayak, C.-W. Shu, and Y.-T. Zhang, Central WENO schemes for Hamilton–Jacobi equations on triangular meshes *SIAM J. on Sci. Comput.*, vol. 28, no. 6, pp. 2229–2247, 2006.

[99] C. Hu and C.-W. Shu, A discontinuous Galerkin finite element method for Hamilton-Jacobi equations *SIAM J. Sci. Comput.*, vol. 21, pp. 666–690, 1999.

[100] F. Li and C.-W. Shu, Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton-Jacobi equations *Appl. Math. Lett.*, vol. 18, pp. 1204–1209, 2005.

[101] Y. Cheng and C.-W. Shu, A discontinuous Galerkin finite element method for directly solving the Hamilton-Jacobi equations *J. Comput. Phys.*, vol. 223, pp. 398–415, 2007.

[102] J. Yan and S. Osher, A local discontinuous Galerkin method for directly solving Hamilton-Jacobi equations *J. Comput. Phys.*, vol. 239, pp. 232–244, 2011.

[103] C.-W. Shu, Survey on discontinuous Galerkin methods for Hamilton-Jacobi equations *Contemp. Math.*, vol. 586, pp. 323–330, 2013.

[104] S. Fechter and C.-D. Munz, A discontinuous Galerkin-based sharp-interface method to simulate three-dimensional compressible two-phase flow *Int. J. for Numer. Methods Fluids*, vol. 78, no. 7, pp. 413–435, 2015.

[105] E. Marchandise, J.-F. Remacle, and N. Chevaugeon, A quadrature-free discontinuous Galerkin method for the level set equation *J. Comput. Phys.*, vol. 212, pp. 338–357, Feb. 2006.

[106] J. Grooss and J. Hesthaven, A level set discontinuous Galerkin method for free surface flows *Comput. Methods Appl. Mech. Eng.*, vol. 195, no. 25, pp. 3406 – 3429, 2006.

[107] A. Karakus, T. Warburton, M. Aksel, and C. Sert, A GPU accelerated level set reinitialization for an adaptive discontinuous Galerkin method *Comput. Math. Appl.*, vol. 72, no. 3, pp. 755 – 767, 2016.

[108] R. F. Ausas, E. A. Dari, and G. C. Buscaglia, A geometric mass-preserving redistancing scheme for the level set function *Int. J. for Numer. Methods Fluids*, vol. 65, no. 8, pp. 989–1010, 2011.

[109] N. Parolini, Computational fluid dynamics for naval engineering problems *PhD thesis, EPFL Lausanne*, 2004.

[110] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids *J. Comput. Phys.*, vol. 227, no. 4, pp. 2674 – 2706, 2008.

[111] B. Muller, F. Kummer, and M. Oberlack, Highly accurate surface and volume integration on implicit domains by means of moment-fitting *Int. J. for Numer. Methods Eng.*, vol. 96, no. 8, pp. 512–528, 2013.

[112] B. Cockburn and C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems *J. Comput. Phys.*, vol. 141, pp. 199–224, 1998.

[113] J. Qiu and C.-W. Shu, Runge–Kutta discontinuous Galerkin method using WENO limiters *SIAM J. on Sci. Comput.*, vol. 26, no. 3, pp. 907–929, 2005.

[114] J. Zhu, J. Qiu, C.-W. Shu, and M. Dumbser, Runge–Kutta discontinuous Galerkin method using WENO limiters ii: Unstructured meshes *J. Comput. Phys.*, vol. 227, no. 9, pp. 4330 – 4353, 2008.

[115] J. Zhu, X. Zhong, C.-W. Shu, and J. Qiu, Runge–Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes *J. Comput. Phys.*, vol. 248, pp. 200 – 220, 2013.

[116] S. Gottlieb and C.-W. Shu, Total variation diminishing Runge-Kutta schemes *Math. Comp.*, vol. 67, pp. 73–85, 1998.

[117] B. Cockburn and C.-W. Shu, Runge-Kutta discontinuous Galerkin finite element methods for convection-dominated problems *J. Sci. Comput.*, vol. 16, pp. 173–261, 2001.

[118] W. Bangerth, R. Hartmann, and G. Kanschat, deal.II – a general purpose object oriented finite element library *ACM Trans. Math. Softw.*, vol. 33, no. 4, pp. 24/1–24/27, 2007.

[119] W. Bangerth, D. Davydov, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, and D. Wells, The `deal.II` library, version 8.4 *J. Numer. Math.*, vol. 24, 2016.

[120] J. Bell, P. Colella, and H. Glaz, A second-order projection method for the incompressible Navier-Stokes equations *J. Comput. Phys.*, vol. 85(2), pp. 257–283, 1989.

[121] W. Rider and D. Kothe, Reconstructing volume tracking *J. Comput. Phys.*, vol. 141(2), pp. 112–152, 1998.

[122] P. Gomez, J. Hernandez, and J. Lopez, On the reinitialization procedure in a narrow-band locally refined level set method for interfacial flows *Int. journal for numerical methods engineering*, vol. 63, no. 10, pp. 1478–1512, 2005.

[123] R. J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow *SIAM J. on Numer. Analysis*, vol. 33, no. 2, pp. 627–665, 1996.

[124] C. Min and F. Gibou, A second order accurate level set method on non-graded adaptive cartesian grids *J. Comput. Phys.*, vol. 225, no. 1, pp. 300 – 321, 2007.

[125] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids *J. computational physics*, vol. 31, no. 3, pp. 335–362, 1979.

[126] J. Zhang and P. Yue, A level-set method for moving contact lines with contact angle hysteresis *J. Comput. Phys.*, 2019. submitted.

[127] W. Ren and W. E, Contact line dynamics on heterogeneous surfaces *Phys. Fluids*, vol. 23, p. 072103, 2011.

[128] J.-J. Xu and W. Ren, A level-set method for two-phase flows with moving contact line and insoluble surfactant *J. Comput. Phys.*, vol. 263, pp. 71 – 90, 2014.

[129] Z. Zhang and W. Ren, Simulation of moving contact lines in two-phase polymeric fluids *Comput. & Math. with Appl.*, vol. 72, no. 4, pp. 1002 – 1012, 2016.

[130] Q. Zhao and W. Ren, An energy-stable finite element method for the simulation of moving contact lines in two-phase flows *arXiv preprint arXiv:2002.12009*, 2020.

[131] L. Wang, H.-B. Huang, and X.-Y. Lu, Scheme for contact angle and its hysteresis in a multiphase lattice boltzmann method *Phys. Rev. E*, vol. 87, p. 013301, Jan 2013.

[132] S. Shin, J. Chergui, and D. Juric, Direct simulation of multiphase flows with modeling of dynamic interface contact angle *Theor. Comput. Fluid Dyn.*, vol. 32, pp. 655–687, Oct 2018.

[133] P. Yue, A thermodynamically consistent phase-field method for contact angle hysteresis *J. Fluid Mech.*, 2019. submitted.

[134] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, Modelling merging and fragmentation in multiphase flows with surfer *J. Comput. Phys.*, vol. 113, no. 1, pp. 134 – 147, 1994.

[135] T. D. Blake, The physics of moving wetting lines *J. Colloid Interface Sci.*, vol. 299, pp. 1–13, 2006.

[136] G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmoeller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells, The `deal.II` library, version 9.0 *J. Numer. Math.*, vol. 26, no. 4, pp. 173–183, 2018.

[137] T. A. Davis, Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method *ACM Trans. Math. Softw.*, vol. 30, pp. 196–199, June 2004.

[138] G. Ryskin and L. Leal, Numerical solution of free-boundary problems in fluid mechanics. Part 2. Buoyancy-driven motion of a gas bubble through a quiescent liquid *J. Fluid Mech.*, vol. 148, pp. 19–35, 1984.

[139] M. Sussman and P. Smereka, Axisymmetric free boundary problems *J. Fluid Mech.*, vol. 341, pp. 269–294, 1997.

[140] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows *J. Comput. Phys.*, vol. 152, no. 2, pp. 423 – 456, 1999.

[141] M. Ohta, T. Imura, Y. Yoshida, and M. Sussman, A computational study of the effect of initial bubble conditions on the motion of a gas bubble rising in viscous liquids *Int. J. Multiph. Flow*, vol. 31, no. 2, pp. 223–237, 2005.

[142] E. D. Wilkes, S. D. Phillips, and O. A. Basaran, Computational and experimental analysis of dynamics of drop formation *Phys. fluids*, vol. 11, no. 12, pp. 3577–3598, 1999.

[143] C. Zhou, P. Yue, and J. J. Feng, Formation of simple and compound drops in microfluidic devices *Phys. fluids*, vol. 18, no. 9, p. 092105, 2006.

[144] P. Yue, C. Zhou, and J. J. Feng, Sharp-interface limit of the Cahn-Hilliard model for moving contact lines *J. Fluid Mech.*, vol. 645, pp. 279–294, 2010.

[145] R. L. Hoffman, A study of the advancing interface *J. Colloid Interface Sci.*, vol. 50, pp. 228–241, 1975.

[146] M. Wörner, X. Cai, H. Alla, and P. Yue, A semi-analytical method to estimate the effective slip length of spreading spherical-cap shaped droplets using cox theory *Fluid Dyn. Res.*, vol. 50, no. 3, p. 035501, 2018.

[147] V. V. Khatavkar, P. D. Anderson, and H. E. H. Meijer, Capillary spreading of a droplet in the partially wetting regime using a diffuse-interface model *J. Fluid Mech.*, vol. 572, pp. 367–387, 2007.

[148] X. Cai, H. Marschall, M. Wöner, and O. Deutschmann, Numerical simulation of wetting phenomena with a phase-field method using openfoam *Chem. Eng. & Technol.*, vol. 38, no. 11, pp. 1985–1992, 2015.

[149] J. Luo, X. Hu, and N. A. Adams, Curvature boundary condition for a moving contact line *J. Comput. Phys.*, vol. 310, pp. 329–341, 2016.

[150] C. Furmidge, Studies at phase interfaces. i. the sliding of liquid drops on solid surfaces and a theory for spray retention *J. Colloid Sci.*, vol. 17, no. 4, pp. 309 – 324, 1962.

[151] E. B. Dussan V. and R. T.-P. Chow, On the ability of drops or bubbles to stick to non-horizontal surfaces of solids *J. Fluid Mech.*, vol. 137, pp. 1–29, 1983.

[152] S. Xu and W. Ren, Reinitialization of the level-set function in 3d simulation of moving contact lines *Commun. Comput. Phys.*, vol. 20, no. 5, pp. 1163–1182, 2016.

[153] Z. Solomenko, P. D. Spelt, and P. Alix, A level-set method for large-scale simulations of three-dimensional flows with moving contact lines *J. Comput. Phys.*, vol. 348, pp. 151–170, 2017.

[154] G. Della Rocca and G. Blanquart, Level set reinitialization at a contact line *J. Comput. Phys.*, vol. 265, pp. 34 – 49, 2014.

[155] A. J. Chorin, Numerical solution of the Navier-Stokes equations *Math. computation*, vol. 22, no. 104, pp. 745–762, 1968.

[156] B. Eral, D. 't Mannetje, and J. Oh, Contact angle hysteresis: a review of fundamentals and applications *Colloid polymer science*, vol. 291, pp. 247–260, 9 2013.