Threshold Dynamics Method: Theories, Algorithms, and Applications

by

Dong Wang

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
in Mathematics

June 2017, Hong Kong

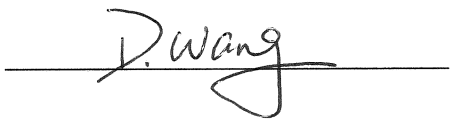
HKUST Library
Reproduction is prohibited without the author's prior written consent

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.



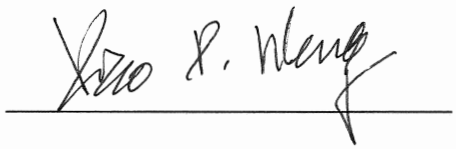
Dong Wang

Threshold Dynamics Method: Theories, Algorithms, and Applications

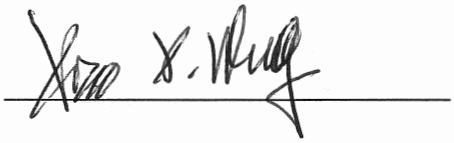
by

Dong Wang

This is to certify that I have examined the above PhD thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.



Prof. Xiao-Ping WANG, Thesis Supervisor



Prof. Xiao-Ping WANG, Acting Head of Department

Department of Mathematics

June 2017

Acknowledgment

First and foremost, I would like to express my deep gratitude to my supervisor, Prof. Xiao-Ping Wang. His high intellectual standard and enthusiasm for research have transformed entirely the path of my studies and research. His wealth of ideas and clarity of thought have made my PhD journey exceptionally enriching and enjoyable. I am also eternally indebted to Prof. Qiang Du(Columbia U) who gave me many advice, supervision, and kindly support when I was exchanging at the Department of Applied Physics and Applied Mathematics in the Columbia University.

I would like to express my deep thanks to Prof. Yaguang Wang(Shanghai Jiaotong U) for the collaboration on the analysis of the sharp interface limit of three phase Cahn-Hilliard equation. I am especially grateful to Prof. Xianmin Xu(Chinese Academy of Sciences) for the collaboration on the application of threshold dynamics method into solid wetting problems. Also, I would like to thank Prof. Shidong Jiang(New Jersey Institute of Technology) who helped me a lot on using non-uniform fast Fourier transform(NUFFT) and gave me many good advice and supervision.

I am grateful to my thesis committee members, Prof. Tiezheng Qian and Prof. Yang Xiang for their guidance and support through all these years. I would also like to thank Prof. Shing-Yu Leung for his continuous encouragement and his courses which have provided inspirations on my thesis work. My sincere appreciation also goes to Prof. Tony F. Chan for helpful discussions about iterative thresholding method for image segmentation.

I would also like to thank Prof. Xingfu Chen(U Pittsburgh) for helpful discussions on the proof of convergence of threshold dynamics method, Prof. Jingfang Huang(UNC-Chapel Hill) for the help on using fast multipole method (FMM),

Prof. Guanghui Hu(U Macau) for introducing AFE package, Prof. Zuowei Shen(NUS) and Prof. Xiaoqun Zhang(Shanghai Jiaotong U) for valuable suggestions on image segmentation problems, and Prof. Braxton Osting(U Utah) for the discussion on optimal partition and super solid problems. My appreciation also goes to Prof. Selim Esedgolu(U Michigan), Prof. Lesile Greengard(NewYork U) and Prof. Hongkai Zhao(UC Irvine) for their helpful discussions and suggestions.

I am fortunate to have had many great fellow PG students and friends throughout these four years. I would like to express my appreciation to my fellow PG students and postdocs in Prof. Wang's group, Min Gao, Yi Shi, Li Luo, Qiaoling He, Zhen Zhang, Hua Zhong, Xing Zhang, Xiaoyu Wei, Haohan Li, Wei Hu, Wei Xu, Xiang Li, Jizu Huang, and Jingrun Chen for all the helpful discussions, general help, kindly encouragement and continuous support. Also, I would like to thank students and postdocs in Prof. Qiang Du's group, Xiaochuan Tian, Yunzhe Tao, Xiaobo Yin, Qi Sun, Zhi Zhou, and Jiang Yang for all the good time they spent with me and insightful discussions they shared with me.

I would also like to acknowledge two groups of friends: Lab Basketball Association (LBA) and table game group. Recreation is always dispensable especially when the research life runs dry. My appreciation also goes to my old friends, Qiulin Deng, Dongfang Lou, and Zhun Sun for their continuous encouragement, kindly support, and accompany through all these years.

Last but not the least, I would like to thank my beloved family especially my parents for the year round support and accompany through my life. My sincere appreciation also goes to Ms. Shengqing Hu, who has always been in favour of me.

Contents

Title Page **i**

Authorization Page **ii**

Signature Page **iii**

Acknowledgments **iv**

Table of Contents **vi**

Abstract **xvii**

1 Introduction **1**

2 An efficient threshold dynamics method for solid wetting problems **9**

2.1 The minimization of surface energies 11

2.2 A new threshold dynamics method for the wetting problem 15

2.2.1 The representation of interface energies in an extended domain . . . 15

2.2.2 Derivation of the threshold dynamics method . . . 17

2.2.3 A simplified algorithm for the two-phase problem . . . 22

2.2.4 Stability analysis . . . 24

2.3 Numerical implementation and accuracy check . . . 25

2.3.1 Calculation of convolution . . . 25

2.3.2 A quicksort algorithm for volume preservation . . . 26

2.3.3 Accuracy check of Algorithm I . . . 28

2.3.4 A time refinement scheme for contact point motion . . . 31

2.3.5 Accuracy check of the Modified Algorithm I . . . 33

2.4 A drop spreading on a chemically pattern solid surface . . . 35

2.5 A drop spreading on a rough solid surface . . . 38

3 Convergence proof of the threshold dynamics method for solid wetting problems 44

3.1 Algorithm and notations . . . 45

3.2 Minimizing movement interpretation . . . 51

3.3 Main results . . . 54

3.4 Estimate for λ . . . 55

3.5 Compactness . . . 63

3.6 Convergence 67

4 An efficient boundary integral scheme for the threshold dynamics method via NUFFT 71

4.1 NUFFT based solver for the heat equation in free space 74

4.1.1 Fourier spectral approximation of the heat kernel for a fixed time 75

4.1.2 Solving the pure initial value problem 77

4.1.3 Extension to non-smooth boundary 80

4.2 Thresholding 82

4.3 Numerical results 84

4.3.1 Example 1: Accuracy of the NUFFT based heat solver . . 86

4.3.2 Example 2: Efficiency and accuracy of the threshold dynamics method 87

4.3.3 Example 3: Interface motion by mean curvature in 2D and 3D 90

4.3.4 Example 4: Area preserving motion of a hexagram in 2D . 91

4.3.5 Example 5: Volume preserving motion of a wedge in 3D . 92

4.3.6 Example 6: Mean curvature motion of hexagram in 2D . . 94

4.3.7 Example 7: Application in solid wetting problems in 2D . 96

4.3.8 Example 8: Two droplets merging in 2D 98

4.3.9 Example 9: Application in solid wetting problems in 3D . 99

5 An efficient threshold dynamics method for image segmentation problems 102

5.1 An efficient threshold dynamics method for image segmentation . 106

5.1.1 The approximate Chan-Vese functional 106

5.1.2 Derivation of the threshold dynamics method 111

5.2 Numerical Results 115

5.2.1 Example 1: Cameraman 115

5.2.2 Example 2: Image with heavy noise 117

5.2.3 Example 3: A synthetic four-phase image 118

5.2.4 Example 4: Flower color image 119

6 Conclusion and future work 120

List of Figures

1.1 Diagram for mean curvature flow: Left: Initial motion along normal direction, right: evolved interface (black solid line). 2

1.2 Diagram for one iteration step in the MBO method, from left to right: 1. Initial domain, 2. heat diffusion after time step Δt, 3. new domain. 4

1.3 Diagram for the local coordinate for the boundary. 5

2.1 Wetting on a rough surface 12

2.2 Extended computational domain $\tilde{\Omega} = \Omega \cup D_3$ 16

2.3 From left to right, top to bottom: 1. The initial condition defined by the characteristic function of the domains. 2. The X-Z plane of the convolution between the heat kernel and the initial condition calculated by extending the domain by reflection (i.e. extending $[-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$ to $[-\frac{\pi}{2}, \frac{3\pi}{2}] \times [-\frac{\pi}{2}, \frac{3\pi}{2}]$). 3. The X-Z plane of the convolution between the heat kernel and the initial condition calculated without extending the domain. 4. The difference between the second figure and the third figure. 27

2.4 An example to demonstrate our new scheme for volume preservation. Initially, there are M interior points. After convolution, we select M grid points with the M lowest values as the new interior points. 28

2.5 The two circles at $t = 0$ (left) and $t = 0.02$(right). 29

2.6 The two semi-circles at $t = 0$ (left) and $t = 0.02$(right). 30

2.7 Left: 256×256 grid points, $\delta t = 2dx$ without refinement in time. Right: 256×256 grid points, $\delta t = 2dx$ initially with refinement in time, $\epsilon = 1.0e^{-10}$ 35

2.8 Left: 512×512 grid points, $\delta t = 2dx$ without refinement in time. Right: 512×512 grid points, $\delta t = 2dx$ initially with refinement in time, $\epsilon = 1.0e^{-10}$ 35

2.9 Left: 1024×1024 grid points, $\delta t = 2dx$ without refinement in time. Right: 1024×1024 grid points, $\delta t = 2dx$ initially with refinement in time, $\epsilon = 1.0e^{-10}$ 36

2.10 Comparison of the numerical solution at equilibrium to the exact solution. Red line represents the exact solution while the blue line represents the numerical solution (computed with 1024×1024 grid points). 37

2.11 A sketch of a drop spreading on a chemically patterned solid surface. Here D_3 (white region) and D_4 (shaded region) represent materials $\mathcal{A}$ and $\mathcal{B}$ respectively. 38

2.12 The stick-slip motion of a drop with k=2 when the volume is increasing. $\theta_{\mathcal{A}} = \frac{\pi}{5}$, $\theta_{\mathcal{B}} = \frac{7\pi}{10}$. 38

2.13 The stick-slip motion of a drop with k=4 when the volume is increasing. $\theta_{\mathcal{A}} = \frac{\pi}{5}$, $\theta_{\mathcal{B}} = \frac{7\pi}{10}$. 39

2.14 The stick-slip motion of a drop with k=2 when the volume is decreasing. $\theta_{\mathcal{A}} = \frac{\pi}{5}$, $\theta_{\mathcal{B}} = \frac{7\pi}{10}$. 39

2.15 The stick-slip motion of a drop with k=4 when the volume is decreasing. $\theta_{\mathcal{A}} = \frac{\pi}{5}$, $\theta_{\mathcal{B}} = \frac{7\pi}{10}$. 40

2.16 Left: A quasi-static drop in the process of growing in volume on a chemically patterned surface when the initial volume is 0.5883. Right: A quasi-static drop in the process of shrinking in volume on a chemically patterned surface when the initial volume is 0.5883. Young's angle in the light blue part is $\mathcal{B}$ while that in the dark blue part is $\mathcal{A}$. Here k=4. 41

2.17 Left: A sketch of a drop spreading on a rough solid surface. The solid surface is given by a sawtooth profile. Right: Diagrammatic sketch of effective contact angle θ on the rough solid surface. . . . 42

2.18 The stick-slip motion of a drop on a rough surface when the volume is increasing. $\theta = \frac{\pi}{2}, k = 4, \alpha = \frac{\pi}{6}$. 42

2.19 The stick-slip motion of a drop on a rough surface when the volume is decreasing. $\theta = \frac{\pi}{2}, k = 4, \alpha = \frac{\pi}{6}$. 42

2.20 Left: A quasi-static drop in the process of expanding in volume on a sawtooth rough surface when the volume is 1.178. Right: A quasi-static drop in the process of reducing in volume on a sawtooth rough surface when the volume is 1.178. $\theta = \frac{\pi}{2}, k = 4, \alpha = \frac{\pi}{6}$. 43

4.1 The Motion by Mean Curvature of a Crescent at Various Times. We set $\Delta t = 0.0005$ and use 400 points to discrete the interface. . 90

4.2 Snapshots of the Motion by Mean Curvature of a Bowl. $\Delta t = 0.001$. 91

4.3 Left: subsets of the source points at $t = 0$. Right: subsets of the target points at $t = 0$. 91

4.4 Snapshots of the Motion by Mean Curvature of a Hexagram under Area Preserving Constraint. $\Delta t = 0.001$. 92

4.5 The front, side, and vertical views of the initial geometric profile in Example 5. 93

4.6 Snapshots of the Volume Preserving Motion of a Wedge. $\Delta t = 0.001$. 93

4.7 Snapshots of the Mean Curvature Motion of a Hexagram. $\Delta t = 0.0005$. 95

4.8 The Diagram of Target Points Generated Along a Non-smooth Hexagram Interface. 96

4.9 Snapshots of Wetting Process on Hydrophilic Solid Material with Young's Angle as $\pi/3$. 97

4.10 Snapshots of Wetting Process on Hydrophobic Solid Material with Young's Angle as $2\pi/3$. 97

4.11 Initial Profile of Two Droplets on the Solid Surface. 98

4.12 Snapshots of Two Droplets Merging Together on Solid Surface. . . 99

4.13 Snapshots of Wetting Process on Hydrophobic Solid Material (3D view and front view both) with Young's Angle as $2\pi/3$. 100

4.14 Snapshots of Wetting Process on Hydrophilic Solid Material (3D view and front view both) with Young's Angle as $\pi/3$. 101

5.1 Segmentation results for the classic cameraman image with $\delta t = 0.03$ and $\lambda = 0.01$. The algorithm converges in 15 iterations with a computational time of 0.1188 seconds 116

5.2 Energy curve for the iteration algorithm with $\delta t = 0.03$ and $\lambda = 0.01$. 116

5.3 Segmentation contours and energy curves for $\delta t = 0.03$ and different λ values. 117

5.4 Noisy Image Segmentation with $\delta t = 0.03$, $\lambda = 0.1$. 117

5.5 Segmentation for images with different resolutions and with the parameters $\delta t = 0.01$ and $\lambda = 0.003$ 118

5.6 Two-phase segmentation for a 375×500 RGB image and with parameters $\delta t = 0.01$ and $\lambda = 0.005$. 119

5.7 Four phase segmentation for a 375×500 RGB image with $\delta t = 0.01$ and $\lambda = 0.003$. 119

List of Tables

2.1 Accuracy check of Algorithm I for the two circle motion 30

2.2 Accuracy check of Algorithm I for the motion of two semi-circles on solid boundary. 31

2.3 Accuracy Check in L^1 norm 34

2.4 Accuracy Check in L^∞ norm 34

4.1 Number of Fourier nodes needed to approximate the 1D heat kernel $G_1(x, \Delta t)$ for $x \in [-\pi, \pi]$. 77

4.2 Relative L^2 error versus number of discretization points on the boundary for the 2D heat solver. The boundary curve is a smooth hexagram in Example 4. 86

4.3 Relative L^2 error versus number of discretization points on the boundary for the 3D heat solver. The boundary is a sphere of radius 0.5. 87

4.4 Accuracy and timing results of Algorithm 2 and the uniform mesh method 88

4.5 Relative error and convergence order in time of Algorithm 2 for the motion by mean curvature of a circle. 88

4.6 Numerical Results for the Motion by Mean Curvature of a Sphere. 89

4.7 Relative L^2 error versus number of discretization points on the boundary for the 2D heat solver. The boundary curve is a non-smooth hexagram. 94

Threshold Dynamics Method: Theories, Algorithms, and Applications

Dong Wang

Department of Mathematics

Abstract

In this thesis, efficient threshold dynamics methods are developed and analysed for free interface problems including wetting dynamics and image segmentation.

The threshold dynamics method developed by Merriman, Bence and Osher (MBO) is an efficient method for simulating the motion by mean curvature flow. Direct generalization of MBO-type methods to the wetting problem with interfaces intersecting the solid boundary is not easy because solving the heat equation in a general domain with a wetting boundary condition is not as efficient as it is with the original MBO method. The dynamics of the contact point also follows a different law compared with the dynamics of the interface away from the boundary. In this thesis, we develop an efficient volume preserving threshold dynamics method for simulating wetting on rough surfaces. This method is based on minimization of the weighted surface area functional over an extended domain that includes the solid phase. The method is simple, stable with $O(NlogN)$ complexity per time step and is not sensitive to the inhomogeneity or roughness of the solid boundary. The convergence of the threshold dynamics method is rigorously analysed.

To further improve the efficiency of the algorithm, we propose an efficient boundary integral scheme for threshold dynamics via non-uniform fast Fourier transform (NUFFT). The first step is carried out by the evaluation of a boundary

integral via NUFFT, and the second step is performed via a root finding algorithm along the normal directions of a discrete set of points on the interface. Unlike most existing methods where a volume discretization is needed for the whole computational domain, our scheme requires the discretization of physical space only in a small neighborhood of the interface and thus is mesh free. The algorithm is spectrally accurate in space for smooth interfaces and has $O(NlogN)$ complexity with N the total number of discrete points near the interface when the time step Δt is not very small. The performance of the algorithm is illustrated via several numerical examples in both two and three dimensions.

We also propose an efficient threshold dynamics method for multi-phase image segmentation. The algorithm is based on minimizing piecewise constant Mumford-Shah functional in which the contour length (or perimeter) is approximated by a non-local multi-phase energy. The minimization problem is solved by an iterative method. Each iteration consists of computing simple convolutions followed by a thresholding step. The algorithm is easy to implement and has the optimal complexity $O(NlogN)$ per iteration. We also show that the iterative algorithm has the total energy decaying property. We present some numerical results to show the efficiency of our method.

Chapter 1

Introduction

Interface motion has been an important topic in many areas of applied mathematics and scientific computing ranging from fluid mechanics, imaging processing, and computer vision. Among them, curvature dependent interface motion plays an important role and attracts many attentions. Specifically, curvature dependent interface motion is given by

$$V_n(x) = f(\kappa(x)) \tag{1.1}$$

where $V_n(x)$ is the normal velocity of a point x on the interface, $\kappa(x)$ is the sum of principal curvature (or mean curvature) at x, and f is a given function defined on $\mathbb{R}$. The simplest example is motion by mean curvature when $f(\kappa) = \kappa$. The mean curvature flow minimizes the surface area, and minimal surfaces are the critical points for the mean curvature flow (See Figure 1.1 for instance).

Many numerical methods have been developed for simulating interface dynamics including front tracking methods [92, 58], level set methods [35, 72, 99], and phase field methods [40, 98, 24, 54, 18].

In front tracking methods, the curve is replaced by a set of discrete points, then the curvature κ or $f(\kappa)$ can be calculated by finite difference method or paramet-

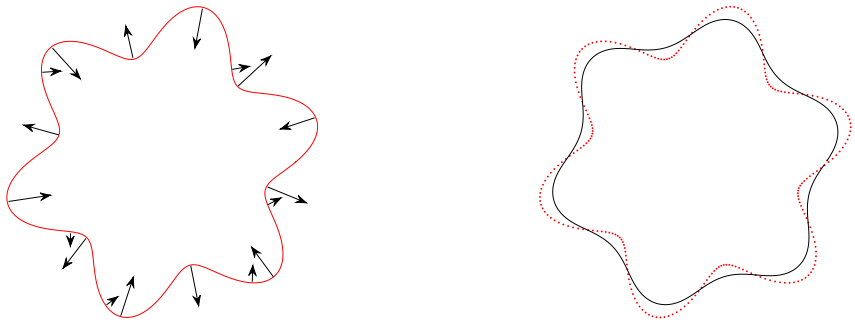


Figure 1.1: Diagram for mean curvature flow: Left: Initial motion along normal direction, right: evolved interface (black solid line).

ric finite element method. Then, one can calculate the normal direction at each discrete point and move those points according to the corresponding motion law. This has the advantages of high accuracy and computational efficiency. However, the main disadvantage is that the algorithm is not efficient to deal with problems with topological change. Also, it is common that we need to deal with some free interface problems with triple junctions or other junctions which are quite important in many physical problems including solid wetting, solid de-wetting, multi phase flow problems, crystal growth, and so on. It is almost impossible for us to have the explicit motion law of the junctions. We refer to a recent work [89] on deriving the explicit dynamic of the junctions in three phase system modelled by Cahn-Hillard equation [13]. The motion law derived is often very complicated and it is extremely difficult to calculate at junctions explicitly.

In level set methods, the curve moves along the normal direction with the velocity V_n is represented by a level set of a function ϕ on the plane. Then this function in turn evolves according to a partial differential equation of Hamilton-Jacobi type:

$$\frac{\partial \phi}{\partial t} = V_n |\nabla \phi| \tag{1.2}$$

where $|\cdot|$ is the Euclidean norm and ∇ represents the spatial gradient. The

advantage here is that the interface can be implicitly determined by the zero level set of a function. Thus, the topological change case does not require separate treatment. The disadvantage of the methods is difficult to deal with interface motions with multiple junctions.

In phase field methods, the interface is represented by a level set (for instance zero level set) of the order parameter function φ which takes two distinct values (for instance $+1$ and -1), with a smooth change between both values in the extremely thin region around the interface. Specifically, one can model the motion by mean curvature by the following Allen-Cahn equation[4, 3]:

$$\frac{\partial \varphi}{\partial t} = \epsilon \Delta \varphi - \frac{1}{\epsilon} f^{'}(\varphi) \tag{1.3}$$

where Δ here represents the spatial Laplace operator, ϵ is the interface thickness, and $f^{'}(\varphi) = \varphi^3 - \varphi$. It is the L^2 gradient flow of the energy functional:

$$\mathcal{E}^{pf}(\varphi) = \frac{\epsilon}{2} \int |\nabla \varphi|^2 \mathrm{d}\mathbf{x} + \frac{1}{\epsilon} \int f(\varphi) \mathrm{d}\mathbf{x} \tag{1.4}$$

where $f(\varphi) = \frac{1}{4}(\varphi^2 - 1)^2$ is the double well potential. The model is also called "diffused interface model". Upon minimization of this functional, the double well potential will force φ to go to 1 or -1; however, the H^1 term forces φ to have some smoothness, thereby removing sharp jumps between the two minima of f. The resulting minimization leads to regions where φ is approximately 1 or -1 with a very thin, $O(\epsilon)$ scale transition region between the two. Thus the minimizer appears to have two phases with an interface between them. We remark that the energy functional (1.4) is roughly proportional to the perimeter of this interface. This can be made rigorous by considering the notion of Gamma convergence of the (1.4) [52]. Also, note that the double well typically restricts φ to take on integral order values (between minus one and one). Hence, it is quite robust from numerical point of view. There have been many successful numerical methods designed for Allen-Cahn equation or Cahn-Hillard equation [50, 87, 51, 39, 40, 60]. However, phase field models simulations must resolve the

thin interface region with thickness of ϵ scale. This would make the simulation very expensive on uniform grid.

In 1992, Merriman, Bence, and Osher (MBO) [64] developed a threshold dynamics method for the motion of the interface driven by the mean curvature, where the governing equation is $V_n(x) = \kappa(x)$. To be more precise, let $D \in \mathbb{R}^n$ be a domain where its boundary $\Gamma = \partial D$ is to be evolved via the motion by mean curvature. For each time step, the MBO method generates the new interface Γ_{new} (or equivalently, D_{new}) via the following two steps:

Step 1. Solve the pure initial value problem of the heat equation to obtain its solution u^* at $t = \Delta t$

$$u_t = \Delta u, \tag{1.5}$$

$$u|_{t=0} = \chi_D \tag{1.6}$$

where χ_D is the indicator function of domain D.

Step 2. Obtain the domain D_{new} by

$$D_{\text{new}} = \left\{ \mathbf{x} : u^*(\mathbf{x}, \Delta t) \geq \frac{1}{2} \right\}. \tag{1.7}$$

Figure 1.2 gives a diagram of one iteration step in the MBO method.

Figure 1.2: Diagram for one iteration step in the MBO method, from left to right: 1. Initial domain, 2. heat diffusion after time step Δt, 3. new domain.

Intuitively, from Figure 1.2, one can quickly see that the diffusion rapidly blunts the sharp points on the boundary, but has extremely little effect on the flatter

parts. It seems that the above two steps can generate some curvature dependent motion. Some formal analysis can be carried out to understand the MBO method. Consider a point P on the boundary, and construct a local polar coordinate with the origin at the center of the circle of curvature at point P (See Figure 1.3). If we write the diffusion equation in the local coordinate, we can have

$$\frac{\partial u}{\partial t} = \frac{1}{r}\frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}. \tag{1.8}$$

Because of the local symmetry of the boundary, we have $\frac{\partial u}{\partial \theta} = \frac{\partial^2 u}{\partial \theta^2} = 0$. Then, the equation will be reduced to

$$\frac{\partial u}{\partial t} = \frac{1}{r}\frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial r^2} \tag{1.9}$$

which is an convection-diffusion equation. Note that the $\frac{1}{2}$ level set does not change if take $u(x, t = 0) = \chi_D$ when solving one dimensional heat diffusion equation. That means, if we consider the dynamic of $\frac{1}{2}$ level set, the front will move in the normal direction with speed as $\frac{1}{r}$ which is the velocity of convection. In the local polar coordinate at P, we actually have $r = \frac{1}{\kappa}$ which means $\frac{1}{r} = \kappa$. Thus, we can formally see why the MBO method can generate the motion by mean curvature.

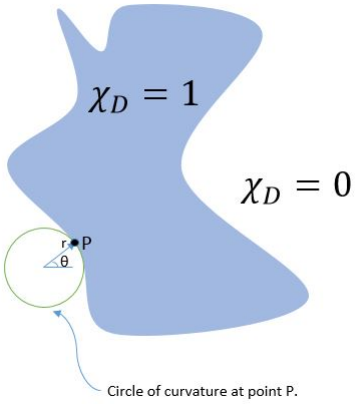


Figure 1.3: Diagram for the local coordinate for the boundary.

The MBO method has been shown to converge to the continuous motion by mean

curvature [9, 15, 36]. The method has attracted much attention and becomes very popular due to its simplicity and unconditional stability, it has been subsequently extended to deal with many other applications. These applications include the multi-phase problems with arbitrary surface tensions [33], the problem of area or volume preserving interface motion [53, 83, 92], image processing [88, 32, 63], problems of anisotropic interface motions [66, 80, 11, 30], and the wetting problem on solid surfaces [95]. Various algorithms and rigorous error analysis have been carried out to refine and extend the original MBO method and related methods for the aforementioned problems (see, for example, [23, 34, 65, 47, 48, 62, 76, 77, 78]). Some adaptive methods are also considered to accelerate this type of method[49] based on non-uniform fast Fourier transform(NUFFT)[26, 42]. Esedoglu and Otto[33] generalize this type of method to multiphase flow with general mobility. Laux et al. [56][55] rigorously proved the convergence of the method proposed in [33].

In this thesis, we first introduce an efficient threshold dynamics method for solid wetting problems in Chapter 2. In solid wetting systems, we first relax the interface energies between different phases by convolutions between indicator functions and a Gaussian kernel. We then develop a simple iterative method (i.e. threshold dynamics method) to minimize the relaxed energy by solving the linearized problem around the k^{th} iteration to get the $k+1^{th}$ iteration at each step. In addition, we develop a quick sort scheme to keep volume preserving. We also prove the energy decaying property which indicates that our algorithm is unconditionally stable. Also, the complexity is $O(NlogN)$ per time step and it is not sensitive to the inhomogeneity or roughness of the solid boundary. Moreover, we do not need to implement the complicated wetting boundary conditions.

In Chapter 3, we rigorously prove the convergence of the threshold dynamics method for solid wetting problems proposed in Chapter 2. The analysis is motivated by the work in [33, 56, 55]. However, the novelty of our proof is that

we consider a case of volume preserving mean-curvature motion described by characteristic function of a domain where part of the boundary is always fixed. Comparing to Section 2 in [55] which establish the convergence proof of a mean-curvature flow with external force, our external force (i.e. the contribution term from D_3 and will be introduced in Section 3.1) depends on h.

The basic step in threshold dynamics method is solving a pure initial value problem of the heat equation with the initial value being the indicator function of some bounded domain. The solution can be represented by the convolution of characteristic function and Gaussian kernel. The convolution can be efficiently evaluated by fast Fourier transform (FFT) on uniform mesh. However, the accuracy of the interface obviously depends on the mesh size. It is still quite expansive if the grid size is quite small (i.e. number of grid points is large). In Chapter 4, we presented an algorithm for the threshold dynamics method to model the interface motion. The algorithm discretizes the physical space only in a neighbourhood of the interface and applies NUFFT to solve the initial value problem of the heat equation. Unlike many grid based methods where the spatial mesh size is required to be of the same order as the time step size, our numerical experiments show that the spatial mesh size can be chosen based upon the accuracy consideration and more or less independent of the time step size for our algorithm.

In Chapter 5, we proposed an efficient threshold dynamics method for the Chan-Vese model for image segmentation. The perimeter terms are approximated by a non-local multi-phase energy constructed based on convolution of the heat kernel with the characteristic functions of regions. The algorithm works by alternating the convolution step with the thresholding step and has the optimal computational complexity of $O(NlogN)$ per iteration. We prove that the iterative algorithm has the property of total energy decay. The numerical results show that the method is stable and the number of iterations before convergence is in-

dependent of the spacial resolution (for a given image). The relative importance of the different effects in the energy functional is studied by tuning the parameter λ. Our numerical results also show that the proposed method is competitive (in terms of efficiency) with many existing methods for image segmentation.

Some conclusions and future works are discussed in Chapter 6.

Chapter 2

An efficient threshold dynamics method for solid wetting problems

Wetting describes how a liquid drop spreads on a solid surface. The most important quantity in wetting is the contact angle between the liquid surface and the solid surface [22]. When the solid surface is homogeneous, the contact angle for a static drop is given by the famous Young's equation:

$$\cos\theta_Y = \frac{\gamma_{SV} - \gamma_{SL}}{\gamma_{LV}}, \tag{2.1}$$

where γ_{SL}, γ_{SV} and γ_{LV} are the solid-liquid, solid-vapor and liquid-vapor surface energy densities, respectively. θ_Y is the so-called Young's angle [96]. Mathematically, Young's equation (2.1) can be derived by minimizing the total energy in the solid-liquid-vapor system. If we ignore gravity, the total energy in the system can be written as

$$\mathcal{E} = \gamma_{LV}|\Sigma_{LV}| + \gamma_{SL}|\Sigma_{SL}| + \gamma_{SV}|\Sigma_{SV}|, \tag{2.2}$$

where Σ_{LV}, Σ_{SL} and Σ_{SV} are respectively the liquid-vapor, solid-liquid and solid-vapor interfaces, and $|\cdot|$ denotes the area of the interfaces. When the solid

surface Γ is a homogeneous planar surface, under the condition that the volume of the drop is fixed, the unique minimizer of the total energy is a domain with a spherical surface in Ω, and the contact angle between the surface and the solid surface Γ is Young's angle θ_Y[91].

The study of wetting and contact angle hysteresis on rough surfaces is of critical importance for many applications and has attracted much interest in the physics and applied mathematics communities [73, 38, 1, 94, 31]. Numerical simulation of wetting on rough surfaces is challenging. One must track the interface motion accurately, as well as deal with complicated boundary shapes and boundary conditions. There are many different types of numerical methods for solving interface and contact line problems, including the front-tracking method [92, 58], the front-capturing method using the level-set function [99], the phase-field methods [29, 10], among others [23].

The generalization of MBO-type methods to the wetting problem where interfaces intersecting the boundary is not straightforward because of a lack of integral representation with a heat kernel for a general domain. In the original MBO scheme, when the interface does not intersect the solid boundary, one can solve the heat equation efficiently on a rectangular domain with a uniform grid using convolution of the heat kernel with the initial condition [77, 76]. The convolution can be evaluated using fast Fourier transform (FFT) at $N \log N$ cost per time step where N is the total number of grid points. One way to generalize MBO-type methods to wetting on solid surfaces is to solve the heat equation with a wetting boundary condition before the volume-preserving thresholding step. However, the usual fast algorithms cannot be applied for this case, especially when the boundary is rough.

In this chapter, we aim to develop an efficient volume-preserving threshold dynamics method for solving wetting problems on rough surfaces. Our method

is based on the approach of Esedoglu-Otto [33]. The key idea is to extend the original domain with a rough boundary to a regular cube and treat the solid part as another phase. In the thresholding step, the solid phase domain remains unchanged. We show that the algorithm has the total interface energy decaying property and our numerical results show that the equilibrium interface satisfies Young's equation near the contact point. The advantage of the method is that it can be implemented efficiently on uniform meshes with a fast algorithm (e.g. FFT) since the computational domain is rectangular and we can simulate wetting on rough boundaries of any shape. We also introduce a fast algorithm for volume preservation based on a quick-sort algorithm and a time refinement scheme to improve the accuracy of the solution at the contact line.

The chapter proceeds as follows. In Section 2.1, we introduce the surface energies of the wetting problem. A direct (but less efficient) MBO-type threshold dynamics method for solving wetting problems is also described. In Section 2.2, we introduce a new threshold dynamics method which is simple, efficient and easy to implement. Several modifications of the method are also discussed. In Section 2.3, we discuss the implementation of the algorithm and perform the accuracy check. We also introduce a quick-sort algorithm for volume preservation and a time refinement technique to improve the accuracy of the contact point motion. In Section 2.4 and Section 2.5, we present numerical examples of wetting on rough surfaces to demonstrate the efficiency of the new method.

2.1 The minimization of surface energies

We consider a wetting problem in a domain $\Omega \in \mathbb{R}^n$, $n = 2, 3$ (see Figure 2.1). The solid surface Γ is part of the domain boundary $\partial\Omega$. Denote the liquid domain by $D_1 \subset \Omega$. For simplicity, we assume that $\partial D_1 \cap \partial\Omega \subset \Gamma$. The volume of the liquid drop is fixed such that $|D_1| = V_0$. We denote $\Sigma_{LV} = \partial D_1 \cap \Omega$,

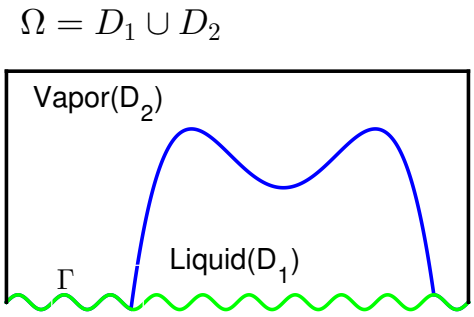


Figure 2.1: Wetting on a rough surface

$\Sigma_{SL} = \partial D_1 \cap \Gamma$ and $\Sigma_{SV} = \Gamma \setminus \partial D_1$ as the liquid-vapor, solid-liquid and solid-vapor interfaces respectively. Then, the equilibrium configuration of the system is obtained by minimizing the total interface energy of the system as follows:

$$\min_{|D_1|=V_0} \mathcal{E}(D_1) = \gamma_{LV}|\partial D_1 \cap \Omega| + \int_{\partial D_1 \cap \Gamma} \gamma_{SL}(x)\mathrm{d}s + \int_{\Gamma \setminus \partial D_1} \gamma_{SV}(x)\mathrm{d}s \qquad (2.3)$$

where the solid boundary Γ is rough and/or chemically inhomogeneous (i.e. $\gamma_{SL}(x)$ and $\gamma_{SV}(x)$ may depend on x). To ensure the problem is well-posed, Young's angle must satisfy $0 < \theta_Y < \pi$. By equation (2.1), this leads to the condition $-1 < \frac{\gamma_{SV}-\gamma_{SL}}{\gamma_{LV}} < 1$. Throughout the chapter, we will assume this condition holds.

To solve problem (2.3) numerically, it is convenient to use a diffuse interface model to approximate the sharp interface energy. Suppose φ is a phase-field function, such that $D_1 = \{\varphi < 0\}$ represents the liquid domain, $\{\varphi > 0\}$ represents the vapor domain and $\Sigma_{LV} = \{\varphi = 0\}$ is the liquid-vapor interface. The total energy (2.2) can be approximated by

$$\mathcal{E}_\varepsilon^{ph}(\varphi) = \int_\Omega \varepsilon|\nabla\varphi|^2 + \frac{f(\varphi)}{\varepsilon}\mathrm{d}\mathbf{x} + \int_\Gamma \gamma(x,\varphi)\mathrm{d}s, \qquad (2.4)$$

where ε is a small parameter representing interface thickness, $f(\varphi) = \frac{(1-\varphi^2)^2}{4}$ is

a double-well function and

$$\gamma(\varphi) = \frac{\tilde{\gamma}_{SV}(x) + \tilde{\gamma}_{SL}(x)}{2} + \frac{\tilde{\gamma}_{SV}(x) - \tilde{\gamma}_{SL}(x)}{4}(3\varphi - \varphi^3).$$

It can be proved that when ε goes to zero, after scaling, the energy in (2.4) converges to that in (2.2) [93]. Therefore, problem (2.3) can be approximated by minimizing the total energy $\mathcal{E}_\varepsilon^{ph}$ under the volume constraint $\int_\Omega (\varphi - 1)/2 \mathrm{d}\mathbf{x} = V_0$.

The H^{-1} gradient flow of the energy functional (2.4) will lead to a Cahn-Hilliard equation with contact angle boundary conditions [20]. Alternatively, the L^2 gradient flow will lead to a modified Allen-Cahn equation:

$$\begin{cases} \varphi_t = \varepsilon \Delta \varphi - \frac{f'(\varphi)}{\varepsilon} + \delta & \text{in } \Omega; \\ \frac{\partial \varphi}{\partial n} + \gamma'(x, \varphi) = 0, & \text{on } \Gamma; \\ \frac{\partial \varphi}{\partial n} = 0, & \text{on } \partial\Omega \setminus \Gamma, \\ \int_\Omega \frac{\varphi - 1}{2} \mathrm{d}\mathbf{x} = V_0. \end{cases} \tag{2.5}$$

Here δ is a Lagrangian multiplier for the volume constraint.

A MBO-type threshold dynamics scheme can be derived easily using a splitting method for (2.5). Assume we have a solution φ^k (characteristic function of a region) at the k-th time step. We can first solve the heat equation

$$\begin{cases} \bar{\varphi}_t = \varepsilon \Delta \bar{\varphi} & \text{in } \Omega. \\ \frac{\partial \bar{\varphi}}{\partial n} + \gamma'(x, \bar{\varphi}) = 0, & \text{on } \Gamma, \\ \frac{\partial \bar{\varphi}}{\partial n} = 0, & \text{on } \partial\Omega \setminus \Gamma, \\ \bar{\varphi}(x, 0) = \bar{\varphi}^k, \end{cases} \tag{2.6}$$

for some time δt_1 and then solve

$$\begin{cases} \varphi_t = -\frac{f'(\varphi)}{\varepsilon} \\ \varphi(x, 0) = \bar{\varphi}(x, \delta t_1) \end{cases} \tag{2.7}$$

for some time δt_2 and set $\varphi^{k+1} = \varphi(x, \delta t_2)$. It is easy to see that when $\delta t_2/\varepsilon$ is large enough, solving the second equation (2.7) is reduced to a thresholding step

$$\varphi(x, \delta t_2) \approx \begin{cases} -1 & \text{if } \varphi(x,0) < 0; \\ 1 & \text{if } \varphi(x,0) > 0, \end{cases} \tag{2.8}$$

which gives a characteristic function φ^{k+1} at the $k+1$ time step. This leads to the following MBO-type scheme for the wetting problem:

A direct MBO threshold dynamics scheme for the wetting problem

Step 0. *Given an initial domain $D_1^0 \subset \Omega$ such that $|D_1^0| = V_0$. Set a tolerance parameter $\varepsilon > 0$.*

Step 1. *For any k, we first solve the heat equation*

$$\begin{cases} \varphi_t = \Delta\varphi & \text{in } \Omega, \\ \frac{\partial \phi}{\partial n} + \gamma'(x, \varphi) = 0, & \text{on } \Gamma, \\ \frac{\partial \varphi}{\partial n} = 0, & \text{on } \partial\Omega \setminus \Gamma, \\ \varphi(x,0) = \chi_{D_1^k}, \end{cases} \tag{2.9}$$

for a time step δt.

Step 2. *Determine a new D_1^{k+1} using thresholding*

$$D_1^{n+1} = \{x : \varphi(x, \delta t) < \frac{1}{2} + \delta\}.$$

Here δ is chosen such that the volume $|D_1^{k+1}| = V_0$.

Step 3. *If $|D_1^k - D_1^{k+1}| < \varepsilon$, stop; otherwise, set $k = k+1$ and go back to Step 1.*

In the original MBO scheme, when the interface does not intersect the solid boundary, one can solve the heat equation efficiently on a uniform grid using

convolution of the heat kernel with the initial condition [77, 76]. The convolution can be evaluated using FFT at $M\log(M)$ cost per time step where M is the total number of grid points. However, when the interface intersects the solid boundary, one must solve the heat equation with the wetting boundary condition as in (2.9). In this case, and in particular for rough boundaries, the usual fast algorithms cannot be applied to solve (2.9). In the next section, we will introduce a new threshold dynamics method.

2.2 A new threshold dynamics method for the wetting problem

In this section, we introduce a new threshold dynamics method motivated by the recent work of Esedoglu and Otto [33]. The main idea is to extend the fluid domain Ω to a larger domain containing the solid phase. In the extended domain, the interface energies between different phases in (2.3) can be approximated by a convolution of characteristic functions and a Guassian kernel (see details below). We then derive a simple scheme to minimize the new energy functional with the constraint that the solid phase does not change and the volume of the liquid phase is preserved. The scheme leads to a new threshold dynamics method for solving the wetting problem.

2.2.1 The representation of interface energies in an extended domain

In the following, we let $D_1, D_2 \subset \Omega$ be the liquid and vapor phases, respectively. Let $\Sigma_{LV} = \partial D_1 \cap \partial D_2$ be the liquid-vapor interface. When $\delta t \ll 1$, the area of

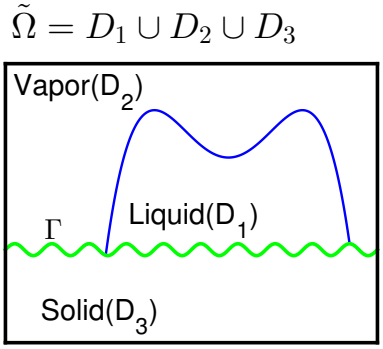


Figure 2.2: Extended computational domain $\tilde{\Omega} = \Omega \cup D_3$

Σ_{LV} can be approximated by (see [2, 67])

$$|\Sigma_{LV}| \approx \frac{1}{\sqrt{\delta t}} \int \chi_{D_1} G_{\delta t} * \chi_{D_2} \mathrm{d}\mathbf{x}, \tag{2.10}$$

where χ_{D_i} is the characteristic function of D_i and

$$G_{\delta t}(\mathbf{x}) = \frac{1}{(4\pi\delta t)^{n/2}} \exp(-\frac{|\mathbf{x}|^2}{4\delta t})$$

is the Gaussian kernel.

In the total energy (2.3), the second and third terms are surface energies defined on the solid surface Γ. They are the solid-liquid interfacial energy term on $\Sigma_{SL} = \partial D_1 \cap \Gamma$ and the solid-vapor interfacial energy term on $\Sigma_{SV} = \partial D_2 \cap \Gamma$. To approximate the two terms using the Gaussian kernel, we extend the domain Ω beyond Γ (see Figure 2.2). The extended domain is $\tilde{\Omega} = \Omega \cup D_3$ where D_3 is the solid region. Then, the solid surface is $\Gamma = \partial\Omega \cap \partial D_3$, the solid-liquid interface is $\Sigma_{SL} = \partial D_1 \cap \partial D_3$ and the solid-vapor interface is $\Sigma_{SV} = \partial D_2 \cap \partial D_3$. Similar to (2.10), the total energy $\mathcal{E}$ in (2.3) can be approximated by

$$\begin{aligned}&\mathcal{E}^{\delta t}(\chi_{D_1}, \chi_{D_2}) = \\ &\frac{\gamma_{LV}}{\sqrt{\delta t}} \int_{\tilde{\Omega}} \chi_{D_1} G_{\delta t} * \chi_{D_2} \mathrm{d}\mathbf{x} + \frac{\gamma_{SL}}{\sqrt{\delta t}} \int_{\tilde{\Omega}} \chi_{D_1} G_{\delta t} * \chi_{D_3} \mathrm{d}\mathbf{x} + \frac{\gamma_{SV}}{\sqrt{\delta t}} \int_{\tilde{\Omega}} \chi_{D_2} G_{\delta t} * \chi_{D_3} \mathrm{d}\mathbf{x}.\end{aligned} \tag{2.11}$$

For simplicity, we assume γ_{SL} and γ_{SV} are constants throughout this section. The analysis and the algorithms can be easily generalized to cases where they are not homogeneous. In section 2.4, we will apply the method to a chemically patterned surface where γ_{SL} and γ_{SV} are piecewise constant functions.

Denote $u_1 = \chi_{D_1}$ and $u_2 = \chi_{D_2}$. We define an admissible set

$$\mathcal{B} = \{(u_1, u_2) \in BV(\Omega) \mid u_i(x) = 0, 1, \text{ and } u_1(x) + u_2(x) = 1, \ a.e.\ x \in \Omega, \int_\Omega u_1 \mathrm{d}\mathbf{x} = V_0\} \tag{2.12}$$

The wetting problem (2.3) can be approximated by

$$\min_{(u_1,u_2)\in\mathcal{B}} \mathcal{E}^{\delta t}(u_1, u_2). \tag{2.13}$$

This is a nonconvex minimization problem since $\mathcal{B}$ is not a convex set. The Γ-convergence of problem (2.13) to (2.3) can be proved in a similar way as in [33].

2.2.2 Derivation of the threshold dynamics method

We will derive the threshold dynamics method for problem (2.13). Notice that the problem is to minimize a concave energy functional defined on a nonconvex admissible set. We first show that it can be relaxed to a problem defined on a convex admissible set. Then we derive a threshold dynamics method for the equivalent problem. The relaxed problem is given by

$$\min_{(u_1,u_2)\in\mathcal{K}} \mathcal{E}^{\delta t}(u_1, u_2). \tag{2.14}$$

where $\mathcal{K}$ is the convex hull of the admissible set $\mathcal{B}$:

$$\mathcal{K} = \{(u_1, u_2) \in BV(\Omega) | 0 \le u_i \le 1, u_1(x) + u_2(x) = 1, \ a.e.\ x \in \Omega, \int_\Omega u_1 \mathrm{d}\mathbf{x} = V_0\}. \tag{2.15}$$

The following lemma shows that the relaxed problem (2.14) is equivalent to the original problem (2.13). For convenience later, we prove the result for a slightly more general problem with an extra linear functional term $\mathcal{L}(u_1, u_2)$.

Lemma 2.2.1. *For any given $\alpha, \beta \geq 0$ and any linear functional $\mathcal{L}(u_1, u_2)$, we have*

$$\min_{(u_1,u_2)\in\mathcal{K}}(\alpha\mathcal{E}^{\delta t}(u_1, u_2) + \beta\mathcal{L}(u_1, u_2)) = \min_{(u_1,u_2)\in\mathcal{B}}(\alpha\mathcal{E}^{\delta t}(u_1, u_2) + \beta\mathcal{L}(u_1, u_2)).$$

Proof. Let $(\tilde{u}_1, \tilde{u}_2) \in \mathcal{K}$ be a minimizer of the functional

$$\alpha\mathcal{E}^{\delta t}(u_1, u_2) + \beta\mathcal{L}(u_1, u_2).$$

Since $\mathcal{B} \subset \mathcal{K}$, we have

$$\begin{aligned}\alpha\mathcal{E}^{\delta t}(\tilde{u}_1, \tilde{u}_2) + \beta\mathcal{L}(\tilde{u}_1, \tilde{u}_2) &= \min_{(u_1,u_2)\in\mathcal{K}} \alpha\mathcal{E}^{\delta t}(u_1, u_2) + \beta\mathcal{L}(u_1, u_2) \\ &\leq \min_{(u_1,u_2)\in\mathcal{B}} \alpha\mathcal{E}^{\delta t}(u_1, u_2) + \beta\mathcal{L}(u_1, u_2).\end{aligned}$$

Therefore, we need only to prove that $(\tilde{u}_1, \tilde{u}_2) \in \mathcal{B}$.

The proof is trivial when $\alpha = 0$, since the minimizer of a linear functional in a convex set must belong to the boundary of the set. When $\alpha > 0$, we prove by contradiction. If $(\tilde{u}_1, \tilde{u}_2) \notin \mathcal{B}$, there is a set $A \in \Omega$ and a constant $0 < C_0 < \frac{1}{2}$, such that $|A| > 0$ and

$$0 < C_0 < \tilde{u}_1(x), \tilde{u}_2(x) < 1 - C_0, \quad \text{for all } x \in A.$$

We divide A into two sets $A = A_1 \cup A_2$ such that $A_1 \cap A_2 = \emptyset$ and $|A_1| = |A_2| = |A|/2$. Denote $u_1^t = \tilde{u}_1 + t\chi_{A_1} - t\chi_{A_2}$ and $u_2^t = \tilde{u}_2 - t\chi_{A_1} + t\chi_{A_2}$. When $0 < t < C_0$, we have $0 < u_1^t, u_2^t < 1$ and

$$u_1^t + u_2^t = \tilde{u}_1 + \tilde{u}_2 = 1, \text{ and } \int_\Omega u_1^t \mathrm{d}\mathbf{x} = \int_\Omega \tilde{u}_1 \mathrm{d}\mathbf{x} = V_0.$$

This implies that $(u_1^t, u_2^t) \in \mathcal{K}$. Furthermore, direct computations give,

$$\begin{aligned}
\frac{d^2}{dt^2}(\alpha\mathcal{E}^{\delta t}(u_1^t, u_2^t) + \beta\mathcal{L}(u_1^t, u_2^t)) &= \frac{1}{\sqrt{\delta t}}\int_{\tilde{\Omega}} \frac{d}{dt}u_1^t G_{\delta t} * \frac{d}{dt}u_2^t \mathrm{d}\mathbf{x} \\
&= \frac{1}{\sqrt{\delta t}}\int_{\tilde{\Omega}} (\chi_{A_1} - \chi_{A_2}) G_{\delta t} * (\chi_{A_2} - \chi_{A_1}) \mathrm{d}\mathbf{x} \\
&= -\frac{1}{\sqrt{\delta t}}\int_{\tilde{\Omega}} (\chi_{A_1} - \chi_{A_2}) G_{\delta t} * (\chi_{A_1} - \chi_{A_2}) \mathrm{d}\mathbf{x} \\
&< 0.
\end{aligned}$$

The functional is concave on the point $(\tilde{u}_1, \tilde{u}_2)$. Thus, $(\tilde{u}_1, \tilde{u}_2)$ cannot be a minimizer of the functional. This contradicts the assumption. □

The above lemma implies that we can solve the relaxed problem (2.14) instead of the original one (2.13). In the following, we show that the problem can be solved iteratively using a thresholding method.

Suppose we solve problem (2.14) using an iterative method. In the k^{th} step, we have an approximated solution (u_1^k, u_2^k). The energy functional $\mathcal{E}^{\delta t}(u_1, u_2)$ can be linearized near the point (u_1^k, u_2^k) as follows:

$$\mathcal{E}^{\delta t}(u_1, u_2) \approx \mathcal{E}^{\delta t}(u_1^k, u_2^k) + \hat{\mathcal{L}}(u_1 - u_1^k, u_2 - u_2^k, u_1^k, u_2^k) + h.o.t.$$

with

$$\begin{aligned}
&\hat{\mathcal{L}}(u_1, u_2, u_1^k, u_2^k) = \\
&\frac{1}{\sqrt{\delta t}}\left(\int_{\tilde{\Omega}} u_1 G_{\delta t} * (\gamma_{LV} u_2^k + \gamma_{SL}\chi_{D_3}) \mathrm{d}\mathbf{x} + \int_{\tilde{\Omega}} u_2 G_{\delta t} * (\gamma_{LV} u_1^k + \gamma_{SV}\chi_{D_3}) \mathrm{d}\mathbf{x}\right).
\end{aligned} \tag{2.16}$$

Then we minimize the linearized functional

$$\min_{(u_1,u_2)\in\mathcal{K}} \hat{\mathcal{L}}(u_1, u_2, u_1^k, u_2^k) \tag{2.17}$$

and set the solution to (u_1^{k+1}, u_2^{k+1}). By Lemma 2.2.1, the solution to (2.17) is in $\mathcal{B}$. In other words, u_1^{k+1} and u_2^{k+1} are characteristic functions of some proper sets D_1^{k+1} and D_2^{k+1} such that $|D_1^{k+1}| = V_0$.

The following lemma shows that the minimizing problem (2.17) is solved via a simple thresholding approach.

Lemma 2.2.2. *Denote*

$$\phi_1 = \frac{1}{\sqrt{\delta t}} G_{\delta t} * (\gamma_{LV} u_2^k + \gamma_{SL} \chi_{D_3}),\ \phi_2 = \frac{1}{\sqrt{\delta t}} G_{\delta t} * (\gamma_{LV} u_1^k + \gamma_{SV} \chi_{D_3}). \quad (2.18)$$

Let

$$D_1^{k+1} = \{x \in \Omega |\ \phi_1 < \phi_2 + \delta\} \quad (2.19)$$

for some δ such that $|D_1^{k+1}| = V_0$. Define $D_2^{k+1} = \Omega \setminus D_1^{k+1}$. Then $(u_1^{k+1}, u_2^{k+1}) = (\chi_{D_1^{k+1}}, \chi_{D_2^{k+1}})$ is a solution to (2.17).

Proof. By Lemma 2.2.1, we need only to prove

$$\hat{\mathcal{L}}(u_1^{k+1}, u_2^{k+1}, u_1^k, u_2^k) \leq \hat{\mathcal{L}}(u_1, u_2, u_1^k, u_2^k), \quad (2.20)$$

for all $(u_1, u_2) \in \mathcal{B}$.

For each $(u_1, u_2) \in \mathcal{B}$, we know $u_1 = \chi_{\hat{D}_1}$ and $u_2 = \chi_{\hat{D}_2}$ for some open sets $\hat{D}_1$, $\hat{D}_2$ in Ω, such that $\hat{D}_1 \cap \hat{D}_2 = \emptyset$, $\hat{D}_1 \cup \hat{D}_2 = \Omega$ and $|\hat{D}_1| = V_0$. Let $A_1 = \hat{D}_1 \setminus D_1^{k+1} = D_2^{k+1} \setminus \hat{D}_2$ and $A_2 = \hat{D}_2 \setminus D_2^{k+1} = D_1^{k+1} \setminus \hat{D}_1$. We must have $|A_1| = |A_2|$ due to the volume conservation property. Since $A_1 \subset D_2^{k+1}$, we have

$$\phi_1(x) \geq \phi_2(x) + \delta, \quad \forall x \in A_1.$$

Similarly, since $A_2 \in D_1^{k+1}$, we have

$$\phi_1(x) < \phi_2(x) + \delta, \quad \forall x \in A_2.$$

Therefore, we have

$$\begin{aligned}
&\hat{\mathcal{L}}(u_1^{k+1}, u_2^{k+1}, u_1^k, u_2^k) - \hat{\mathcal{L}}(u_1, u_2, u_1^k, u_2^k) \\
=& \int_{\tilde{\Omega}} (u_1^{k+1} - u_1)\phi_1 + (u_2^{k+1} - u_2)\phi_2 \mathrm{d}\mathbf{x} \\
=& -\int_{A_1} \phi_1 \mathrm{d}\mathbf{x} + \int_{A_2} \phi_1 \mathrm{d}\mathbf{x} - \int_{A_2} \phi_2 \mathrm{d}\mathbf{x} + \int_{A_1} \phi_2 \mathrm{d}\mathbf{x} \\
=& \int_{A_1} (\phi_2 - \phi_1)\mathrm{d}\mathbf{x} + \int_{A_2} (\phi_1 - \phi_2)\mathrm{d}\mathbf{x} \\
\leq& -\delta \int_{A_1} \mathrm{d}\mathbf{x} + \delta \int_{A_2} \mathrm{d}\mathbf{x} = 0.
\end{aligned}$$

□

We are led to the following threshold dynamics algorithm:

Algorithm I:

Step 0. *Given initial $D_1^0, D_2^0 \subset \Omega$, such that $D_1^0 \cap D_2^0 = \emptyset$, $D_1^0 \cup D_2^0 = \Omega$ and $|D_1^0| = V_0$. Set a tolerance parameter $\varepsilon > 0$.*

Step 1. *For given sets (D_1^k, D_2^k), we define two functions*

$$\phi_1 = \frac{1}{\sqrt{\delta t}} G_{\delta t} * (\gamma_{LV}\chi_{D_2^k} + \gamma_{SL}\chi_{D_3}),\ \phi_2 = \frac{1}{\sqrt{\delta t}} G_{\delta t} * (\gamma_{LV}\chi_{D_1^k} + \gamma_{SV}\chi_{D_3}). \tag{2.21}$$

Step 2. *Find a δ such that the set*

$$\tilde{D}_1^\delta = \{x \in \Omega | \phi_1 < \phi_2 + \delta.\} \tag{2.22}$$

satisfies $|\tilde{D}_1^\delta| = V_0$. Denote $D_1^{k+1} = \tilde{D}_1^\delta$ and $D_2^{k+1} = \Omega \setminus D_1^{k+1}$.

Step 3. *If $|D_1^k - D_1^{k+1}| \leq \varepsilon$, stop; otherwise, go back to Step 1.*

Remark 2.2.1. *The method is simple and easy to implement.*

(1) We can always extend Ω to a cubic domain $\tilde{\Omega}$, since the only constraints on the extension are $D_1 \in \tilde{\Omega}$ and $|D_1| = V_0$. For the cube domain, the convolution in (3.2) can be computed by fast algorithms (e.g. the FFT).

(2) To keep the volume of the liquid phase unchanged, we need to find a proper δ in Step 2. This can be done by using an iterative method (such as bisection method), as shown in [78] for mean curvature flow. In the next section, we will give a simpler and more efficient technique to determine δ.

(3) The above derivation of the thresholding method for the wetting problem can be easily generalized to a multiphase system with wetting boundary conditions, e.g. the three-phase system [81], in the same spirit of Esedoglu and Otto [33].

2.2.3 A simplified algorithm for the two-phase problem

For the two-phase problem, Algorithm I can be simplified as follows. Noticing that $u_1 + u_2 = 1$ in Ω, we actually have only one unknown u_1 in (2.14). Define

$$\mathcal{K}_1 = \{u \in BV(\Omega) | 0 \le u \le 1, \ a.e.\ x \in \Omega, \int_\Omega u \mathrm{d}\mathbf{x} = V_0\}.$$

It is easy to see that (2.14) can be rewritten as

$$\begin{aligned}\min_{u_1 \in \mathcal{K}_1} \tilde{\mathcal{E}}^{\delta t}(u_1) = &-\gamma_{LV} \int_{\tilde{\Omega}} u_1 G_{\delta t} * u_1 \mathrm{d}\mathbf{x} + \gamma_{LV} \int_{\tilde{\Omega}} u_1 G_\delta * \chi_\Omega \mathrm{d}\mathbf{x} \\ &+ \int_{\tilde{\Omega}} (\gamma_{SL} - \gamma_{SV}) u_1 G_{\delta t} * \chi_{D_3} \mathrm{d}\mathbf{x} + \int_{\tilde{\Omega}} \gamma_{SV} \chi_\Omega G_{\delta t} * \chi_{D_3} \mathrm{d}\mathbf{x}. \quad (2.23)\end{aligned}$$

Suppose we solve the problem using an iterative method. For any given u_1^k, we could linearize the functional as

$$\tilde{\mathcal{E}}^{\delta t}(u_1) = \tilde{\mathcal{E}}^{\delta t}(u_1^k) + \tilde{\mathcal{L}}(u - u_1^k, u_1^k) + h.o.t.$$

with

$$\begin{aligned}\tilde{\mathcal{L}}(u,u_1^k) =& -2\gamma_{LV}\int_{\tilde{\Omega}} u_1 G_{\delta t} * u_1^k \mathrm{d}\mathbf{x} + \gamma_{LV}\int_{\tilde{\Omega}} u_1 G_\delta * \chi_\Omega \mathrm{d}\mathbf{x} \\ &+ \int_{\tilde{\Omega}} (\gamma_{SL}-\gamma_{SV}) u_1 G_{\delta t} * \chi_{D_3} \mathrm{d}\mathbf{x} \\ =&\gamma_{LV}\int_{\tilde{\Omega}} u_1 G_{\delta t} * (u_2^k - u_1^k)\mathrm{d}\mathbf{x} + \int_{\tilde{\Omega}} (\gamma_{SL}-\gamma_{SV}) u_1 G_{\delta t} * \chi_{D_3}\mathrm{d}\mathbf{x} \\ =&\gamma_{LV}\int_{\tilde{\Omega}} u_1 G_{\delta t} * (u_2^k - u_1^k - \cos\theta_Y \chi_{D_3})\mathrm{d}\mathbf{x}. \end{aligned} \tag{2.24}$$

Here we use Young's equation $\gamma_{LV}\cos\theta_Y = \gamma_{SV} - \gamma_{SL}$.

As in the previous subsection, for the linearized functional (2.24), we can prove the following result. The proof is similar to that for Lemma (2.2.2).

Lemma 2.2.3. *Suppose $u_1^k = \chi_{D_1^k}$ for some sets $D_1^k \subset \Omega$ and $D_2^k = \Omega \setminus D_1^k$. Denote*

$$\phi = \frac{\gamma_{LV}}{\sqrt{\delta t}} G_{\delta t} * (\chi_{D_2^k} - \chi_{D_1^k} - \cos(\theta_Y)\chi_{D_3}),$$

Let $\tilde{D}_1^\delta = \{x \in \Omega \mid \phi < \delta\}$, with some δ such that $|D_1^{k+1}| = V_0$. Then $u_1^{k+1} = \chi_{D_1^{k+1}}$ is a minimizer of $\tilde{\mathcal{L}}(u,u_1^k)$ in $\mathcal{K}_1$.

This leads to the following algorithm.

Algorithm II:

Step 0. *Given initial $D_1^0 \subset \Omega$, such that $|D_1^0| = V_0$. Set a tolerance parameter $\varepsilon > 0$.*

Step 1. *For given set D_1^k, set $D_2^k = \Omega \setminus D_1^k$, define a function*

$$\phi = \frac{\gamma_{LV}}{\sqrt{\delta t}} G_{\delta t} * (\chi_{D_2^k} - \chi_{D_1^k} - \cos(\theta_Y)\chi_{D_3}). \tag{2.25}$$

Step 2. *Find a $\delta \in (-1,1)$, so that the set*

$$\tilde{D}_1^\delta = \{x \in \Omega \mid \phi < \delta.\} \tag{2.26}$$

satisfying $|\tilde{D}_1^\delta| = V_0$. Denote $D_1^{k+1} = \tilde{D}_1^\delta$.

Step 3. *If* $|D_1^k - D_1^{k+1}| \leq \varepsilon$*, stop; otherwise, go back to Step 1.*

The following proposition shows that Algorithm I and Algorithm II are equivalent.

Proposition 2.2.1. *For any domain* $(D_1^k, D_2^k) \in \mathcal{B}$*, after one iteration, Algorithm I and Algorithm II generate the same* (D_1^{k+1}, D_2^{k+1})*.*

Proof. We need only consider the thresholding equations (3.3) and (2.26). Direct computations give

$$\begin{aligned}
\phi_1 - \phi_2 =& \frac{1}{\sqrt{\delta t}} G_{\delta t} * (\gamma_{LV}\chi_{D_2^k} + \gamma_{SL}\chi_{D_3}) - \frac{1}{\sqrt{\delta t}} G_{\delta t} * (\gamma_{LV}\chi_{D_1^k} + \gamma_{SV}\chi_{D_3}) \\
=& \frac{1}{\sqrt{\delta t}} G_{\delta t} * \Big(\gamma_{LV}(\chi_{D_2^k} - \chi_{D_1^k}) + (\gamma_{SL} - \gamma_{SV})\chi_{D_3}\Big) \\
=& \frac{\gamma_{LV}}{\sqrt{\delta t}} G_{\delta t} * (\chi_{D_2^k} - \chi_{D_1^k} - \cos\theta_Y \chi_{D_3}) = \phi.
\end{aligned}$$

In the last equation, we used Young's equation. Therefore, the thresholding equation (3.3) is equivalent to the thresholding equation (2.26). □

2.2.4 Stability analysis

In this subsection, we will show that the two algorithms above are stable, in the sense that the total energy of $\mathcal{E}^{\delta t}$ always decreases in the algorithm for any $\delta t > 0$. We have the following theorem.

Theorem 2.2.1. *Denote* $(u_1^k, u_2^k) = (\chi_{D_1^k}, \chi_{D_2^k})$*,* $k = 0, 1, 2, ...$*, obtained in Algorithm I (or Algorithm II). We have*

$$\mathcal{E}^{\delta t}(u_1^{k+1}, u_2^{k+1}) \leq \mathcal{E}^{\delta t}(u_1^k, u_2^k), \tag{2.27}$$

for all $\delta t > 0$*.*

Proof. By Proposition 2.2.1, we need only to prove the theorem for Algorithm I. By the definition of the linearization $\hat{\mathcal{L}}$ and Lemma 2.2.2, we know that

$$\begin{aligned}
&\mathcal{E}^{\delta t}(u_1^k, u_2^k) + \frac{\gamma_{LV}}{\sqrt{\delta t}} \int_{\tilde{\Omega}} u_1^k G_{\delta t} * u_2^k \mathrm{d}\mathbf{x} = \hat{\mathcal{L}}(u_1^k, u_2^k, u_1^k, u_2^k) \\
&\geq \mathcal{L}(u_1^{k+1}, u_2^{k+1}, u_1^k, u_2^k) = \mathcal{E}^{\delta t}(u_1^{k+1}, u_2^{k+1}) \\
&\quad + \frac{\gamma_{LV}}{\sqrt{\delta t}} \left(\int_{\tilde{\Omega}} u_1^{k+1} G_{\delta t} * u_2^k \mathrm{d}\mathbf{x} + \int_{\tilde{\Omega}} u_2^{k+1} G_{\delta t} * u_1^k \mathrm{d}\mathbf{x} - \int_{\tilde{\Omega}} u_1^{k+1} G_{\delta t} * u_2^{k+1} \mathrm{d}\mathbf{x} \right).
\end{aligned}$$

This leads to

$$\mathcal{E}^{\delta t}(u_1^k, u_2^k) \geq \mathcal{E}^{\delta t}(u_1^{k+1}, u_2^{k+1}) + I, \tag{2.28}$$

with

$$\begin{aligned}
I = \frac{\gamma_{LV}}{\sqrt{\delta t}} \Big(&\int_{\tilde{\Omega}} u_1^{k+1} G_{\delta t} * u_2^k \mathrm{d}\mathbf{x} + \int_{\tilde{\Omega}} u_2^{k+1} G_{\delta t} * u_1^k \mathrm{d}\mathbf{x} \\
&- \int_{\tilde{\Omega}} u_1^{k+1} G_{\delta t} * u_2^{k+1} \mathrm{d}\mathbf{x} - \int_{\tilde{\Omega}} u_1^k G_{\delta t} * u_2^k \mathrm{d}\mathbf{x} \Big) \\
= -\frac{\gamma_{LV}}{\sqrt{\delta t}} &\int_{\tilde{\Omega}} (u_1^{k+1} - u_1^k) G_{\delta t} * (u_2^{k+1} - u_2^k) \mathrm{d}\mathbf{x}.
\end{aligned}$$

By the fact that $u_1^k + u_2^k = u_1^{k+1} + u_2^{k+1}$, we have

$$I = \frac{\gamma_{LV}}{\sqrt{\delta t}} \int_{\tilde{\Omega}} (u_1^{k+1} - u_1^k) G_{\delta t} * (u_1^{k+1} - u_1^k) \mathrm{d}\mathbf{x} \geq 0.$$

This inequality together with (2.28) implies (2.27). □

2.3 Numerical implementation and accuracy check

In this section, we will introduce several techniques used to implement the algorithm efficiently.

2.3.1 Calculation of convolution

In Algorithm I, we need to calculate the two convolutions $G_{\delta t} * (\gamma_{LV} \chi_{D_2^k} + \gamma_{SL} \chi_{D_3})$ and $G_{\delta t} * (\gamma_{LV} \chi_{D_1^k} + \gamma_{SV} \chi_{D_3})$ in an extended domain $\tilde{\Omega}$ which we can always choose

to be a rectangular domain. We can use FFT to efficiently calculate the convolutions when the functions are periodic. In our simulation, the characteristic functions (e.g. $\gamma_{LV}\chi_{D_2^k} + \gamma_{SL}\chi_{D_3}$) are not periodic. To calculate convolutions for non-periodic functions, we can further extend the domain by reflection so that the functions are periodic in the extended domain. However, the heat kernel $G_{\delta t}$ decays exponentially and is negligible when $|x| > 10\sqrt{\delta t}$. When we calculate the convolution, each target point will only be affected by a few neighboring points. Hence, if we apply the FFT without extending the computational domain, we will only have some error near the boundary of the computational domain (See Figure 2.3). When the dynamic interface is far away from the boundary of the computational domain, the solutions calculated with or without the domain extension are the same, after the thresholding step. Therefore, in our calculation, we always directly apply the FFT without extending the computational domain.

2.3.2 A quicksort algorithm for volume preservation

In Step 2 of Algorithm I, we need to enforce volume preservation. This is achieved by shifting the thresholding level by δ as in (3.3). The usual way to find δ is by some iteration method (e.g. bisection method, Newton method, fixed point iteration, see [78]). However, these iterative methods may be sensitive to the initial guess. In this section, we will introduce a direct and more efficient algorithm to find a proper δ. If we consider a uniform mesh (in two dimensions) and denote the mesh size by dx, the volume of a domain can be approximated by $V_0 \approx M \times dx^2$ (with first-order accuracy). To maintain the same volume after thresholding, what we need to do in Step 2 is to find a threshold δ such that there are M grid values of $\phi_1 - \phi_2 = g$ which are less than δ. Since we have the values of $\phi_1 - \phi_2$ at each grid point, we can use the quicksort algorithm (available in Matlab) [46] to sort the values in ascending order into a list $\mathcal{S} = \{g_1, g_2, ..., g_M, g_{M+1}, ...\}$. We then take the average of the M^{th} value and $(M+1)^{th}$ value in the ordered list $\mathcal{S}$,

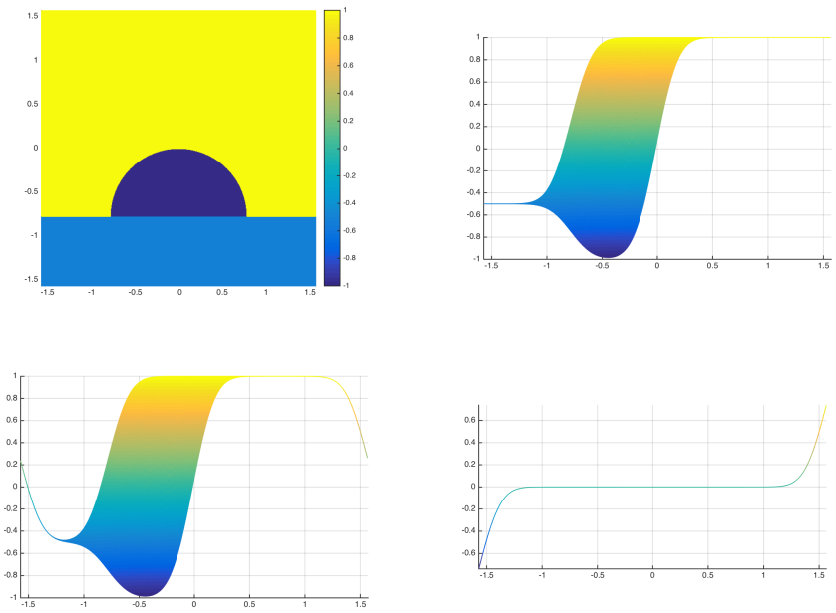


Figure 2.3: From left to right, top to bottom: 1. The initial condition defined by the characteristic function of the domains. 2. The X-Z plane of the convolution between the heat kernel and the initial condition calculated by extending the domain by reflection (i.e. extending $[-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$ to $[-\frac{\pi}{2}, \frac{3\pi}{2}] \times [-\frac{\pi}{2}, \frac{3\pi}{2}]$). 3. The X-Z plane of the convolution between the heat kernel and the initial condition calculated without extending the domain. 4. The difference between the second figure and the third figure.

i.e. $\delta = \frac{g_M + g_{M+1}}{2}$ to be the threshold value δ. A simple example to demonstrate this fast scheme is shown in Figure 2.4. The scheme is summarized as follows:

A quicksort scheme for volume conservation

Step 0. *Set V_0 as the volume to be preserved and M as the integer part of V_0/dx^2.*

Step 1. *Use a quicksort algorithm to sort $g = \phi_1 - \phi_2$, which is defined in Step 2 in Algorithm 2, in ascending order into a list $\mathcal{S} =$*

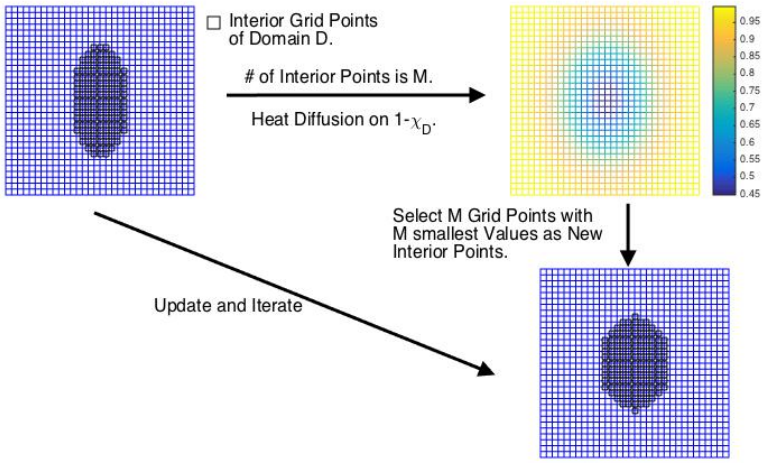


Figure 2.4: An example to demonstrate our new scheme for volume preservation. Initially, there are M interior points. After convolution, we select M grid points with the M lowest values as the new interior points.

$$\{g_1, g_2, ..., g_M, g_{M+1}, ...\}.$$

Step 2. *Set* $\delta = \frac{g_M + g_{M+1}}{2}$.

In summary, the computational complexity involved in finding δ is $O(N \log(N))$ when the quicksort algorithm is used. It is straightforward to see that this scheme will give the same δ as the iterative scheme proposed by Ruuth [78] (with first order accuracy). However, our scheme costs much less computationally.

2.3.3 Accuracy check of Algorithm I

In this subsection, we will check the accuracy of Algorithm I. We first consider an example of motion of two circles. One circle is centered at $(0.35, 0.35)$ with radius 0.2 and the other is centered at $(0.7, 0.7)$ with radius 0.15 (see in Figure 2.5). The volume-preserving mean curvature flow is governed by the interface

motion law $v_n = \kappa - \kappa_a$, where v_n represents the normal velocity of the interface, κ is the curvature and κ_a is the average curvature of the interface. By this motion, the larger circle will grow in volume while the smaller circle will shrink gradually. The exact solution can be calculated and the area enclosed by the smaller circle after a time $t = 0.02$ is given by 0.0445079 [78]. Using Algorithm I, we compute numerically the motion of the two circles and compare the results with the exact solution. Table 2.1 shows the relative volume error as well as relative L^∞ error compared with the exact solution at $t = 0.02$ for different $\delta t = 0.002, 0.001, 0.0005, 0.00025, 0.000125$ with the same spatial resolution (16384×16384). The results indicate the first-order accuracy of our scheme.

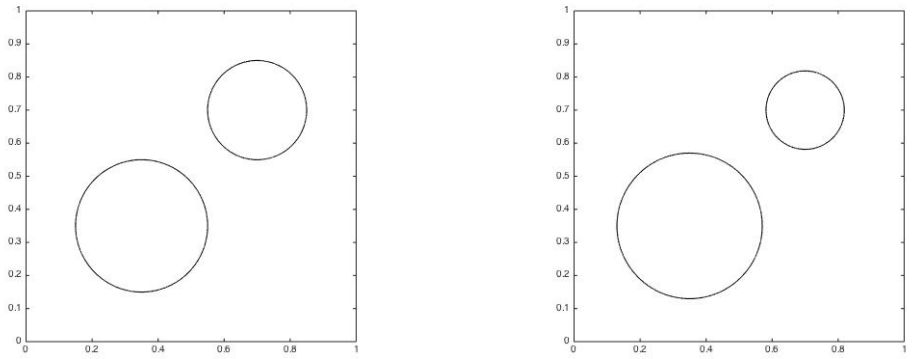


Figure 2.5: The two circles at $t = 0$ (left) and $t = 0.02$(right).

We next consider the motion of two semi-circles on the solid surface. One is centered at $(0.3, 0.25)$ with radius 0.2 and the other one is centered at $(0.8, 0.25)$ with radius 0.15 (See Figure 2.6). In this problem, we set Young's angle to $\pi/2$. Then the wetting boundary condition in (2.5) will reduce to a homogeneous Neumann boundary condition. One can obtain the same motion as that of two full circles by symmetric reflection. Therefore, the exact solution can also be calculated and the area enclosed by the smaller semi-circle after a time $t = 0.02$ is 0.022254 (half of the volume of the smaller circle in the previous example).

Table 2.1: Accuracy check of Algorithm I for the two circle motion

δt	Relative Vol. Err.	Conv. Rate	Relative L^∞ Err.	Conv. Rate
0.002	-0.075	–	0.061	–
0.001	-0.035	1.14	0.019	2.21
0.0005	-0.016	1.19	0.0085	1.24
0.00025	-0.0080	1.00	0.0043	0.98
0.000125	-0.0039	1.05	0.0022	0.95

Again, using Algorithm I, we compute numerically the motion of the two circles and compare the results with the exact solution. Table 2.2 shows the relative volume error as well as relative L^∞ error compared with the exact solution at $t = 0.02$ for different $\delta t = 0.002, 0.001, 0.0005, 0.00025, 0.000125, 0.0000625$ with the same spatial resolution (16384×16384). The results indicate a half-order convergence. This is typical for multi-phase problems with a junction.

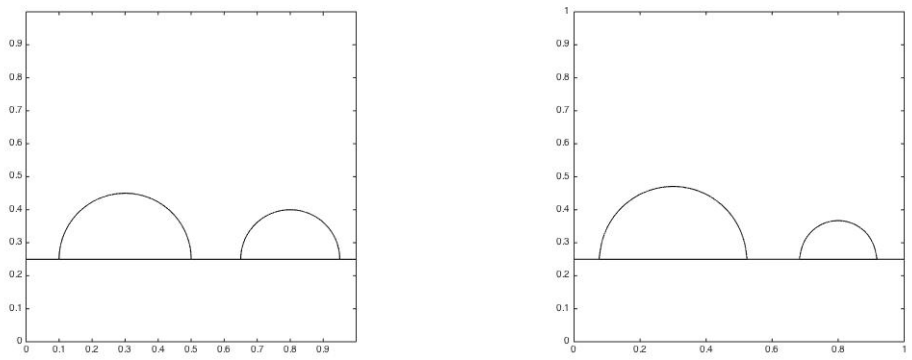


Figure 2.6: The two semi-circles at $t = 0$ (left) and $t = 0.02$(right).

Table 2.2: Accuracy check of Algorithm I for the motion of two semi-circles on solid boundary.

δt	Relative Vol. Err.	Conv. Rate	Relative L^∞ Err.	Conv. Rate
0.002	-0.32	–	0.25	–
0.001	-0.15	1.14	0.092	1.73
0.0005	-0.076	0.98	0.049	0.87
0.00025	-0.045	0.70	0.031	0.58
0.000125	-0.028	0.62	0.020	0.53
0.0000625	-0.017	0.61	0.013	0.54

2.3.4 A time refinement scheme for contact point motion

For any given space mesh, the only parameter in Algorithm I is the time step δt. According to Merriman, Bence and Osher [64], the choice of δt should meet two requirements. The first one is that δt should be small enough so that the approximation of the energy is reasonably accurate. The second is that δt should also be large enough so that the boundary curve moves at least one grid cell on the spatial grid (otherwise the interface would not move after the thresholding step), that is, $\delta t \gg \frac{\delta x}{\kappa}$ where κ is the average curvature and δx is the space mesh size. Since we have volume conservation, the interface will eventually become circular with a constant curvature. Therefore, for a given space mesh size δx, there is a δt threshold below which the interface will not move. Therefore time step refinement beyond this threshold will not improve the accuracy of the interface location. However, when the interface intersects the solid boundary, the motion of the contact point follows different dynamics and is controlled by the Young stress $f = \gamma_{LV}(\cos\theta - \cos\theta_Y)$. This may lead to a different time scale (and a different time step constraint). Numerical results show that time step

refinement improves the accuracy near the contact point. Hence, we propose a time refinement scheme to minimize the interfacial energy. The idea is to first use a proper (large enough) time step δt so that the evolution of the interface reaches equilibrium. We then improve the contact point accuracy by repeatedly halving the time step δt until the difference between the solutions of succeeding steps is within a tolerance ε_1.

Modified Algorithm I

Step 0. *Given initial* $D_1^0, D_2^0 \subset \Omega$, *such that* $D_1^0 \cap D_2^0 = \emptyset$, $D_1^0 \cup D_2^0 = \Omega$ *and* $|D_1^0| = V_0$. *Set* $D_1^* = D_1^0$. *Set a tolerance parameter* $\varepsilon > 0$.

Step 1. *For given set* (D_1^k, D_2^k), *we define two functions*

$$\phi_1 = \frac{1}{\sqrt{\delta t}} G_{\delta t} * (\gamma_{LV}\chi_{D_2^k} + \gamma_{SL}\chi_{D_3}),\ \phi_2 = \frac{1}{\sqrt{\delta t}} G_{\delta t} * (\gamma_{LV}\chi_{D_1^k} + \gamma_{SV}\chi_{D_3}). \tag{2.29}$$

Step 2. *Find a constant* δ *to ensure volume preservation using the quick-sort algorithm in section 2.3.2, so that the set*

$$\tilde{D}_1^\delta = \{x \in \Omega | \phi_1 < \phi_2 + \delta.\} \tag{2.30}$$

satisfying $|\tilde{D}_1^\delta| \approx V_0$. *Denote* $D_1^{k+1} = \tilde{D}_1^\delta$, $D_2^{k+1} = \Omega \setminus D_1^{k+1}$.

Step 3. *IF* $|D_1^k - D_1^{k+1}| \leq \varepsilon$,

if $|D_1^* - D_1^{k+1}| \geq \varepsilon$,*set* $\delta t = \frac{\delta t}{2}$, $D_1^* = D_1^{k+1}$, *and go back to step 1.*

else, set $D_1^* = D_1^{k+1}$ *and stop.*

endif

ELSE, go back to step 1.

ENDIF

2.3.5 Accuracy check of the Modified Algorithm I

To check the accuracy of the Modified Algorithm I described in Section 2.3.4, we consider a two-dimensional drop spreading on a solid surface. The equilibrium state is a circular arc with Young's angle when the minimum of the total interfacial energy is reached. In our experiment, the initial liquid phase is given by a semi-circle centered at $(0, -\frac{\pi}{4})$ with radius $\frac{\pi}{4}$. So the volume of the drop is $\frac{\pi^3}{32}$. We set three surface tensions as $\gamma_{LV} = 1, \gamma_{LS} = 1$ and $\gamma_{SV} = 1 + \sqrt{3}/2$, which gives Young's angle $\frac{\pi}{3}$. In this case, the exact equilibrium state can be computed explicitly.

In Figure 2.7, Figure 2.8 and Figure 2.9, we show the errors of solutions (characteristic functions) computed by both Algorithm I and the Modified Algorithm I, compared with the exact solution (the characteristic function of the exact equilibrium state) which shows the location error of the interface. It is obvious that the errors near contact points are much larger than those at other places of the interfaces. However, after time step refinement, the Modified Algorithm 1 gives much improved results. Figure 2.10 compares well the numerical solution and exact solution at the equilibrium.

We then check the accuracy of the algorithms via calculating the convergence rate of the L^1 error and L^∞ error with respect to the mesh refinement. Table 2.3 shows the L^1 errors of both schemes. Again the Modified Algorithm I gives much better results. The results also show that the convergence rate for L^1 error of our algorithm is of first order. Table 2.4 shows the L^∞ errors of both schemes. Again the Modified Algorithm I gives superior results. The example shows that the time refinement scheme improves the accuracy dramatically. But this does not necessarily mean that the convergence order is also improved, especially for the L^1 error.

Table 2.3: Accuracy Check in L^1 norm

Grid points	L^1 Err.	Conv. rate	L^1 Err. with time refinement	Conv. rate
128× 128	0.1473	-	0.0515	–
256× 256	0.0482	2.06	0.0271	0.90
512× 512	0.0200	1.41	0.0109	1.49
1024 × 1024	0.0116	0.72	0.0054	1.02

Table 2.4: Accuracy Check in L^∞ norm

Grid points	L^∞ Err.	Conv. rate	L^∞ Err. with time refinement	Conv. rate
128× 128	0.1473	-	0.0982	–
256× 256	0.0831	0.77	0.0585	0.68
512× 512	0.0552	0.51	0.0307	0.91
1024 × 1024	0.0333	0.66	0.0149	1.1

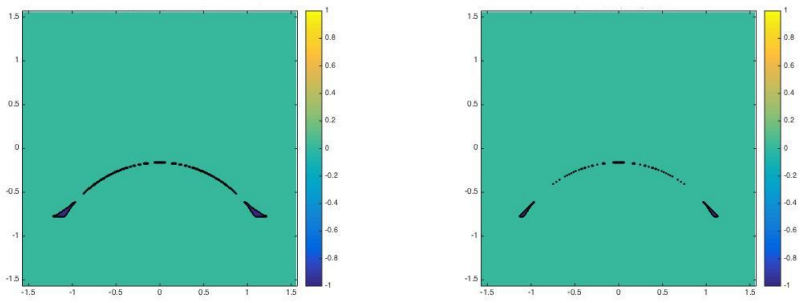


Figure 2.7: Left: 256×256 grid points, $\delta t = 2dx$ without refinement in time. Right: 256×256 grid points, $\delta t = 2dx$ initially with refinement in time, $\epsilon = 1.0e^{-10}$

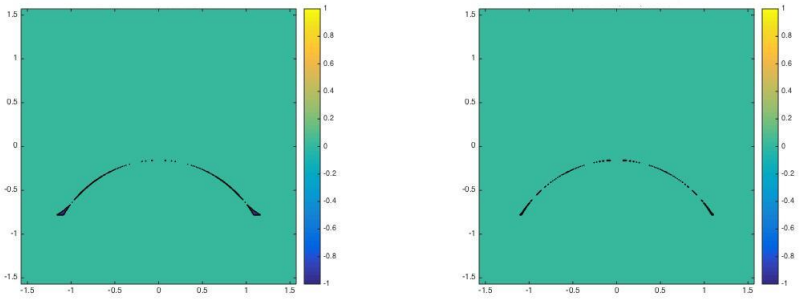


Figure 2.8: Left: 512×512 grid points, $\delta t = 2dx$ without refinement in time. Right: 512×512 grid points, $\delta t = 2dx$ initially with refinement in time, $\epsilon = 1.0e^{-10}$

2.4 A drop spreading on a chemically pattern solid surface

We first study the hysteresis behavior of a drop spreading on a chemically patterned surface. We consider the quasi-static spreading of a drop. To simulate the hysteresis process. we need to increase or decrease the volume of the drop gradually. In each step, we need to compute the equilibrium state of the drop after liquid is added or extracted, which is very computationally demanding. We

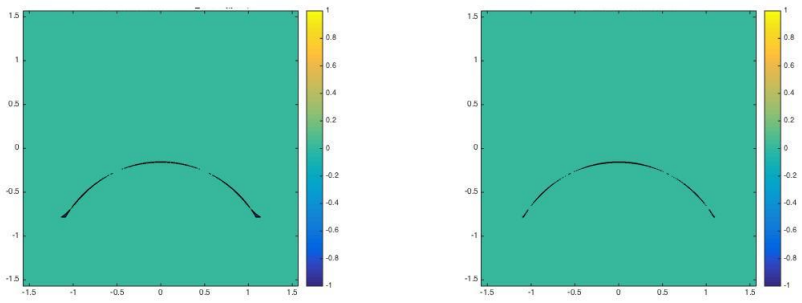


Figure 2.9: Left: 1024×1024 grid points, $\delta t = 2dx$ without refinement in time. Right: 1024×1024 grid points, $\delta t = 2dx$ initially with refinement in time, $\epsilon = 1.0e^{-10}$

show that our threshold dynamics method can simulate the process efficiently.

We assume that the surface is periodically patterned in the interval $(-\pi/2, \pi/2)$ and the interval is divided into $2k+1$ periods with an equal partition of two materials $\mathcal{A}, \mathcal{B}$ away from the center. The center part is occupied by the material $\mathcal{B}$ (See Figure 2.11). Assume $\theta_{\mathcal{A}}$, $\theta_{\mathcal{B}}$ are Young's angles for materials $\mathcal{A}$ and $\mathcal{B}$ respectively. r is the initial radius of a semi-circle on the surface and ΔV is the volume we add to the drop each time. The procedure for explicitly calculating the change in contact angle and position of contact points with respect to the volume for simple two-phase systems on a chemically patterned surface is given in [93].

To implement the Modified Algorithm I, we need to divide our solid region into two parts D_3 and D_4 representing material $\mathcal{A}$ and material $\mathcal{B}$ with different surface tensions, respectively (as shown in Figure 2.11), and modify the original $\gamma_{SL}\chi_{D_3}$ and $\gamma_{SV}\chi_{D_3}$ to $\gamma_{S_1L}\chi_{D_3} + \gamma_{S_2L}\chi_{D_4}$ and $\gamma_{S_1V}\chi_{D_3} + \gamma_{S_2V}\chi_{D_4}$. As the volume of the drop increases quasi-statically, we use the Modified Algorithem I to calculate the equilibrium state for each fixed volume.

We take $\theta_{\mathcal{A}} = \frac{\pi}{5}$, $\theta_{\mathcal{B}} = \frac{7\pi}{10}$. For the advancing drop, we plot the contact angle

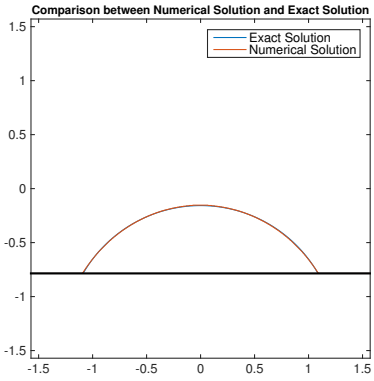


Figure 2.10: Comparison of the numerical solution at equilibrium to the exact solution. Red line represents the exact solution while the blue line represents the numerical solution (computed with 1024×1024 grid points).

and position of contact point as functions of increasing volume in Figure 2.12 for $k = 2$ and in Figure 2.13 for $k = 4$. The contact point goes through the stick-slip motion, and the contact angle oscillates near the advancing angle $\theta_{\mathcal{B}}$ for larger k.

For the receding drop, we plot the contact angle and location of the contact point as functions of increasing volume in Figure 2.14 for $k = 2$ and in Figure 2.15 for $k = 4$. Again, the contact point goes through the stick-slip motion, and the contact angle oscillates near the receding angle $\theta_{\mathcal{A}}$ for larger k.

In Figure 2.16, we show two quasi-static drops. One is in the process of increasing in volume (advancing) and the other is in the process of decreasing in volume (receding). We see that the two states have very different contact angles although the volume is the same. This clearly shows that the contact angle hysteresis as the shape of a drop on a chemically patterned surface depends on its history.

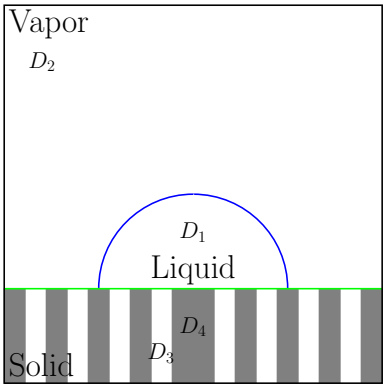


Figure 2.11: A sketch of a drop spreading on a chemically patterned solid surface. Here D_3 (white region) and D_4 (shaded region) represent materials $\mathcal{A}$ and $\mathcal{B}$ respectively.

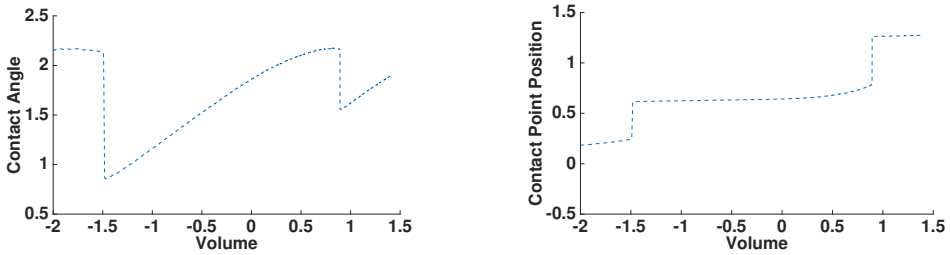


Figure 2.12: The stick-slip motion of a drop with k=2 when the volume is increasing. $\theta_{\mathcal{A}} = \frac{\pi}{5}$, $\theta_{\mathcal{B}} = \frac{7\pi}{10}$.

2.5 A drop spreading on a rough solid surface

In this section, we will simulate the contact angle hysteresis on a geometrically rough surface. In our experiments, the computational domain is $[-\frac{\pi}{2}, \frac{\pi}{2}]\times[-\frac{\pi}{2}, \frac{\pi}{2}]$, and we take the solid surface of shape given by a sawtooth function

$$y = -\frac{\pi}{4} + \tan(\alpha)\frac{\pi}{4k+2}|s((2k+1)x - \pi)|$$

where $s(x)$ is a sawtooth periodic function with period 2π defined as

$$s(x) = \begin{cases} \frac{2}{\pi}(x+\pi) - 1 & -\pi \le x \le 0; \\ -\frac{2}{\pi}x + 1 & 0 \le x \le \pi. \end{cases}$$

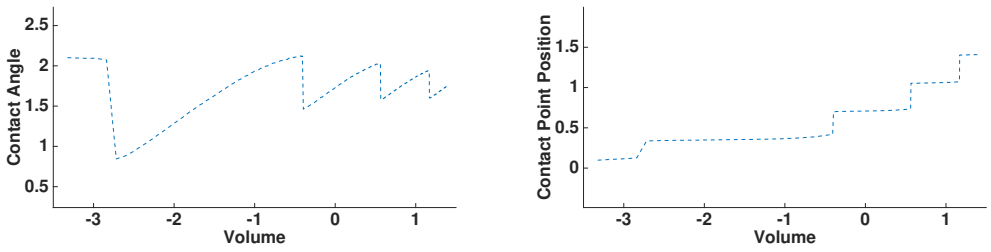


Figure 2.13: The stick-slip motion of a drop with k=4 when the volume is increasing. $\theta_{\mathcal{A}} = \frac{\pi}{5}$, $\theta_{\mathcal{B}} = \frac{7\pi}{10}$.

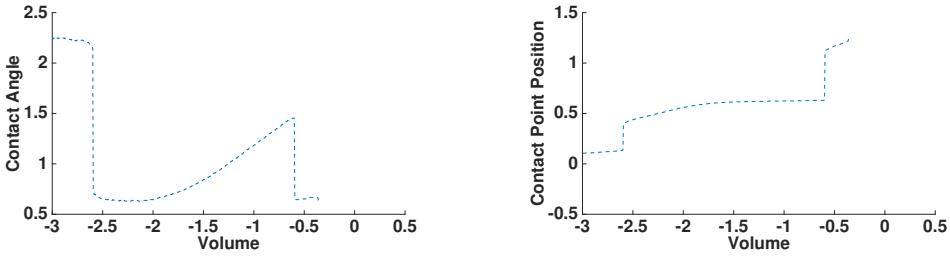


Figure 2.14: The stick-slip motion of a drop with k=2 when the volume is decreasing. $\theta_{\mathcal{A}} = \frac{\pi}{5}$, $\theta_{\mathcal{B}} = \frac{7\pi}{10}$.

For a rough surface, it is more meaningful to see how the effective contact angle behaves when the volume of the drop is increased or decreased [21]. The effective contact angle is defined as the angle between the contact line and the horizontal surface (See Figure 2.17). Figure 2.18 and Figure 2.19 show the bahavior of the contact angle and the x-coordinate of the contact point for the case when $k = 4, \alpha = \frac{\pi}{6}$. Young's angle of the solid surface is $\theta_Y = \frac{\pi}{2}$. We can see obvious stick-slip motion when we increase or decrease the volume of the drop. Furthermore, the advancing contact angle is almost $\frac{2\pi}{3}$ and the receding contact angle is approximately $\frac{\pi}{3}$.

In Figure 2.20, again, we show two quasi-static drops. One is in the process of expanding in volume (advancing) and the other is in the process of reducing in volume (receding). Similar to the chemically patterned surface case, the two

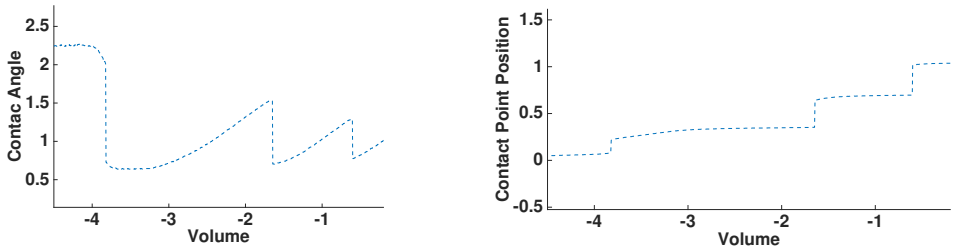


Figure 2.15: The stick-slip motion of a drop with k=4 when the volume is decreasing. $\theta_{\mathcal{A}} = \frac{\pi}{5}$, $\theta_{\mathcal{B}} = \frac{7\pi}{10}$.

states have very different apparent contact angles corresponding to the contact angle hysteresis on rough surfaces.

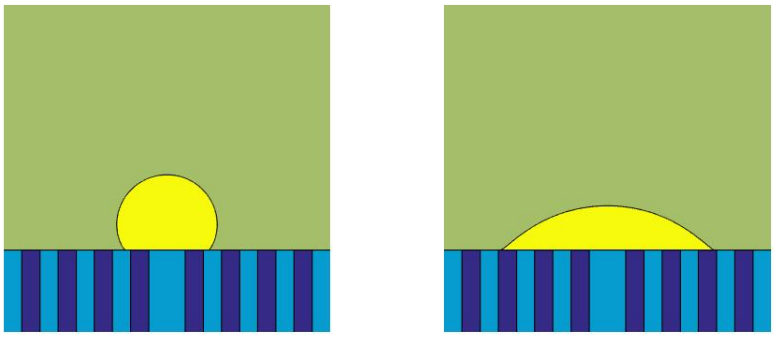


Figure 2.16: Left: A quasi-static drop in the process of growing in volume on a chemically patterned surface when the initial volume is 0.5883. Right: A quasi-static drop in the process of shrinking in volume on a chemically patterned surface when the initial volume is 0.5883. Young's angle in the light blue part is $\mathcal{B}$ while that in the dark blue part is $\mathcal{A}$. Here k=4.

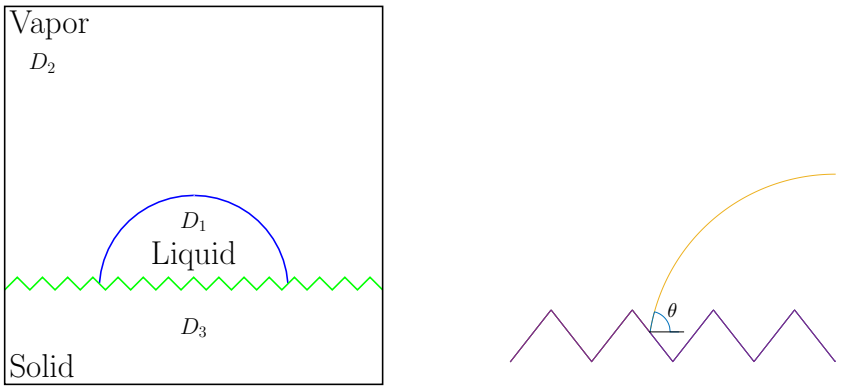


Figure 2.17: Left: A sketch of a drop spreading on a rough solid surface. The solid surface is given by a sawtooth profile. Right: Diagrammatic sketch of effective contact angle θ on the rough solid surface.

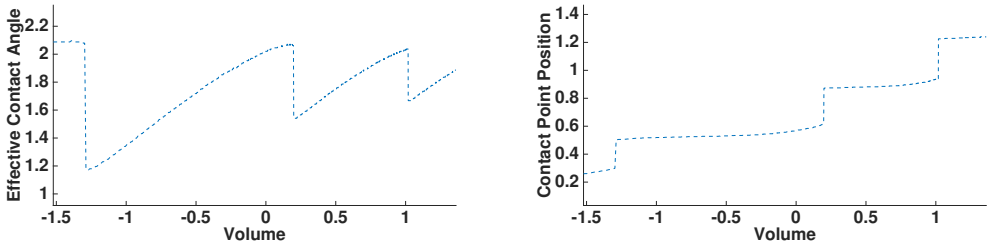


Figure 2.18: The stick-slip motion of a drop on a rough surface when the volume is increasing. $\theta = \frac{\pi}{2}, k = 4, \alpha = \frac{\pi}{6}$.

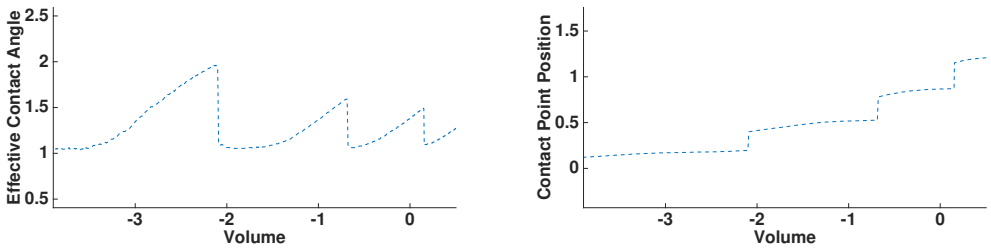


Figure 2.19: The stick-slip motion of a drop on a rough surface when the volume is decreasing. $\theta = \frac{\pi}{2}, k = 4, \alpha = \frac{\pi}{6}$.

Figure 2.20: Left: A quasi-static drop in the process of expanding in volume on a sawtooth rough surface when the volume is 1.178. Right: A quasi-static drop in the process of reducing in volume on a sawtooth rough surface when the volume is 1.178. $\theta = \frac{\pi}{2}, k = 4, \alpha = \frac{\pi}{6}$.

Chapter 3

Convergence proof of the threshold dynamics method for solid wetting problems

In this chapter, we establish the convergence proof of a threshold dynamics method for solid wetting problems (i.e. Algorithm I in Chapter 2). The analysis is motivated by the work in [33, 56, 55]. However, the novelty of our proof is that we consider a case of volume preserving mean-curvature motion described by characteristic function of a domain where part of the boundary is always fixed. Comparing to second section in [55] which establish the convergence proof of a mean-curvature flow with external force, our external force (i.e. the contribution term from D_3) depends on h.

This chapter is organized as the follows: In Section 3.1, we shortly review the algorithm for modelling the solid wetting process proposed in Chapter 2, notations we will use throughout the note and cite some results from [33]. In Section 3.2, we find the corresponding energy and prove that the argument of the minimum(i.e. solution of the corresponding Euler-Lagrange equation) is equivalent to the ap-

proximate solutions found in Algorithm I which will be introduce in Section 3.1. In Section 3.3, we introduce the main result of this note. To put it simply, when $h \searrow 0$, the approximate solutions (χ_1^h, χ_2^h) converge to (χ_1, χ_2) which satisfy the weak definition (3.20) of volume preserving mean-curvature with the angle condition (2.1). In Section 3.4, we give an estimate for λ in Algorithm I. In Section 3.5, we prove the compactness of the approximate solutions. That can indicate the existence of the limit of approximate solutions (χ_1^h, χ_2^h) which will be defined in Section 3.1. In the last section, we prove the convergence of each terms in the Euler-Lagrange equation.

3.1 Algorithm and notations

In this section, we shortly review the algorithm of threshold dynamics method for solid wetting problems stated in Chapter 2 and state some notations, identities we will use throughout this chapter.

When considering a wetting problem in a domain $\Omega \in \mathbb{R}^d$, $d = 2, 3$ (see Figure 2.1). The solid surface Γ (smooth *a.e.*) is part of the domain boundary $\partial\Omega$. Denote the liquid domain by $D_1 \subset \Omega$. For simplicity, we assume that $\partial D_1 \cap \partial\Omega \subset \Gamma$. The volume of the liquid drop is fixed such that $|D_1| = 1$. We denote $\Sigma_{LV} = \partial D_1 \cap \Omega$, $\Sigma_{SL} = \partial D_1 \cap \Gamma$ and $\Sigma_{SV} = \Gamma \setminus \partial D_1$ as the liquid-vapor, solid-liquid and solid-vapor interfaces respectively. We also denote $\gamma_{LV} > 0, \gamma_{SL} > 0, \gamma_{SV} > 0$ as the surface tension on the liquid-vapor, solid-liquid and solid-vapor interfaces respectively. Here, $\gamma_{LV}, \gamma_{SL}, \gamma_{SV}$ satisfy Young's equation (2.1) which can be derived by minimizing the total energy in the solid-liquid-vapor system. If we ignore gravity, the total energy in the system can be written as 2.2.

The main idea in Chapter 2 is to extend the liquid-vapor domain to a larger

domain $\tilde{\Omega}$ (without loss of generality, we assume here $\tilde{\Omega} = [0, \Lambda/2]^d$ with $\Lambda < \infty$) containing the solid phase denoted by D_3 (see Figure 2.2). Analogous to [33], the energy (2.2) can be approximated by

$$\mathcal{E}^h(\chi_{D_1}, \chi_{D_2}) = \\ \frac{\gamma_{LV}}{\sqrt{h}} \int_{\tilde{\Omega}} \chi_{D_1} G_h * \chi_{D_2} \mathrm{d}\mathbf{x} + \frac{\gamma_{SL}}{\sqrt{h}} \int_{\tilde{\Omega}} \chi_{D_1} G_h * \chi_{D_3} \mathrm{d}\mathbf{x} + \frac{\gamma_{SV}}{\sqrt{h}} \int_{\tilde{\Omega}} \chi_{D_2} G_h * \chi_{D_3} \mathrm{d}\mathbf{x} \quad (3.1)$$

where

$$G_h(\mathbf{x}) := \frac{1}{(4\pi h)^{n/2}} exp\left(-\frac{|\mathbf{x}|^2}{4h}\right)$$

is the heat kernel at time h.

Based on the approximate energy, we rewrite Algorithm I which we proposed in Chapter 2 here:

Algorithm I:

Given phase $D_1, D_2 \subset \Omega$ with $|D_1| = 1$ such that $D_1 \cap D_2 = \emptyset$, $D_1 \cup D_2 = \Omega$ at $t = (n-1)h$, denote χ_{D_i} as the characteristic function of $D_i (i = 1, 2, 3)$. Obtain the new phase $D_1^{'}, D_2^{'}$ at $t = nh$ by :

Step 1. Define two functions

$$\phi_1 = G_h * (\gamma_{LV}\chi_{D_2} + \gamma_{SL}\chi_{D_3}), \ \phi_2 = G_h * (\gamma_{LV}\chi_{D_1} + \gamma_{SV}\chi_{D_3}). \quad (3.2)$$

Step 2. Find a λ such that the set

$$\tilde{D}_1^h = \{x \in \Omega | \phi_1 < \phi_2 + \lambda.\} \quad (3.3)$$

satisfies $|\tilde{D}_1^h| = 1$.

Step 3. Denote $D_1^{'} = \tilde{D}_1^h$ and $D_2^{'} = \Omega \setminus D_1^{'}$.

Remark 3.1.1. *In the following, we assume the domain $\tilde{\Omega}$ is extended evenly to $\widehat{\Omega} = [0, \Lambda)^d$ by reflection and Ω is also correspondingly extended which we will still denote as Ω. Then, the domain we considered is periodic.*

Now, we introduce some notations which we will use throughout the note. Denote D_1^n, D_2^n as the domain of liquid phase and solid phase respectively at the n^{th} time step. Then, we denote the characteristic function of D_1^n, D_2^n by χ_1^n, χ_2^n respectively and interpolate these functions piecewise constantly in time, i.e.

$$\chi_1^h(t,x) := \chi_1^n \quad \text{for} \quad t \in [nh, (n+1)h),$$
$$\chi_2^h(t,x) := \chi_2^n \quad \text{for} \quad t \in [nh, (n+1)h).$$

Define the admissible set as

$$\begin{aligned}\mathcal{B} = \{(\chi_1, \chi_2) \in BV(\widehat{\Omega}) \mid \chi_i(x) = 0, 1,\ \chi_1(x) + \chi_2(x) = 1,\ a.e.\ x \in \Omega \\ \text{and}\, \chi_1(x) + \chi_2(x) = 0,\ a.e.\ x \in \widehat{\Omega}\backslash\Omega, \\ \int \chi_1 \mathrm{d}\mathbf{x} = 1\}. \end{aligned} \tag{3.4}$$

Here and in the sequel, $\int \mathrm{d}\mathbf{x}$ represents $\int_{\widehat{\Omega}} \mathrm{d}\mathbf{x}$.

For the solid wetting problem, assume $\operatorname{supp} \chi_1 \subset\subset \widehat{\Omega}$, we define the following approximate energy for liquid-vapor interface

$$\mathcal{E}_h(\chi_1, \chi_2) := \frac{\gamma_{LV}}{\sqrt{h}} \int \chi_1 G_h * \chi_2 \mathrm{d}\mathbf{x} \tag{3.5}$$

for $(\chi_1, \chi_2) \in \mathcal{B}$ and the approximate dissipation functional as

$$\mathcal{D}_h(\omega_1, \omega_2) := \frac{\gamma_{LV}}{\sqrt{h}} \int \omega_1 G_h * \omega_2 \mathrm{d}\mathbf{x} \tag{3.6}$$

for $\omega_1, \omega_2 : \widehat{\Omega} \to \{-1, 0, 1\}$ *a.e.* with $\omega_1 + \omega_2 = 0$ and the approximate energy for solid-liquid and solid-vapor interface as

$$\mathcal{S}_h(\chi_1, \chi_2) := \frac{\gamma_{SL}}{\sqrt{h}} \int \chi_1 G_h * \chi_3 \mathrm{d}\mathbf{x} + \frac{\gamma_{SV}}{\sqrt{h}} \int \chi_2 G_h * \chi_3 \mathrm{d}\mathbf{x} \tag{3.7}$$

for any $(\chi_1, \chi_2) \in \mathcal{B}$ and the fixed function $\chi_3 = \chi_{D_3}$.

We also define a notation $C_s^\infty((0,T)\times\widehat{\Omega},\mathbb{R}^d)$ for a special function space we will use throughout this note by

$$C_s^\infty((0,T)\times\widehat{\Omega},\mathbb{R}^d) := \{f | f \in C_0^\infty((0,T)\times\widehat{\Omega},\mathbb{R}^d) \text{ and } f\cdot\nu = 0 \ a.e. \ in \ \widehat{\Omega}\} \tag{3.8}$$

where ν represents the unit normal vector of Γ and $C_s^\infty(\widehat{\Omega},\mathbb{R}^d)$ is also similarly defined.

Remark 3.1.2. *The definition of function space C_s^∞ is a constraint that enforce the contact points to move only along the solid surface (i.e. the interface between D_3 and $D_1\cup D_2$).*

Now, we introduce an identity for mean curvature of interface through the following lemma.

Lemma 3.1.1. *For any interface Σ and all $\xi\in C_0^\infty(\widehat{\Omega},\mathbb{R}^d)$, we have*

$$\int_\Sigma(\nabla\cdot\xi-\nu\cdot\nabla\xi\nu)d\Sigma = \int_\gamma b\cdot\xi ds + \int_\Sigma \mathcal{H}\nu\cdot\xi d\Sigma \tag{3.9}$$

where $\gamma=\partial\Sigma$, b denotes the conormal vector, $\mathcal{H}=\nabla\cdot\nu$ denotes the mean curvature of Σ, and ν denotes the unit normal vector of interface Σ.

Proof. When $d=3$, note that the conormal vector at a point on γ is a vector parallel to the tangential plane of Σ but orthogonal to γ. That is, denote $\vec{t}$ as the tangential vector of γ in the direction of counterclockwise, we have $b=\vec{t}\times\nu$. Then from basic vector identities and Stoke's theorem, we have

$$\begin{aligned}
\int_\gamma b\cdot\xi ds &= -\int_\gamma \xi\cdot(\nu\times d\vec{s}) \\
&= -\int_\gamma(\xi\times\nu)\cdot d\vec{s} \\
&= -\int_\Sigma \nabla\times(\xi\times\nu)\cdot\nu d\Sigma \\
&= -\int_\Sigma(\xi(\nabla\cdot\nu)-(\nabla\cdot\xi)\nu+(\nu\cdot\nabla)\xi-(\xi\cdot\nabla)\nu)\cdot\nu d\Sigma.
\end{aligned}$$

Since ν is a unit normal vector, we have $(\xi \cdot \nabla)\nu \cdot \nu = 0$ and $(\nabla \cdot \xi)\nu \cdot \nu = (\nabla \cdot \xi)$. Hence, we have

$$\int_\gamma b \cdot \xi ds + \int_\Sigma (\nabla \cdot \nu)\xi \cdot \nu d\Sigma = \int_\Sigma \nabla \cdot \xi - \nu \cdot \nabla \xi \nu d\Sigma.$$

The case when $d = 2$ can be seen as the projection of a special case when $d = 3$ where the domain is represented by characteristic function χ such that $\chi(x_1, x_2, x_3) = \chi(x_1, x_2)$. □

Remark 3.1.3. *From above identity, we can recover the curvature from* $\int_\Sigma (\nabla \cdot \xi - \nu \cdot \nabla \xi \nu) d\Sigma$. *What's more, we can also recover the conormal vector of the interface which is crucial for deriving the Young's equation at the contact point.*

We cite two results in [33] here which will be used later. The first is the following monotonicity statement about $\mathcal{E}_h$ and $\mathcal{S}_h$: For any $0 < h \leq h_0$ and any $(\chi_1, \chi_2) \in \mathcal{B}$, we have

$$\mathcal{E}_h(\chi_1, \chi_2) \geq \left(\frac{\sqrt{h_0}}{\sqrt{h} + \sqrt{h_0}}\right)^{d+1} \mathcal{E}_{h0}(\chi_1, \chi_2), \tag{3.10}$$

$$\mathcal{S}_h(\chi_1, \chi_2) \geq \left(\frac{\sqrt{h_0}}{\sqrt{h} + \sqrt{h_0}}\right)^{d+1} \mathcal{S}_{h0}(\chi_1, \chi_2). \tag{3.11}$$

The second is the consistency result about the convergence of $\mathcal{E}_h$ and $\mathcal{S}_h$: For any $(\chi_1, \chi_2) \in \mathcal{B}$, we have

$$\lim_{h \to 0} \mathcal{E}_h(\chi_1, \chi_2) = \mathcal{E}(\chi_1, \chi_2), \tag{3.12}$$

$$\lim_{h \to 0} \mathcal{S}_h(\chi_1, \chi_2) = \mathcal{S}(\chi_1, \chi_2) \tag{3.13}$$

and

$$\mathcal{E}_0 := \mathcal{E}(\chi_1^0, \chi_2^0) \geq \mathcal{E}_h(\chi_1^0, \chi_2^0), \tag{3.14}$$

$$\mathcal{S}_0 := \mathcal{S}(\chi_1^0, \chi_2^0) \geq \mathcal{S}_h(\chi_1^0, \chi_2^0) \tag{3.15}$$

where $\mathcal{E}(\chi_1, \chi_2) := \frac{\gamma_{LV}}{\sqrt{\pi}} \int |\nabla \chi_1| + |\nabla \chi_2| - |\nabla(\chi_1 + \chi_2)|$ and $\mathcal{S}(\chi_1, \chi_2) := \frac{\gamma_{SL}}{\sqrt{\pi}} \int (|\nabla \chi_1| + |\nabla \chi_3| - |\nabla(\chi_1 + \chi_3)|) + \frac{\gamma_{SV}}{\sqrt{\pi}} \int (|\nabla \chi_2| + |\nabla \chi_3| - |\nabla(\chi_2 + \chi_3)|)$. Here and in the

sequel, $\int |\nabla\chi|$ is defined by

$$\int |\nabla\chi| := \sup\left\{\int \chi\nabla\cdot f\mathrm{d}\mathbf{x}\,|\, f\in C^1(\widehat{\Omega},\mathbb{R}^d), |g|\leq 1\right\}. \tag{3.16}$$

The following definition of our weak notion of volume preserving mean-curvature flow with the angle condition (2.1) on solid surface [19, 89] is a distributional formulation which is suited to the frame work of functions of bounded variation. We refer readers to [5, 6, 61, 55] for more details about this type of distributional formulation for mean curvature flow represented by the variation of characteristic functions.

Definition 3.1.1 (Dynamic of solid wetting process)**.** *Fix some finite time horizon $T<\infty$, three surface tensions $\gamma_{LV},\gamma_{SL},\gamma_{SV}$ and initial data $(\chi_1,\chi_2)\in\mathcal{B}$ with $\mathcal{E}_0<\infty,\mathcal{S}_0<\infty$. We say that*

$$(\chi_1,\chi_2):(0,T)\times\widehat{\Omega}\to\{0,1\}^2 \tag{3.17}$$

with $(\chi_1(t),\chi_2(t))\in\mathcal{B}$ and

$$\sup_t \mathcal{E}(\chi_1(t),\chi_2(t))<\infty,\qquad \sup_t \mathcal{S}(\chi_1(t),\chi_2(t))<\infty \tag{3.18}$$

moves by volume preserving mean-curvature flow with a relaxation boundary condition on the solid surface if there exists a function $V:(0,T)\times\widehat{\Omega}\to R$ with

$$\int_0^T\int V^2|\nabla\chi_1|dt<\infty \tag{3.19}$$

which satisfies

$$\begin{aligned}
&\gamma_{LV}\int_0^T\int(\nabla\cdot\xi-\nu_1\cdot\nabla\xi\nu_1-\xi\cdot\nu_1(V+\widehat{H}))(|\nabla\chi_1|+|\nabla\chi_2|-|\nabla(\chi_1+\chi_2)|)dt\\
+&\gamma_{SV}\int_0^T\int(\nabla\cdot\xi-\nu_3\cdot\nabla\xi\nu_3)(|\nabla\chi_2|+|\nabla\chi_3|-|\nabla(\chi_2+\chi_3)|)dt\\
+&\gamma_{SL}\int_0^T\int(\nabla\cdot\xi-\nu_3\cdot\nabla\xi\nu_3)(|\nabla\chi_1|+|\nabla\chi_3|-|\nabla(\chi_1+\chi_3)|)dt=0
\end{aligned} \tag{3.20}$$

for all $\xi\in C_s^\infty((0,T)\times\widehat{\Omega},\mathbb{R}^d)$ and

$$\int_0^T\int\partial_t\zeta\chi_1dxdt+\int\zeta(0)\chi_i^0dx=-\int_0^T\int\zeta V|\nabla\chi_1|dt \tag{3.21}$$

for all $\zeta \in C^\infty((0,T)\times\widehat{\Omega},\mathbb{R})$ *with* $\zeta(T)=0$ *where* $\widehat{H}\in L^2(0,T)$ *is the average of the generalized mean curvature* $H\in L^2((|\nabla\chi_1|+|\nabla\chi_2|-|\nabla(\chi_1+\chi_2)|)dt)$ *of the liquid-vapor interface:*

$$\widehat{H} := \frac{\int H(|\nabla\chi_1|+|\nabla\chi_2|-|\nabla(\chi_1+\chi_2)|)}{\int(|\nabla\chi_1|+|\nabla\chi_2|-|\nabla(\chi_1+\chi_2)|)}. \tag{3.22}$$

Note that from the definition (3.20) and identity (3.9), we can not only derive that the normal velocity for the liquid-vapor interface is the mean curvature minus the average of mean curvature, but also have the Young's equation ((2.1)) at the contact point since the last two term only recover the conormal vector from the definition of $C_s^\infty((0,T)\times\widehat{\Omega},\mathbb{R}^d)$.

In the following, we write $A\lesssim B$ to represent that $A\le CB$ for a generic constant $C<\infty$ that only depends on the dimension d.

3.2 Minimizing movement interpretation

In this section, we first introduce an equivalent minimizing movement scheme to Algorithm I. That is, when given χ_1^{n-1} and χ_2^{n-1}, we find a total energy such that (χ_1^n,χ_2^n) found in Algorithm I is corresponding minimizer. Then, we introduce the first variation of bounded variation functions and find the Euler-Lagrange equation of the energy.

Lemma 3.2.1. *Given* $(\chi_1^0,\chi_2^0)\in\mathcal{B}$*, let* $\phi_1,\phi_2,\lambda,\chi_1^1$ *and* χ_2^1 *be obtained by Algorithm I. Then* (χ_1^1,χ_2^1) *solves*

$$\min\quad \mathcal{E}_h(\chi_1,\chi_2)-\mathcal{D}_h(\chi_1-\chi_1^0,\chi_2-\chi_2^0)+\mathcal{S}_h(\chi_1,\chi_2)-\frac{\lambda}{\sqrt{h}}\int\chi_1 d\mathbf{x}, \tag{3.23}$$

where the minimum runs over the admissible set $\mathcal{B}$*.*

Proof. From the definition of $\mathcal{E}_h$, $\mathcal{D}_h$ and $\mathcal{S}_h$, we can see

$$\begin{aligned}&\mathcal{E}_h(\chi_1,\chi_2)-\mathcal{D}_h(\chi_1-\chi_1^0,\chi_2-\chi_2^0)+\mathcal{S}_h(\chi_1,\chi_2)-\frac{\lambda}{\sqrt{h}}\int\chi_1\mathrm{d}\mathbf{x}\\=&\mathcal{L}_h(\phi_1,\phi_2,\chi_1,\chi_2)-\int\chi_1^0G_h*\chi_2^0\mathrm{d}\mathbf{x},\end{aligned}$$

where

$$\mathcal{L}_h(\phi_1,\phi_2,\chi_1,\chi_2)=\frac{1}{\sqrt{h}}\int\left\{\chi_1(\phi_1-\lambda)+\chi_2\phi_2\right\}\mathrm{d}\mathbf{x}. \tag{3.24}$$

Hence, (3.23) is equivalent to minimizing $\mathcal{L}_h(\phi_1,\phi_2,\chi_1,\chi_2)$ on $\mathcal{B}$. Now, we show that (χ_1^1,χ_2^1) achieved in Algorithm I is a minimizer of $\mathcal{L}_h(\phi_1,\phi_2,\chi_1,\chi_2)$ on $\mathcal{B}$. The integrand in (3.24) is clearly bounded below by $\min(\phi_1-\lambda,\phi_2)$ at each $\mathbf{x}$ where $(\chi_1,\chi_2)\in\mathcal{B}$ and $\chi_1(\mathbf{x})+\chi_2(\mathbf{x})=1$. Obviously, by definition of Algorithm I, (χ_1^1,χ_2^1) admits this minimum pointwise. Thus (χ_1^1,χ_2^1) minimizes $\mathcal{L}_h(\phi_1,\phi_2,\chi_1,\chi_2)$ for all $(\chi_1,\chi_2)\in\mathcal{B}$. □

Below is the estimate for the bound of the approximate energy $\mathcal{E}(\chi_1^h,\chi_2^h)$.

Lemma 3.2.2 (Energy-dissipation estimate)**.** *The approximate solutions* (χ_1^n,χ_2^n) *satisfy the following energy-dissipation estimate*

$$\mathcal{E}_h(\chi_1^N,\chi_2^N)+\mathcal{S}_h(\chi_1^N,\chi_2^N)-\sum_{n=1}^{N}D_h(\chi_1^n-\chi_1^{n-1},\chi_2^n-\chi_2^{n-1})\le\mathcal{E}_0+\mathcal{S}_0. \tag{3.25}$$

Proof. From the minimization procedure (3.23), we can have

$$\begin{aligned}&\mathcal{E}_h(\chi_1^n,\chi_2^n)+\mathcal{S}_h(\chi_1^n,\chi_2^n)-D_h(\chi_1^n-\chi_1^{n-1},\chi_2^n-\chi_2^{n-1})\\\le&\mathcal{E}_h(\chi_1^{n-1},\chi_2^{n-1})+\mathcal{S}_h(\chi_1^{n-1},\chi_2^{n-1})-D_h(\chi_1^{n-1}-\chi_1^{n-1},\chi_2^{n-1}-\chi_2^{n-1})\\=&\mathcal{E}_h(\chi_1^{n-1},\chi_2^{n-1})+\mathcal{S}_h(\chi_1^{n-1},\chi_2^{n-1}).\end{aligned}$$

After iterating this estimate from $n=1$ to N together with (3.14) and (3.15), we can get the result. □

Definition 3.2.1 (First variation)**.** *For any* $(\chi_1, \chi_2) \in \mathcal{B}$ *and* $\xi \in C_s^\infty(\widehat{\Omega}, \mathbb{R}^d)$*, let* $\chi_{i,s}$ *be generated by the flow of* ξ*, i.e.* $\chi_{i,s}$ *solves the following distributional equation:*

$$\frac{\partial \chi_{i,s}}{\partial s} + \xi \cdot \nabla \chi_{i,s} = 0.$$

We denote the first variation along this flow by

$$\begin{aligned}
\delta\mathcal{E}_h(\chi_1, \chi_2, \xi) :&= \frac{d}{ds}\mathcal{E}_h(\chi_{1,s}, \chi_{2,s})|_{s=0} \\
&= -\frac{\gamma_{LV}}{\sqrt{h}} \int \xi \cdot \nabla \chi_1 G_h * \chi_2 + \chi_1 G_h * (\xi \cdot \nabla \chi_2) \mathrm{d}\mathbf{x}, \quad (3.26) \\
\delta\mathcal{D}_h(\chi_1, \chi_2, \widetilde{\chi_1}, \widetilde{\chi_2}, \xi) :&= \frac{d}{ds}\mathcal{D}_h(\chi_{1,s} - \widetilde{\chi_1}, \chi_{2,s} - \widetilde{\chi_2})|_{s=0} \\
&= -\frac{\gamma_{LV}}{\sqrt{h}} \int \xi \cdot \nabla \chi_1 G_h * (\chi_2 - \widetilde{\chi_2}) + (\chi_1 - \widetilde{\chi_1}) G_h * (\xi \cdot \nabla \chi_2), \\
& \qquad (3.27) \\
\delta\mathcal{S}_h(\chi_1, \chi_2, \xi) :&= \frac{d}{ds}\mathcal{S}_h(\chi_{1,s}, \chi_{2,s})|_{s=0} \\
&= -\frac{\gamma_{SL}}{\sqrt{h}} \int \xi \cdot \nabla \chi_1 G_h * \chi_3 \mathrm{d}\mathbf{x} - \frac{\gamma_{SV}}{\sqrt{h}} \int \xi \cdot \nabla \chi_2 G_h * \chi_3 \mathrm{d}\mathbf{x}, \\
& \qquad (3.28)
\end{aligned}$$

where $(\widetilde{\chi_1}, \widetilde{\chi_2}) \in \mathcal{B}$ *is fixed.*

Lemma 3.2.3 (Euler-Lagrange equation)**.** *Given* $(\chi_1^0, \chi_2^0) \in \mathcal{B}$*, let* (χ_1^1, χ_2^1) *be obtained by Algorithm I with threshold value* λ*. Then* (χ_1^1, χ_2^1) *solves the Euler-Lagrange equation*

$$\delta\mathcal{E}_h(\chi_1^1, \chi_2^1, \xi) - \delta\mathcal{D}_h(\chi_1^1, \chi_2^1, \chi_1^0, \chi_2^0, \xi) + \delta\mathcal{S}_h(\chi_1, \chi_2, \xi) - \frac{\lambda}{\sqrt{h}} \int (\nabla \cdot \xi) \chi_1^1 \mathrm{d}\mathbf{x} = 0. \tag{3.29}$$

Proof. The last term in (3.29) is the first variation of $\frac{\lambda}{\sqrt{h}} \int \chi_1 \mathrm{d}\mathbf{x}$. It can be quickly calculated as:

$$\delta\{\frac{\lambda}{\sqrt{h}} \int \chi_1 \mathrm{d}\mathbf{x}\} = \frac{\lambda}{\sqrt{h}} \int \frac{d\chi_{1,s}}{ds} \mathrm{d}\mathbf{x}|_{s=0} = \frac{\lambda}{\sqrt{h}} \int -\xi \cdot \nabla \chi_1 \mathrm{d}\mathbf{x} = \frac{\lambda}{\sqrt{h}} \int (\nabla \cdot \xi) \chi_1 \mathrm{d}\mathbf{x}.$$

□

3.3 Main results

In this section, we introduce the main result and give a brief summary of the main idea of the whole note. We first establish the equivalence between the solution of (3.29) and Algorithm I which was done in Lemma 3.2.3. Then, we give an estimate of λ in (3.29) to show that λ is bounded and goes to 0 as $h \searrow 0$ which will be done in Section 3.4. In Section 3.5, we prove the existence of the limit of (χ_1^h, χ_1^h) which is denoted as (χ_1, χ_2) such that $(\chi_1, \chi_2) \in \mathcal{B}$. Then, we analysis the convergence of $\delta\mathcal{E}_h, \delta\mathcal{D}_h$ and $\delta\mathcal{S}_h$ in Proposition 3.6.1, 3.6.2 and 3.6.3 respectively.

Now, we give the main result of this note in the following theorem. To put it simply, that is, under some assumption of the convergence, we can have that the approximate solutions (χ_1^h, χ_2^h) converge to (χ_1, χ_2) which satisfy Definition 3.20 as $h \searrow 0$.

Theorem 3.3.1. *Given $\gamma_{LV}, \gamma_{SV}, \gamma_{SL}$, a finite time horizon $T < \infty$ and $(\chi_1^0, \chi_2^0) \in \mathcal{B}$ satisfied with $\mathcal{E}(\chi_1^0, \chi_2^0) + \mathcal{S}(\chi_1^0, \chi_2^0) < \infty$. Then, for any sequence there exists a subsequence $h \searrow 0$ and a $(\chi_1, \chi_2) : (0, T) \times \widehat{\Omega} \to \{0, 1\}^2$ with $\mathcal{E}(\chi_1(t), \chi_2(t)) + \mathcal{S}(\chi_1(t), \chi_2(t)) \leq \mathcal{E}_0 + \mathcal{S}_0$ such that the approximate solutions (χ_1^h, χ_2^h) obtained in Algorithm I converge to (χ_1, χ_2). Furthermore, if we assume*

$$\int_0^T \mathcal{E}_h(\chi_1^h, \chi_2^h)dt \to \int_0^T \mathcal{E}(\chi_1, \chi_2)dt,$$
$$\int_0^T \mathcal{S}_h(\chi_1^h, \chi_2^h)dt \to \int_0^T \mathcal{S}(\chi_1, \chi_2)dt,$$

then (χ_1, χ_2) moves by the volume preserving mean-curvature with the angle condition (2.1) on the solid surface in the sense of Definition 3.20 with initial data (χ_1^0, χ_2^0).

Proof. From Proposition 3.5.1, we can have (χ_1^h, χ_2^h) converge to some limit (χ_1, χ_2). In Proposition 3.6.1, we recover the mean curvature term from $\delta\mathcal{E}_h$.

In Proposition 3.6.2, we recover the velocity term from $\delta\mathcal{D}_h$. In Proposition 3.6.3, we recover the conormal vector of solid-liquid interface and solid-vapor interface from $\delta\mathcal{S}_h$.

Now, we consider the last term in (3.29). By Proposition 3.4.1, we can find a function $\widehat{H} \in L^2(0,T)$ such that

$$\frac{\lambda_h}{\sqrt{h}} \to \frac{\widehat{H}}{\sqrt{\pi}} \text{ in } L^2(0,T). \tag{3.30}$$

Since

$$\int (\nabla \cdot \xi)\chi_1^h \mathrm{d}\mathbf{x} \to \int (\nabla \cdot \xi)\chi_1 \mathrm{d}\mathbf{x} \text{ in } L^2(0,T), \tag{3.31}$$

we can have

$$\frac{\lambda_h}{\sqrt{h}} \int (\nabla \cdot \xi)\chi_1^h \mathrm{d}\mathbf{x} \to \frac{\widehat{H}}{\sqrt{\pi}} \int (\nabla \cdot \xi)\chi_1 \mathrm{d}\mathbf{x} \text{ in } L^2(0,T). \tag{3.32}$$

Now, we only need to prove (3.22). Indeed, by the volume preserving of χ_1, we have

$$\begin{aligned}
0 =& \frac{d}{dt}\int \chi_1 \mathrm{d}\mathbf{x} \\
=& \int V|\nabla\chi_1| \\
=& \int V_1(|\nabla\chi_1| + |\nabla\chi_2| - |\nabla(\chi_1+\chi_2)|) \\
=& \int (H - \widehat{H})(|\nabla\chi_1| + |\nabla\chi_2| - |\nabla(\chi_1+\chi_2)|) \\
=& \int H(|\nabla\chi_1| + |\nabla\chi_2| - |\nabla(\chi_1+\chi_2)|) - \widehat{H}\int (|\nabla\chi_1| + |\nabla\chi_2| - |\nabla(\chi_1+\chi_2)|)
\end{aligned}$$

a.e. in $(0,T)$. Direct computation yields (3.22). □

3.4 Estimate for λ

In this section, we give a quantitative estimate of the threshold values λ_n. We denote λ_h as the piecewise constant interpolation of λ_n in time:

$$\lambda_h(t) := \lambda_n \quad \text{for} \quad t \in [nh, (n+1)h).$$

Proposition 3.4.1. *Given the approximate solutions* $(\chi_1^h, \chi_2^h) \in \mathcal{B}$ *obtained by Algorithm I with threshold values* λ_h*, for* $h < \frac{1}{C(\mathcal{E}_0+\mathcal{S}_0)^2}$ *with* $C = C(d) < \infty$*, we have*

$$\int_0^T \lambda_h^2 dt \lesssim (1+T)(1+(\mathcal{E}_0+\mathcal{S}_0)^4)h. \tag{3.33}$$

Proof. Squaring the Euler-Lagrange equation (3.29), we obtain for each $n \leq \frac{T}{h}$

$$\begin{aligned}&\frac{1}{h}\lambda_n^2 \left(\int (\nabla\cdot\xi)\chi_1^n \mathrm{d}\mathbf{x}\right)^2\\ \lesssim&[\delta\mathcal{E}_h(\chi_1^n,\chi_2^n,\xi)]^2 + [\delta\mathcal{D}_h(\chi_1^n,\chi_2^n,\chi_1^{n-1},\chi_2^{n-1},\xi)]^2 + [\delta\mathcal{S}_h(\chi_1^n,\chi_2^n,\xi)]^2\end{aligned} \tag{3.34}$$

for any $\xi \in C_s^\infty(\widehat{\Omega},\mathbb{R}^d)$. We first estimate the right-hand side for an arbitrary test field ξ in the first three steps. Then, we construct a specific vector field as in [56] such that the integral on the left-hand side is bounded from below.

Step 1: Estimates on $\delta\mathcal{E}_h(\chi_1,\chi_2,\xi)$**.** For any $(\chi_1,\chi_2) \in \mathcal{B}$ and any $\xi \in C_s^\infty(\widehat{\Omega},\mathbb{R}^n)$, we have

$$|\delta\mathcal{E}_h(\chi_1,\chi_2,\xi)| \lesssim ||\nabla\xi||_\infty \mathcal{E}_h(\chi_1,\chi_2). \tag{3.35}$$

Argument for (3.35): We use the definition in (3.26) and integration by parts to have

$$\begin{aligned}\frac{1}{\gamma_{LV}}\delta\mathcal{E}_h(\chi_1,\chi_2,\xi) =& -\frac{1}{\sqrt{h}}\int \xi\cdot\nabla\chi_1 G_h * \chi_2 + \chi_1 G_h * (\xi\cdot\nabla\chi_2)\mathrm{d}\mathbf{x},\\ =& I_1 + I_2\end{aligned}$$

where

$$\begin{aligned}I_1 :=& \frac{1}{\sqrt{h}}\int \xi\cdot[\chi_1\nabla G_h * \chi_2] - \chi_1\nabla G_h * (\xi\chi_2)\mathrm{d}\mathbf{x}\\ I_2 :=& \frac{1}{\sqrt{h}}\int (\nabla\cdot\xi)\chi_1 G_h * \chi_2 + \chi_1 G_h * ((\nabla\cdot\xi)\chi_2)\mathrm{d}\mathbf{x}.\end{aligned}$$

I_1 can be estimated via

$$\begin{aligned}
I_1 =&\frac{1}{\sqrt{h}}\int \xi\cdot[\chi_1\nabla G_h * \chi_2] - \chi_1\nabla G_h * (\xi\chi_2)\mathrm{d}\mathbf{x}\\
=&\frac{1}{\sqrt{h}}\int \xi(\mathbf{x})\cdot[\chi_1(\mathbf{x})\int \nabla G_h(\mathbf{z})\chi_2(\mathbf{x}-\mathbf{z})\mathrm{d}\mathbf{z}]\mathrm{d}\mathbf{x}\\
&-\frac{1}{\sqrt{h}}\int \chi_1(\mathbf{x})\int \nabla G_h(\mathbf{z})\xi(\mathbf{x}-\mathbf{z})\chi_2(\mathbf{x}-\mathbf{z})\mathrm{d}\mathbf{z}\mathrm{d}\mathbf{x}\\
=&\frac{1}{\sqrt{h}}\int\int \nabla G_h(\mathbf{z})\cdot(\xi(\mathbf{x})-\xi(\mathbf{x}-\mathbf{z}))\chi_1(\mathbf{x})\chi_2(\mathbf{x}-\mathbf{z})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z}\\
=&\frac{1}{\sqrt{h}}\int\int -G_h(\mathbf{z})\frac{\mathbf{z}}{2h}\cdot(\xi(\mathbf{x})-\xi(\mathbf{x}-\mathbf{z}))\chi_1(\mathbf{x})\chi_2(\mathbf{x}-\mathbf{z})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z}.
\end{aligned}$$

We can have

$$|I_1| \leq \frac{1}{\sqrt{h}}\int\int G_h(\mathbf{z})\frac{|\mathbf{z}|^2}{2h}||\nabla\xi||_\infty\chi_1(\mathbf{x})\chi_2(\mathbf{x}-\mathbf{z})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z}.$$

It is obvious that $\frac{|\mathbf{z}|^2}{h}G_h(\mathbf{z}) \lesssim G_{2h}(\mathbf{z})$. Then we can get

$$|I_1| \lesssim ||\nabla\xi||_\infty\frac{1}{\gamma_{LV}}\mathcal{E}_{2h}(\chi_1,\chi_2).$$

$|I_2|$ is clearly bounded by $\frac{2}{\gamma_{LV}}||\nabla\xi||_\infty\mathcal{E}_h(\chi_1,\chi_2)$. Thus, we have

$$|\delta\mathcal{E}_h(\chi_1,\chi_2,\xi)| \lesssim ||\nabla\xi||_\infty\left(\mathcal{E}_h(\chi_1,\chi_2)+\mathcal{E}_{2h}(\chi_1,\chi_2)\right). \tag{3.36}$$

The approximate monotonicity of the energy (3.10) yields (3.35).

Step 2: Estimates on $\delta\mathcal{D}_h(\chi_1^1,\chi_2^1,\chi_1^0,\chi_2^0,\xi)$. We have

$$h\sum_{n=1}^{N}(\delta\mathcal{D}_h(\chi_1^n,\chi_2^n,\chi_1^{n-1},\chi_2^{n-1},\xi))^2 \lesssim \sup_n||\xi||^2_{W^{1,\infty}}(1+\mathcal{E}_0^2). \tag{3.37}$$

Argument for (3.37): Before we proof (3.37), we first mention that the solution $u(\mathbf{x},t)$ for

$$\begin{aligned}
&u_t = \Delta u,\\
&u(\mathbf{x},0) = f,\ f\in L^p(\mathbb{R}^d),\ 1\leq p<\infty
\end{aligned}$$

is a semigroup on $L^p(\mathbb{R}^d), 1\leq p<\infty$. Specifically, it is called Gauss-Weierstrass semigroup. That is, for any $h_1,h_2\geq 0$, the heat kernel satisfies

$$G_{h_1+h_2} * f = G_{h_1} * G_{h_2} * f. \tag{3.38}$$

Then, we can quickly have

$$\int fG_{h_1+h_2} * g\mathrm{d}\mathbf{x} = \int (G_{h_2} * f)(G_{h_1} * g)\mathrm{d}\mathbf{x} = \int (G_{h_1} * f)(G_{h_2} * g)\mathrm{d}\mathbf{x}. \quad (3.39)$$

Now, we give the estimate for $\delta\mathcal{D}_h(\chi_1^1, \chi_2^1, \chi_1^0, \chi_2^0, \xi)$. For any $\xi \in C_s^\infty(\widehat{\Omega}, \mathbb{R}^d)$ and any $n \in \{1, \ldots, N\}$, using integration by parts and (3.39), we have

$$\begin{aligned}
&-\frac{1}{\gamma_{LV}}\delta\mathcal{D}_h(\chi_1^n, \chi_2^n, \chi_1^{n-1}, \chi_2^{n-1}, \xi)\\
=&\frac{1}{\sqrt{h}}\int \xi\cdot\nabla\chi_1^n G_h * (\chi_2^n - \chi_2^{n-1}) + (\chi_1^n - \chi_1^{n-1})G_h * (\xi\cdot\nabla\chi_2^n)\mathrm{d}\mathbf{x}\\
=&\frac{1}{\sqrt{h}}\int [G_{h/2} * (\xi\cdot\nabla\chi_1^n)][G_{h/2} * (\chi_2^n - \chi_2^{n-1})]\mathrm{d}\mathbf{x}\\
&+\frac{1}{\sqrt{h}}\int [G_{h/2} * (\chi_1^n - \chi_1^{n-1})][G_{h/2} * (\xi\cdot\nabla\chi_2^n)]\mathrm{d}\mathbf{x}\\
=&J_1 + J_2
\end{aligned}$$

where

$$\begin{aligned}
J_1 =&\frac{1}{\sqrt{h}}\int [\nabla G_{h/2} * (\xi\chi_1^n)][G_{h/2} * (\chi_2^n - \chi_2^{n-1})] - [G_{h/2} * ((\nabla\cdot\xi)\chi_1^n)][G_{h/2} * (\chi_2^n - \chi_2^{n-1})]\mathrm{d}\mathbf{x},\\
J_2 =&\frac{1}{\sqrt{h}}\int [G_{h/2} * (\chi_1^n - \chi_1^{n-1})][\nabla G_{h/2} * (\xi\chi_2^n)] - [G_{h/2} * (\chi_1^n - \chi_1^{n-1})][G_{h/2} * ((\nabla\cdot\xi)\chi_2^n)]\mathrm{d}\mathbf{x}.
\end{aligned}$$

Using the Cauchy-Schwartz inequality, we obtain

$$\begin{aligned}
|J_1|^2 \lesssim&\left(\frac{1}{\sqrt{h}}\int [\nabla G_{h/2} * (\xi\chi_1^n)][G_{h/2} * (\chi_2^n - \chi_2^{n-1})]\mathrm{d}\mathbf{x}\right)^2\\
&+ ||\nabla\xi||_\infty^2\left(\frac{1}{\sqrt{h}}\int [G_{h/2} * (\chi_1^n)][G_{h/2} * (\chi_2^n - \chi_2^{n-1})]\mathrm{d}\mathbf{x}\right)^2\\
\leq&\int \frac{1}{\sqrt{h}}[\nabla G_{h/2} * (\xi\chi_1^n)]^2\mathrm{d}\mathbf{x}\int \frac{1}{\sqrt{h}}[G_{h/2} * (\chi_2^n - \chi_2^{n-1})]^2\mathrm{d}\mathbf{x}\\
&+ ||\nabla\xi||_\infty^2\int \frac{1}{\sqrt{h}}[G_{h/2} * (\chi_1^n)]^2\mathrm{d}\mathbf{x}\int \frac{1}{\sqrt{h}}[G_{h/2} * (\chi_2^n - \chi_2^{n-1})]^2\mathrm{d}\mathbf{x}, \quad (3.40)
\end{aligned}$$

and

$$\begin{aligned}
|J_2|^2 \lesssim&\int \frac{1}{\sqrt{h}}[\nabla G_{h/2} * (\xi\chi_2^n)]^2\mathrm{d}\mathbf{x}\int \frac{1}{\sqrt{h}}[G_{h/2} * (\chi_1^n - \chi_1^{n-1})]^2\mathrm{d}\mathbf{x}\\
&+ ||\nabla\xi||_\infty^2\int \frac{1}{\sqrt{h}}[G_{h/2} * (\chi_2^n)]^2\mathrm{d}\mathbf{x}\int \frac{1}{\sqrt{h}}[G_{h/2} * (\chi_1^n - \chi_1^{n-1})]^2\mathrm{d}\mathbf{x}.
\end{aligned}$$

Before we estimate J_1, J_2, we denote

$$\mu_1^n = \int \frac{1}{\sqrt{h}}[G_{h/2} * (\chi_1^n - \chi_1^{n-1})]^2 \mathrm{d}\mathbf{x},$$
$$\mu_2^n = \int \frac{1}{\sqrt{h}}[G_{h/2} * (\chi_2^n - \chi_2^{n-1})]^2 \mathrm{d}\mathbf{x}.$$

It is easy to check that $\mu_2^n = \mu_1^n$ from $(\chi_1^n, \chi_2^n) \in \mathcal{B}$. Note that the first integral term in the right-hand side of (3.40) can be estimated via

$$\begin{aligned}
&\int \frac{1}{\sqrt{h}}[\nabla G_{h/2} * (\xi\chi_1^n)]^2 \mathrm{d}\mathbf{x} \\
=&\int \frac{1}{\sqrt{h}} \left| \int \nabla G_{h/2}(\mathbf{z}) \cdot (\xi(\mathbf{x}+\mathbf{z})\chi_1^n(\mathbf{x}+\mathbf{z})) \mathrm{d}\mathbf{z} \right| |\nabla G_{h/2} * (\xi\chi_1^n)| \mathrm{d}\mathbf{x} \\
=&\int \frac{1}{\sqrt{h}} \left| \int \nabla G_{h/2}(\mathbf{z}) \cdot (\xi(\mathbf{x}+z)\chi_1^n(\mathbf{x}+\mathbf{z}) - \xi(\mathbf{x})\chi_1^n(\mathbf{x}+\mathbf{z}) + \xi(\mathbf{x})\chi_1^n(\mathbf{x}+\mathbf{z})) dz \right| \\
&|\nabla G_{h/2} * (\xi\chi_1^n)| \mathrm{d}\mathbf{x} \\
\leq& K_1 + K_2
\end{aligned}$$

where

$$\begin{aligned}
K_1 &= \int \frac{1}{\sqrt{h}} \int \left| \nabla G_{h/2}(\mathbf{z}) \cdot (\xi(\mathbf{x}+\mathbf{z}) - \xi(\mathbf{x}))\chi_1^n(\mathbf{x}+\mathbf{z}) \right| \mathrm{d}\mathbf{z} |\nabla G_{h/2} * (\xi\chi_1^n)| \mathrm{d}\mathbf{x}, \\
K_2 &= \int \frac{1}{\sqrt{h}} \left| \xi(\mathbf{x}) \cdot \int \nabla G_{h/2}(\mathbf{z}) \chi_1^n(\mathbf{x}+\mathbf{z}) \mathrm{d}\mathbf{z} \right| |\nabla G_{h/2} * (\xi\chi_1^n)| \mathrm{d}\mathbf{x}.
\end{aligned}$$

Using $\nabla G_{h/2}(\mathbf{z}) = -\frac{\mathbf{z}}{h} G_{h/2}(\mathbf{z})$, $\frac{|\mathbf{z}|}{\sqrt{h}} G_h(\mathbf{z}) \lesssim G_{2h}(\mathbf{z})$ and $\frac{|\mathbf{z}|^2}{h} G_h(\mathbf{z}) \lesssim G_{2h}(\mathbf{z})$, we can get

$$\begin{aligned}
K_1 &= \int \int \frac{1}{\sqrt{h}} \left| G_{h/2}(\mathbf{z}) \chi_1^n(\mathbf{x}+\mathbf{z}) (\frac{\mathbf{z}}{h} \cdot (\xi(\mathbf{x}+\mathbf{z}) - \xi(\mathbf{x}))) \right| |\frac{\mathbf{z}}{h} G_{h/2} * (\xi\chi_1^n)| \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} \\
&\lesssim \int \int \frac{1}{\sqrt{h}} \left| \frac{|\mathbf{z}|^2}{h} G_{h/2}(\mathbf{z}) \chi_1^n(\mathbf{x}+\mathbf{z}) \left(\nabla \cdot \xi(\mathbf{x})\right) \right| |\frac{\mathbf{z}}{h} G_{h/2} * (\xi\chi_1^n)| \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} \\
&\lesssim \int \int ||\nabla\xi||_\infty \frac{1}{\sqrt{h}} \left| G_h(\mathbf{z}) \chi_1^n(\mathbf{x}+\mathbf{z}) \right| |\frac{1}{\sqrt{h}} G_h * (\xi\chi_1^n)| \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} \\
&\lesssim \frac{1}{h} ||\xi||_\infty ||\nabla\xi||_\infty \lesssim \frac{1}{h} ||\xi||_{W^{1,\infty}}^2
\end{aligned}$$

and from the antisymmetry of ∇G_h, $\frac{\mathbf{z}}{\sqrt{h}} G_{h/2}(\mathbf{z}) \lesssim G_h(\mathbf{z})$, $\nabla G_{h/2} * \chi_{\widehat{\Omega}} = 0$ and

the monotonicity statement (3.10) (3.11) , we have

$$\begin{aligned}
K_2 &= \int \frac{1}{\sqrt{h}} \left|\xi(\mathbf{x}) \cdot \nabla G_{h/2} * (\chi_1^n - \chi_{\widehat{\Omega}})\right| |\nabla G_{h/2} * (\xi \chi_1^n)| \mathrm{d}\mathbf{x} \\
&\lesssim ||\xi||_\infty \frac{1}{h^{\frac{3}{2}}} \int G_h * (\chi_2^n + \chi_3^n) G_h * \chi_1^n \mathrm{d}\mathbf{x} \\
&\lesssim \frac{1}{h} ||\xi||_\infty \left(\mathcal{E}_{2h}(\chi_1, \chi_2) + \mathcal{S}_{2h}(\chi_1, \chi_2)\right) < \frac{1}{h} ||\xi||_\infty (\mathcal{E}_h(\chi_1, \chi_2) + \mathcal{S}_h(\chi_1, \chi_2)).
\end{aligned}$$

The second integral term in the right-hand side of (3.40) can be simply estimated by

$$\begin{aligned}
&||\nabla \xi||_\infty^2 \int \frac{1}{\sqrt{h}} [G_{h/2} * (\chi_1^n)]^2 \mathrm{d}\mathbf{x} \int \frac{1}{\sqrt{h}} [G_{h/2} * (\chi_2^n - \chi_2^{n-1})]^2 \mathrm{d}\mathbf{x} \\
=&||\nabla \xi||_\infty^2 \mu_2 \int \frac{1}{\sqrt{h}} [\chi_1^n G_h * (\chi_1^n)] \mathrm{d}\mathbf{x} \\
\leq&||\nabla \xi||_\infty^2 \mu_2 \int \frac{1}{\sqrt{h}} \chi_1^n \mathrm{d}\mathbf{x} \lesssim \frac{1}{\sqrt{h}} ||\nabla \xi||_\infty^2 \mu_2.
\end{aligned}$$

Thus, we have

$$|J_1|^2 \lesssim \frac{1}{h} \left(||\xi||_\infty^2 (\mathcal{E}_0 + \mathcal{S}_0) + ||\xi||_{W^{1,\infty}}^2 + \sqrt{h} ||\nabla \xi||_\infty^2\right) \mu_1^n.$$

Similarly, we also have

$$|J_2|^2 \lesssim \frac{1}{h} \left(||\xi||_\infty^2 (\mathcal{E}_0 + \mathcal{S}_0) + ||\xi||_{W^{1,\infty}}^2 + \sqrt{h} ||\nabla \xi||_\infty^2\right) \mu_1^n.$$

Hence, we have

$$[\delta \mathcal{D}_h(\chi_1^n, \chi_2^n, \chi_1^{n-1}, \chi_2^{n-1}, \xi_n)]^2 \lesssim \frac{1}{h} \left(||\xi_n||_\infty^2 (\mathcal{E}_0 + \mathcal{S}_0) + ||\xi_n||_{W^{1,\infty}}^2 + \sqrt{h} ||\nabla \xi_n||_\infty^2\right) \mu_1^n. \tag{3.41}$$

Using (3.25) and summing (3.41) for n from 1 to N will lead us to (3.37).

Step 3: Estimates on $\delta \mathcal{S}_h(\chi_1, \chi_2, \xi)$. For any $(\chi_1, \chi_2) \in \mathcal{B}$ and any $\xi \in C_s^\infty(\widehat{\Omega}, \mathbb{R}^d)$, we have

$$|\delta \mathcal{S}_h(\chi_1, \chi_2, \xi)| \lesssim ||\nabla \xi||_\infty (\mathcal{S}_{2h}(\chi_1, \chi_2) + \mathcal{S}_h(\chi_1, \chi_2)). \tag{3.42}$$

Argument for (3.42): Note that from the definition of function space of ξ, we can have that $\int (\xi \cdot \nabla \chi_3) f = 0$ for any $f \in C^\infty(\widehat{\Omega}, \mathbb{R})$, then we can written

$\delta\mathcal{S}_h(\chi_1,\chi_2,\xi)$ by a symmetric form as

$$\begin{aligned}\delta\mathcal{S}_h(\chi_1,\chi_2,\xi) =& -\frac{\gamma_{SL}}{\sqrt{h}}\int \xi\cdot\nabla\chi_1 G_h * \chi_3 + \xi\cdot\nabla\chi_3 G_h * \chi_1 \\ & -\frac{\gamma_{SV}}{\sqrt{h}}\int \xi\cdot\nabla\chi_2 G_h * \chi_3 + \xi\cdot\nabla\chi_3 G_h * \chi_2.\end{aligned}$$

Then, using the similar argument in Step 1 yields (3.42).

Step 4: Construction of ξ. Analogous to [56], we have the similar statement for $\xi \in C_s^\infty(\widehat{\Omega},\mathbb{R}^d)$: For any $\mathcal{E}_0>0$, $\mathcal{S}_0>0$, $h<\frac{1}{C(\mathcal{E}_0+\mathcal{S}_0)^2}$ for some constant $C=C(d)$, and any χ_1 with $\int\chi_1 d\mathbf{x}=1$, $\operatorname{supp}\chi_1 \subset\subset \widehat{\Omega}$, there exists $\xi\in C_s^\infty(\widehat{\Omega},\mathbb{R}^d)$ with

$$\int(\nabla\cdot\xi)\chi_1 d\mathbf{x} \geq \frac{1}{2}, \tag{3.43}$$

$$||\xi||_{W^{1,\infty}} \lesssim 1+\mathcal{E}_0+\mathcal{S}_0. \tag{3.44}$$

Argument: We will follow the idea of the argument in Step 3 of the proof in Proposition 1.12 in [56]. Set $\epsilon^2=\frac{1}{C(\mathcal{E}_0+\mathcal{S}_0)^2}$ and $\chi_\epsilon := \phi_\epsilon * \chi_1$ for some mollifier $\phi_\epsilon(\mathbf{z})=\frac{1}{\epsilon^d}\phi_1(\frac{\mathbf{z}}{\epsilon})$ with $0<\phi_1<1$, $\int\phi_1 d\mathbf{z}=1$, $\phi_1\lesssim G_1$ and $\int|\nabla\phi_1|d\mathbf{x}\lesssim 1$. Then $\chi_\epsilon\in C_0^\infty(\widehat{\Omega},\mathbb{R}^d)$, denote u as the solution of

$$\Delta u = \chi_\epsilon, \tag{3.45}$$

$$\frac{\partial u}{\partial\vec{n}} = 0 \ on\ \tilde{\Gamma} \tag{3.46}$$

where $\tilde{\Gamma}$ is a smooth interface satisfies

$$dist(\tilde{\Gamma},\Gamma)=0\ a.e.$$

where $dist(A,B)$ represents the Hausdroff distance from set A to set B and Γ is interface between D_3 and $D_1\cup D_2$. Then, $u\in C_0^\infty(\widehat{\Omega},\mathbb{R}^d)$ and $\frac{\partial u}{\partial\vec{n}}=0$ $a.e.$ on Γ. We define $\xi:=\nabla u$, then from above definition, we have $\xi\in C_s^\infty(\widehat{\Omega},\mathbb{R}^d)$. Now,

we estimate $\int(\nabla\cdot\xi)\chi_1 \mathrm{d}\mathbf{x}$ as

$$\begin{aligned}\int(\nabla\cdot\xi)\chi_1 \mathrm{d}\mathbf{x} &= \int \chi_\epsilon\chi_1 \mathrm{d}\mathbf{x}\\ &= \int \chi_\epsilon(1-(1-\chi_1))\mathrm{d}\mathbf{x}\\ &= \int \chi_\epsilon \mathrm{d}\mathbf{x} - \int \chi_\epsilon(1-\chi_1)\mathrm{d}\mathbf{x}\\ &\geq 1 - \int(1-\chi_1)G_{\epsilon^2}\chi_1 \mathrm{d}\mathbf{x}.\end{aligned} \tag{3.47}$$

It is easy to see that

$$\int(1-\chi_1)G_{\epsilon^2}\chi_1 \mathrm{d}\mathbf{x} \lesssim \epsilon(\mathcal{E}_0+\mathcal{S}_0). \tag{3.48}$$

Hence, from (3.47) and (3.48), we can choose sufficiently large $C = C(d)$ such that

$$\int(\nabla\cdot\xi)\chi_1 \mathrm{d}\mathbf{x} \geq \frac{1}{2}. \tag{3.49}$$

For the proof of (3.44), we cite the result from Step 3 of the proof in Proposition 1.12 in [56]. It is exactly same when we $\widehat{\Omega}$ is large enough so that $\operatorname{supp}\chi_1 \subset\subset \widehat{\Omega}$.

Step 5:Conclusion. Now, we apply Step 4 on χ_1^n, χ_2^n to find the corresponding $\xi^n \in C_s^\infty(\widehat{\Omega}, \mathbb{R}^d)$ satisfies

$$\begin{aligned}\int(\nabla\cdot\xi^n)\chi_1 \mathrm{d}\mathbf{x} &\geq \frac{1}{2},\\ ||\xi^n||_{W^{1,\infty}} &\lesssim 1+\mathcal{E}_0+\mathcal{S}_0.\end{aligned}$$

From (3.35), (3.37) and (3.42), we can get

$$\begin{aligned}\int_0^T \lambda_h^2 dt = h\sum_{n=1}^N \lambda_n^2 &\lesssim \sup_n ||\xi||_{W^{1,\infty}}^2(T\mathcal{E}_0^2+1+\mathcal{E}_0^2+T\mathcal{S}_0^2)h\\ &\lesssim (1+T)(1+(\mathcal{E}_0+\mathcal{S}_0)^4)h.\end{aligned}$$

□

3.5 Compactness

In this section, we prove the compactness of the approximate solutions (χ_1^h, χ_2^h). That is, for a sequence $h \searrow 0$, we can have a sequence for the approximate solutions (χ_1^h, χ_2^h), we prove the existence of the limit of the sequence. We will first introduce the proposition states the compactness followed by two lemmas for the proof of the proposition. The proofs are similar to the proof in Lemma 2.5 in [56]. However, our energy functional is a little different from the energy functional in [56].

Proposition 3.5.1 (Compactness)**.** *There exists a sequence $h \searrow 0$ and a limit $(\chi_1, \chi_2) : (0,T) \times \widehat{\Omega} \to \{0,1\}^2$ such that*

$$(\chi_1^h, \chi_2^h) \to (\chi_1, \chi_2) \quad a.e. \quad in \quad (0,T) \times \widehat{\Omega} \tag{3.50}$$

and $\mathcal{E}(\chi_1, \chi_2) + \mathcal{S}(\chi_1, \chi_2) \leq \mathcal{E}_0 + \mathcal{S}_0$ with $(\chi_1, \chi_2) \in \mathcal{B}$ for a.e. $t \in (0,T)$.

Proof. From Lemma 3.5.1 and Lemma 3.5.2, we can have

$$\int_0^T \int |\chi_1^h(\mathbf{x} + \delta e, t + \tau) - \chi_1^h(\mathbf{x}, t)| \mathrm{d}\mathbf{x} dt \lesssim (1+T)(\delta + \sqrt{h} + \tau)(\mathcal{E}_0 + \mathcal{S}_0). \tag{3.51}$$

Then, we directly apply the proof of Proposition 2.1 in [56] which is adaptation of Riesz-Kolmogorov L^p-compactness theorem. □

Now, we first give an estimate for $\int_0^T \int |\chi_1^h(\mathbf{x}+\delta e, t) - \chi_1^h(\mathbf{x}, t)| \mathrm{d}\mathbf{x} dt$ to show that the approximate solutions are almost BV in space. Note that (χ_1^h, χ_2^h) satisfy $\int_0^T \int |\chi_2^h(\mathbf{x} + \delta e, t) - \chi_2^h(\mathbf{x}, t)| \mathrm{d}\mathbf{x} dt = \int_0^T \int |\chi_1^h(\mathbf{x} + \delta e, t) - \chi_1^h(\mathbf{x}, t)| \mathrm{d}\mathbf{x} dt$.

Lemma 3.5.1. *The approximate solutions (χ_1^h, χ_2^h) satisfy*

$$\int_0^T \int |\chi_1^h(\mathbf{x} + \delta e, t) - \chi_1^h(\mathbf{x}, t)| \mathrm{d}\mathbf{x} dt \lesssim (1+T)(\delta + \sqrt{h})(\mathcal{E}_0 + \mathcal{S}_0) \tag{3.52}$$

for any $\delta > 0$ and $e \in S^{d-1}$.

Proof. We write

$$\begin{aligned}
&\chi_1^h(\mathbf{x}+\delta e,t)-\chi_1^h(\mathbf{x},t)\\
=&\chi_1^h(\mathbf{x}+\delta e,t)-(G_h*\chi_1^h)(\mathbf{x}+\delta e,t)-(\chi_1^h(\mathbf{x},t)-(G_h*\chi_1^h)(\mathbf{x},t))\\
&+(G_h*\chi_1^h)(\mathbf{x}+\delta e,t)-(G_h*\chi_1^h)(\mathbf{x},t)
\end{aligned}$$

Then, we have

$$\begin{aligned}
&\int_0^T\int|\chi_1^h(\mathbf{x}+\delta e,t)-\chi_1^h(\mathbf{x},t)|\mathrm{d}\mathbf{x}dt\\
\le& I_1+I_2
\end{aligned}$$

where $I_1=2\int_0^T\int|G_h*\chi_1^h-\chi_1^h|\mathrm{d}\mathbf{x}dt$ and $I_2=\int_0^T\int|(G_h*\chi_1^h)(\mathbf{x}+\delta e,t)-(G_h*\chi_1^h)(\mathbf{x},t)|\mathrm{d}\mathbf{x}dt$. Since $\chi_1^h\in\{0,1\}$ *a.e.* in $\widehat{\Omega}$, we have

$$\begin{aligned}
I_1=&2\int_0^T\int|\chi_{\widehat{\Omega}}G_h*\chi_1^h-\chi_1^hG_h*\chi_{\widehat{\Omega}}|\mathrm{d}\mathbf{x}dt\\
=&2\int_0^T\int|(\chi_{\widehat{\Omega}}-\chi_1^h)G_h*\chi_1^h-\chi_1^hG_h*(\chi_{\widehat{\Omega}}-\chi_1^h)|\mathrm{d}\mathbf{x}dt\\
\le&2\left(\int_0^T\int|(\chi_2^h+\chi_3)G_h*\chi_1^h)|\mathrm{d}\mathbf{x}dt+\int_0^T\int|(\chi_1^hG_h*(\chi_2^h+\chi_3)|\mathrm{d}\mathbf{x}dt\right)\\
\lesssim&\int_0^T\sqrt{h}(\mathcal{E}_h+\mathcal{S}_h)dt\lesssim(1+T)\sqrt{h}(\mathcal{E}_0+\mathcal{S}_0).
\end{aligned}$$

For I_2, we have

$$\begin{aligned}
I_2\lesssim&\delta\int_0^T\int|\nabla G_h*\chi_1^h|\mathrm{d}\mathbf{x}dt\\
=&\delta\int_0^T\int\left|\int\nabla G_h(\mathbf{z})*(\chi_1^h(\mathbf{x}-\mathbf{z})-\chi_1^h(\mathbf{x}))\mathrm{d}\mathbf{z}\right|\mathrm{d}\mathbf{x}dt\\
\lesssim&\frac{\delta}{\sqrt{h}}\int_0^T\int G_{2h}(\mathbf{z})\int\left|\chi_1^h(\mathbf{x}-\mathbf{z})-\chi_1^h(\mathbf{x})\right|\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z}dt\\
=&\frac{\delta}{\sqrt{h}}\int_0^T\int G_{2h}(\mathbf{z})\int\left|\chi_1^h(\mathbf{x}-\mathbf{z})(1-\chi_1^h(\mathbf{x}))-\chi_1^h(\mathbf{x})(1-\chi_1^h(\mathbf{x}-\mathbf{z}))\right|\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z}dt\\
=&\frac{\delta}{\sqrt{h}}\int_0^T\int G_{2h}(\mathbf{z})\int\chi_1^h(\mathbf{x}-\mathbf{z})(1-\chi_1^h(\mathbf{x}))+\chi_1^h(\mathbf{x})(1-\chi_1^h(\mathbf{x}-\mathbf{z}))\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z}dt\\
\lesssim&\delta(\mathcal{E}_{2h}+\mathcal{S}_{2h})\lesssim\delta(\mathcal{E}_0+\mathcal{S}_0).
\end{aligned}$$

Combining the estimates for I_1 and I_2 yields (3.52). □

Then, we give an estimate for $\int_\tau^T \int |\chi_1^h(t) - \chi_1^h(t-\tau)| \mathrm{d}\mathbf{x} dt$ to show that the approximate solutions are almost BV in time. It is similar as we mentioned before that (χ_1^h, χ_2^h) satisfy $\int_\tau^T \int |\chi_2^h(t) - \chi_2^h(t-\tau)| \mathrm{d}\mathbf{x} dt = \int_\tau^T \int |\chi_1^h(t) - \chi_1^h(t-\tau)| \mathrm{d}\mathbf{x} dt$.

Lemma 3.5.2. *The approximate solutions (χ_1^h, χ_2^h) satisfy*

$$\int_\tau^T \int |\chi_1^h(t) - \chi_1^h(t-\tau)| \mathrm{d}\mathbf{x} dt \lesssim (1+T)(\tau + \sqrt{h})(\mathcal{E}_0 + \mathcal{S}_0) \tag{3.53}$$

for any $\tau > 0$.

Proof. Follow the idea in [56], we denote $\tau = \alpha\sqrt{h}$ and claim that we only need to prove

$$\int_\tau^T \int |\chi_1^h(t) - \chi_1^h(t-\tau)| \mathrm{d}\mathbf{x} dt \lesssim (1+T)(\mathcal{E}_0 + \mathcal{S}_0)\tau \tag{3.54}$$

for any $\alpha \in [1, 2]$. Indeed, if $\alpha \in (0, 1)$, we can apply (3.54) twice, once for $\tau = \sqrt{h}$ and once for $\tau = (1+\alpha)\sqrt{h}$ and then we have (3.53). If $\alpha > 2$, we iterate (3.54) to get (3.53).

For simplicity, we assume $T = L\tau$, $\tau = Kh$ and $N = LK$ where L, K, N are all natural numbers. Then

$$\begin{aligned}\int_\tau^T \int |\chi_1^h(t) - \chi_1^h(t-\tau)| \mathrm{d}\mathbf{x} dt &= h \sum_{k=0}^{K-1} \sum_{l=1}^{L} \int \left| \chi_1^{Kl+k} - \chi_1^{K(l-1)+k} \right| \mathrm{d}\mathbf{x} \\ &= \frac{1}{K} \sum_{k=0}^{K-1} \tau \sum_{l=1}^{L} \int \left| \chi_1^{Kl+k} - \chi_1^{K(l-1)+k} \right| \mathrm{d}\mathbf{x}. \end{aligned} \tag{3.55}$$

Now, consider

$$\begin{aligned}&\left| \chi_1^{Kl+k} - \chi_1^{K(l-1)+k} \right| \\ =&(\chi_1^{Kl+k} - \chi_1^{K(l-1)+k}) G_h * (\chi_1^{Kl+k} - \chi_1^{K(l-1)+k}) \\ &- (\chi_1^{Kl+k} - \chi_1^{K(l-1)+k})(\chi_1^{Kl+k} - \chi_1^{K(l-1)+k} - G_h * (\chi_1^{Kl+k} - \chi_1^{K(l-1)+k})).\end{aligned}$$

Then we have

$$
\begin{aligned}
&(\chi_1^{Kl+k}-\chi_1^{K(l-1)+k})G_h*(\chi_1^{Kl+k}-\chi_1^{K(l-1)+k})\\
=&\left(\sum_{n=K(l-1)+k+1}^{Kl+k}(\chi_1^n-\chi_1^{n-1})\right)G_h*\left(\sum_{n=K(l-1)+k+1}^{Kl+k}(\chi_1^n-\chi_1^{n-1})\right)\\
\le&\left(\sum_{n=K(l-1)+k+1}^{Kl+k}\left((\chi_1^n-\chi_1^{n-1})G_h*(\chi_1^n-\chi_1^{n-1})\right)^{1/2}\right)^2\\
\le&K\sum_{n=K(l-1)+k+1}^{Kl+k}(\chi_1^n-\chi_1^{n-1})G_h*(\chi_1^n-\chi_1^{n-1})
\end{aligned}
$$

and

$$
\begin{aligned}
&(\chi_1^{Kl+k}-\chi_1^{K(l-1)+k})(\chi_1^{Kl+k}-\chi_1^{K(l-1)+k}-G_h*(\chi_1^{Kl+k}-\chi_1^{K(l-1)+k}))\\
\le&\left|\chi_1^{Kl+k}-G_h*\chi_1^{Kl+k}\right|+\left|\chi_1^{K(l-1)+k}-G_h*\chi_1^{K(l-1)+k}\right|\\
=&(1-\chi_1^{Kl+k})G_h*\chi_1^{Kl+k}+\chi_1^{Kl+k}G_h*(1-\chi_1^{Kl+k})\\
&+(1-\chi_1^{K(l-1)+k})G_h*\chi_1^{K(l-1)+k}+\chi_1^{K(l-1)+k}G_h*(1-\chi_1^{K(l-1)+k}).
\end{aligned}
$$

Using the fact that $\chi_1^n-\chi_1^{n-1}=-(\chi_2^{n-1}-\chi_2^{n-1})$, we can have

$$
\begin{aligned}
\sum_{l=1}^{L}\int\left|\chi_1^{Kl+k}-\chi_1^{K(l-1)+k}\right|\mathrm{d}\mathbf{x}\lesssim&\sqrt{h}K\sum_{n=1}^{N}-\mathcal{D}_h(\chi_1^n-\chi_1^{n-1},\chi_2^n-\chi_2^{n-1})\\
&+4L\max_n\int(1-\chi_1^n)G_h\chi_1^n\mathrm{d}\mathbf{x}\\
\lesssim&\alpha(\mathcal{E}_0+\mathcal{S}_0)+\frac{4T}{\alpha}(\mathcal{E}_0+\mathcal{S}_0)\\
\lesssim&(1+T)(\mathcal{E}_0+\mathcal{S}_0)
\end{aligned}
$$

where the penultimate step comes from the energy dissipation estimate (3.25). Substituting above estimates into (3.55) yields (3.54) which is equivalent to (3.54) as we discussed before. □

3.6 Convergence

In this section, we consider the convergence of each terms in (3.29). We will directly apply Proposition 3.1 of [56] to our case to have the convergence of $\delta\mathcal{E}_h$, apply Proposition 1.18 of [55] to have the convergence of $\delta\mathcal{D}_h$ and prove that we can use Proposition 3.1 of [56] to get the convergence of $\delta\mathcal{S}_h$. We first cite the results in [56] in the following proposition.

Proposition 3.6.1 (Liquid-vapor interface energy and mean curvature; Proposition 3.1 in [56]). *Let* $(\chi_1^h, \chi_2^h), (\chi_1, \chi_2) : (0,T) \times \widehat{\Omega} \to \{0,1\}^2$ *be such that* $(\chi_1^h, \chi_2^h) \in \mathcal{B}$ *and* $(\chi_1, \chi_2) \in \mathcal{B}$ *and* $\mathcal{E}(\chi_1(t), \chi_2(t)) < \infty$ *for a.e.* t. *Let*

$$\chi^h \to \chi \text{ a.e. in } (0,T) \times \widehat{\Omega}, \tag{3.56}$$

and furthermore assume that

$$\int_0^T \mathcal{E}_h(\chi_1^h, \chi_2^h)dt \to \int_0^T \mathcal{E}(\chi_1, \chi_2)dt. \tag{3.57}$$

Then, for any $\xi \in C_s^\infty((0,T) \times \widehat{\Omega}, \mathbb{R}^n)$, *we have*

$$\lim_{h\to 0} \int_0^T \delta\mathcal{E}_h(\chi^h, \xi)dt \tag{3.58}$$

$$= \frac{\gamma_{LV}}{\sqrt{\pi}} \int_0^T \int (\nabla \cdot \xi - \nu_1 \cdot \nabla\xi\nu_1)\frac{1}{2}(|\nabla\chi_1| + |\nabla\chi_2| - |\nabla(\chi_1 + \chi_2)|)dt. \tag{3.59}$$

Then, we derive the following proposition to describe the convergence of the dissipation $\int_0^T \mathcal{D}_h(\chi_1^h(t), \chi_2^h(t), \chi_1^h(t-h), \chi_2^h(t-h), \xi(t))dt$ when $h \searrow 0$ based on Proposition 1.18 in [55].

Proposition 3.6.2 (Dissipation and normal velocity). *Let*

$$(\chi_1^h, \chi_2^h), (\chi_1, \chi_2) : (0,T) \times \widehat{\Omega} \to \{0,1\}^2$$

be such that $(\chi_1^h, \chi_2^h) \in \mathcal{B}$, $(\chi_1, \chi_2) \in \mathcal{B}$, *and* $\mathcal{E}(\chi_1(t), \chi_2(t)) < \infty$ *for a.e.* t. *Let*

$$\chi^h \to \chi \text{ a.e. in } (0,T) \times \widehat{\Omega}, \tag{3.60}$$

and furthermore assume that

$$\int_0^T \mathcal{E}_h(\chi_1^h, \chi_2^h)dt \to \int_0^T \mathcal{E}(\chi_1, \chi_2)dt. \tag{3.61}$$

There exists a function $V : (0,T) \times \widehat{\Omega} \to \mathbb{R}$ *and* $V \in L^2(|\nabla\chi_1|dt)$ *which is a normal velocity in the sense of (3.21). Then, for any* $\xi \in C_s^\infty((0,T) \times \widehat{\Omega}, \mathbb{R}^d)$*, we have*

$$\begin{aligned}&\lim_{h\to 0}\int_0^T \delta\mathcal{D}_h(\chi_1^h(t), \chi_2^h(t), \chi_1^h(t-h), \chi_2^h(t-h), \xi(t))dt\\ =&\frac{1}{\sqrt{\pi}}\int_0^T \int V_1\xi\cdot\nu_1\frac{1}{2}(|\nabla\chi_1| + |\nabla\chi_2| - |\nabla(\chi_1+\chi_2)|)dt.\end{aligned} \tag{3.62}$$

Proof. Note that from the definition of $\mathcal{D}_h$, we can have

$$\begin{aligned}&\mathcal{D}_h(\chi_1^h(t), \chi_2^h(t), \chi_1^h(t-h), \chi_2^h(t-h), \xi(t))\\ =&\mathcal{D}_h(\chi_1^h(t), \chi_2^h(t)+\chi_3, \chi_1^h(t-h), \chi_2^h(t-h)+\chi_3, \xi(t))\end{aligned}$$

for any $t \in (0,T)$ and $\xi \in C_s^\infty((0,T) \times \widehat{\Omega}, \mathbb{R}^n)$. Then, we also have

$$\begin{aligned}&\delta\mathcal{D}_h(\chi_1^h(t), \chi_2^h(t), \chi_1^h(t-h), \chi_2^h(t-h), \xi(t))\\ =&\delta\mathcal{D}_h(\chi_1^h(t), \chi_2^h(t)+\chi_3, \chi_1^h(t-h), \chi_2^h(t-h)+\chi_3, \xi(t))\end{aligned}$$

for any $t \in (0,T)$ and $\xi \in C_s^\infty((0,T) \times \widehat{\Omega}, \mathbb{R}^n)$. From Proposition 1.18 in [55], we can have

$$\begin{aligned}&\lim_{h\to 0}\int_0^T \delta\mathcal{D}_h(\chi_1^h(t), \chi_2^h(t)+\chi_3, \chi_1^h(t-h), \chi_2^h(t-h)+\chi_3, \xi(t))dt\\ =&\frac{\gamma_{LV}}{\sqrt{\pi}}\int_0^T \int V_1\xi\cdot\nu_1\frac{1}{2}|\nabla\chi_1|dt\\ =&\frac{\gamma_{LV}}{\sqrt{\pi}}\int_0^T \int V_1\xi\cdot\nu_1\frac{1}{2}(|\nabla\chi_1| + |\nabla\chi_2| - |\nabla(\chi_1+\chi_2)|)dt\\ &+\frac{\gamma_{LV}}{\sqrt{\pi}}\int_0^T \int V_1\xi\cdot\nu_1\frac{1}{2}(|\nabla\chi_1| + |\nabla\chi_3| - |\nabla(\chi_1+\chi_3)|)dt.\end{aligned}$$

We claim that the second term on the right hand side is 0. Indeed, when we consider $(|\nabla\chi_1| + |\nabla\chi_3| - |\nabla(\chi_1+\chi_3)|)dt$, ν_1 represents the normal vector to the solid interface. From the definition of $C_s^\infty((0,T) \times \widehat{\Omega}, \mathbb{R}^n)$, we can have $\xi \cdot \nu_1 = 0$ *a.e.* which will indicate the second term is 0. Then, we will have (3.62). □

We mention that the proof of Proposition 3.1 of [56] highly depends on the symmetric formulation of the first variation so that they can use the same technical as we used in the estimate for $\delta\mathcal{E}_h$ in Step 1 in the proof of Proposition 3.4.1. For the solid interface energy term in (3.29), since the first variation (3.28) is not a symmetric form. it is not trivial to directly apply the Proposition 3.1 of [56]. However, we can use the feature of function space(i.e. $C_s^\infty((0,T)\times\widehat{\Omega},\mathbb{R}^n)$) of the test field ξ as we use in Step 3 in the proof of Proposition 3.4.1.

Proposition 3.6.3 (Solid interface energy and conormal vector)**.** *Let* (χ_1^h,χ_2^h), $(\chi_1,\chi_2):(0,T)\times\widehat{\Omega}\to\{0,1\}^2$ *be such that* $(\chi_1^h,\chi_2^h)\in\mathcal{B}$ *and* $(\chi_1,\chi_2)\in\mathcal{B}$ *and* $\mathcal{E}(\chi_1(t),\chi_2(t))<\infty$ *for a.e.* t*. Let*

$$\chi^h\to\chi \text{ a.e. in } (0,T)\times\widehat{\Omega}, \tag{3.63}$$

and furthermore assume that

$$\int_0^T\mathcal{S}_h(\chi_1^h,\chi_2^h)dt\to\int_0^T\mathcal{S}(\chi_1,\chi_2)dt. \tag{3.64}$$

Then, for any $\xi\in C_s^\infty((0,T)\times\widehat{\Omega},\mathbb{R}^n)$*, we have*

$$\begin{aligned}&\lim_{h\to0}\int_0^T\delta\mathcal{S}_h(\chi^h,\xi)dt\\ =&\frac{\gamma_{SV}}{\sqrt{\pi}}\int_0^T\int(\nabla\cdot\xi-\nu_3\cdot\nabla\xi\nu_3)\frac{1}{2}(|\nabla\chi_2|+|\nabla\chi_3|-|\nabla(\chi_2+\chi_3)|)dt\\ &+\frac{\gamma_{SL}}{\sqrt{\pi}}\int_0^T\int(\nabla\cdot\xi-\nu_3\cdot\nabla\xi\nu_3)\frac{1}{2}(|\nabla\chi_1|+|\nabla\chi_3|-|\nabla(\chi_1+\chi_3)|)dt.\end{aligned} \tag{3.65}$$

Proof. Note that from the definition of function space of ξ, we can have that $\int(\xi\cdot\nabla\chi_3)f=0$ for any $f\in C^\infty(\widehat{\Omega},\mathbb{R})$, then we can written $\delta\mathcal{S}_h(\chi_1,\chi_2,\xi)$ into $\delta\widehat{\mathcal{S}}_h(\chi_1,\chi_2,\xi)$ by a symmetric form as

$$\begin{aligned}\delta\widehat{\mathcal{S}}_h(\chi_1,\chi_2,\xi)=&-\frac{\gamma_{SL}}{\sqrt{h}}\int\xi\cdot\nabla\chi_1G_h*\chi_3+\xi\cdot\nabla\chi_3G_h*\chi_1\\ &-\frac{\gamma_{SV}}{\sqrt{h}}\int\xi\cdot\nabla\chi_2G_h*\chi_3+\xi\cdot\nabla\chi_3G_h*\chi_2.\end{aligned}$$

Then, from Proposition 3.1 of [56] , we can have

$$\begin{aligned}
&\lim_{h\to 0}\int_0^T \delta\widehat{\mathcal{S}}_h(\chi^h,\xi)dt\\
=&\frac{\gamma_{SV}}{\sqrt{\pi}}\int_0^T\int(\nabla\cdot\xi-\nu_3\cdot\nabla\xi\nu_3)\frac{1}{2}(|\nabla\chi_2|+|\nabla\chi_3|-|\nabla(\chi_2+\chi_3)|)dt\\
&+\frac{\gamma_{SL}}{\sqrt{\pi}}\int_0^T\int(\nabla\cdot\xi-\nu_3\cdot\nabla\xi\nu_3)\frac{1}{2}(|\nabla\chi_1|+|\nabla\chi_3|-|\nabla(\chi_1+\chi_3)|)dt
\end{aligned}$$

which will lead us to (3.65). □

Chapter 4

An efficient boundary integral scheme for the threshold dynamics method via NUFFT

In this chapter, we propose an efficient algorithm for the threshold dynamics method with an NUFFT based heat equation solver. Standard potential theory [37] shows that the solution to the pure initial value problem of the heat equation is given by the convolution of the heat kernel $G(\mathbf{x}, \mathbf{y}; t)$ with the initial data u_0. The convolution integral on the whole domain can be converted to a boundary integral by the divergence theorem or Green's theorem. We first show that an equispaced grid is sufficient to obtain a spectral approximation for the heat kernel at a fixed time $t = \Delta t$. We then apply NUFFT [26, 27, 42] to evaluate the boundary integral at a set of target points. For threshold dynamics, the interface moves within a band around previous positions with a bandwidth of $O(\sqrt{\Delta t})$ due to the diffusion process. Thus we may choose the target points to be a set of pN points with N the number of points on the interface and p the number of points along each normal direction (p is usually about 10–20).

Finally, we use standard root finding procedure for thresholding. The whole algorithm is spectrally accurate in space. It is also very efficient when Δt is not very small due to the following reasons. First, there is no need to discretize the whole computational domain and only pN points near the interface are needed in physical space. Second, NUFFT has the same $O(N \log N)$ complexity as the standard FFT. Third, the root finding procedure converges very quickly, usually in about 4–6 iterations to full double precision, due to the smoothing effect of the diffusion process. Our method is also mesh free since it tracks the motion of a set of points only.

There have been many excellent numerical methods for threshold dynamics. Instead of trying to present a comprehensive review of the existing methods, we would like to discuss other NUFFT based methods and clarify the differences between our method and those methods. In [75, 77], efficient algorithms have been developed for diffusion-generated motion by mean curvature. The algorithms assume the Neumann boundary conditions on a rectangular box, represent the solution to the heat equation via a cosine series, and then use the NUFFT to solve the heat equation on an adaptive grid.

In [43], a spectral approximation of the heat kernel is constructed with uniform absolute error for all $t > \delta$. Due to this much stricter requirement, the spectral approximation constructed in [43] needs an adaptive nonuniform nodes in the Fourier space with dyadic refinement towards the origin. In [59], a fast solver has been developed to solve the initial value problem of the heat equation using NUFFT based on the adaptive nonuniform approximation of the heat kernel in [43]. The paper assumes that the initial data is smooth and requires time marching since it aims at finding the solution for all $t > \delta$.

Due to the thresholding step, we only need to find the solution to the heat equation at $t = \Delta t$ in threshold dynamics method since the diffusion process

starts anew for each time step. Thus we only need to approximate the heat kernel at $t = \Delta t$. Our analysis shows that an equispaced grid in the Fourier space is sufficient to approximate the heat kernel with spectral accuracy. That is, there is no need to use adaptive Gaussian nodes in the Fourier space as in [43], and one does not need to impose the Neumann condition on an artificial rectangular box in order to be able to use the cosine series for the representation of the solution as in [77]. The computational advantages are two fold. First, we need less number of Fourier modes in the approximation of the heat kernel. Second, we only need type-1 and type-2 NUFFTs instead of much more expensive type-3 NUFFT for the evaluation of the solution to the heat equation in an arbitrary set of points in physical space.

Another difference is that [59] assumes that the initial data is smooth, while we are dealing with discontinuous piecewise constant initial data. Thus the direct application of the method in [59] would decrease the order of accuracy sharply due to the slow decaying of the Fourier transform of the initial data. We have used the divergence theorem or Green's theorem to reduce the volume or area integral to a boundary integral, which reduces the dimension of the problem by one and retains the high-order feature of the algorithm for smooth or piecewise-smooth boundaries.

As compared with the methods in [75, 77], we carry out the computation either on the interface or near the interface in both the solving stage and the thresholding stage. This makes our algorithm more efficient and easier to achieve high-order in space, which is appealing for certain applications in computational fluid dynamics. Our scheme may be extended to deal with topological changes and multiphase flows such as the wetting problems as in [75, 77].

This chapter proceeds as the follows: In Section 4.1, we present an NUFFT based solver for the initial value problem of the heat equation with piecewise constant

initial data in both two and three dimensions. In Section 4.2, we discuss the thresholding step by the root finding procedure. Section 4.3 contains several numerical examples demonstrating the performance of the numerical algorithm.

4.1 NUFFT based solver for the heat equation in free space

In this section, we consider solving the pure initial value problem of the heat equation:

$$\begin{aligned} u_t &= \Delta u, \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}) \end{aligned} \tag{4.1}$$

with the initial data u_0 the indicator function of some bounded domain $D \in \mathbb{R}^d$ $(d = 2, 3)$. , i.e.,

$$u_0(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in D, \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

We assume that the boundary ∂D of D is a smooth manifold which is homomorphic to S^{d-1} (i.e., the unit circle in $\mathbb{R}^2$ and unit sphere in $\mathbb{R}^3$). We are interested in calculating $u(\mathbf{x}, \Delta t)$ for a given set of target points $\mathbf{x}$ (say, in the neighborhood of ∂D) at some $\Delta t < 1$. The Green's function (or the fundamental solution) of the heat equation in $\mathbb{R}^d$ is given by the formula

$$G_d(\mathbf{x}, \mathbf{y}; t) = \frac{1}{(4\pi t)^{d/2}} e^{-\|\mathbf{x}-\mathbf{y}\|^2/4t}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \tag{4.3}$$

And $u(\mathbf{x}, \Delta t)$ is given by the formula

$$u(\mathbf{x}, \Delta t) = \int \cdots \int_{\mathbb{R}^d} G_d(\mathbf{x}, \mathbf{y}; \Delta t) u_0(\mathbf{y}) d\mathbf{y} = \int \cdots \int_D G_d(\mathbf{x}, \mathbf{y}; \Delta t) d\mathbf{y}, \tag{4.4}$$

where the second equality follows from (4.2).

4.1.1 Fourier spectral approximation of the heat kernel for a fixed time

It is well known that G_d admits the following Fourier representation

$$G_d(\mathbf{x},\mathbf{y};t)=\frac{1}{(2\pi)^d}\int\cdots\int_{\mathbb{R}^d} e^{-\|\mathbf{k}\|^2 t+i\mathbf{k}\cdot(\mathbf{x}-\mathbf{y})}d\mathbf{k},\quad \mathbf{k}\in\mathbb{R}^d. \tag{4.5}$$

We now try to approximate the integral in (4.5) by a discrete summation. We first consider the approximation of the one dimensional heat kernel

$$G_1(x,\Delta t)=\frac{1}{\sqrt{4\pi\Delta t}}e^{-x^2/4\Delta t}=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{-k^2\Delta t+ikx}dk. \tag{4.6}$$

Theorem 4.1.1 (Fourier Spectral Approximation of the 1D Heat Kernel for a Fixed Time)**.** *Suppose that $\epsilon < 1/2$ is the prescribed accuracy. Let $h = \min\left(\frac{\pi}{R}, \frac{\pi}{2\sqrt{\Delta t|\ln\epsilon|}}\right)$, and $M=\frac{1}{h}\sqrt{\frac{\ln(\sqrt{\pi}\epsilon/2h\sqrt{\Delta t})}{\Delta t}}$. Then for all $|x|\le R$,*

$$\left|G_1(x,\Delta t)-\frac{h}{2\pi}\sum_{m=-M}^{M-1}e^{-m^2h^2\Delta t+imhx}\right|\le\frac{4\epsilon}{\sqrt{4\pi\Delta t}}. \tag{4.7}$$

Proof. Let $\hat{f}$ be the Fourier transform of f, that is,

$$\hat{f}(k)=\int_{\mathbb{R}}e^{-ikx}f(x)dx,\qquad f(x)=\frac{1}{2\pi}\int_{\mathbb{R}}e^{ikx}\hat{f}(k)dk. \tag{4.8}$$

Then (4.6) shows that the Fourier transform $\hat{G}_1$ of the 1D heat kernel is

$$\hat{G}_1(k,\Delta t)=e^{-k^2\Delta t}. \tag{4.9}$$

Now, the Poison summation formula (see, for example, [28]) states that

$$\sum_{n=-\infty}^{\infty}f\left(x+\frac{2\pi n}{h}\right)=\frac{h}{2\pi}\sum_{m=-\infty}^{\infty}\hat{f}(mh)e^{imhx}. \tag{4.10}$$

Rearranging terms, we have

$$f(x)-\frac{h}{2\pi}\sum_{m=-\infty}^{\infty}\hat{f}(mh)e^{imhx}=-\sum_{n=-\infty,n\neq0}^{\infty}f\left(x+\frac{2\pi n}{h}\right). \tag{4.11}$$

Applying (4.11) on G_1 and using the fact that the heat kernel is positive, we obtain

$$\left| G_1(x, \Delta t) - \frac{h}{2\pi} \sum_{m=-\infty}^{\infty} e^{-m^2 h^2 \Delta t + imhx} \right| \leq \sum_{n=-\infty, n\neq 0}^{\infty} G_1\left(x + \frac{2\pi n}{h}, \Delta t\right). \quad (4.12)$$

We would like to bound the right hand side of (4.12) for all $|x| \leq R$. For this, we first require that h satisfy the condition $\frac{2\pi}{h} - R \geq R$, i.e., $h \leq \frac{\pi}{R}$. Note that with this condition, we also have $\frac{2\pi}{h} - R \geq \frac{\pi}{h}$. Assume for now that $0 \leq x \leq R$. Then

$$\left| x + \frac{2\pi n}{h} \right| = x + \frac{2\pi n}{h} > \frac{2\pi n}{h}, \quad \text{for } n > 1, \quad (4.13)$$

and

$$\left| x + \frac{2\pi n}{h} \right| = \frac{2\pi |n|}{h} - x > \frac{\pi(2|n| - 1)}{h}, \quad \text{for } n < -1. \quad (4.14)$$

And similar results hold for $x \in [-R, 0]$. To summarize, if $h \leq \frac{\pi}{R}$, then

$$\sum_{n=-\infty, n\neq 0}^{\infty} G_1\left(x + \frac{2\pi n}{h}, \Delta t\right) \leq \sum_{n=1}^{\infty} G_1\left(\frac{\pi n}{h}, \Delta t\right) \quad \text{for } |x| \leq R. \quad (4.15)$$

Thus, if $h = \frac{\pi}{2\sqrt{\Delta t |\ln \epsilon|}}$, then we have $e^{-\pi^2/(4h^2\Delta t)} = \epsilon$ and

$$\sum_{n=1}^{\infty} G_1\left(\frac{\pi n}{h}, \Delta t\right) < 2G_1\left(\frac{\pi}{h}, \Delta t\right) = \frac{2\epsilon}{\sqrt{4\pi \Delta t}}. \quad (4.16)$$

Finally, we would like to truncate the infinite Fourier series. For this, we choose M such that $\frac{h}{\pi} e^{-M^2 h^2 \Delta t} = \frac{\epsilon}{\sqrt{4\pi\Delta t}}$, i.e., $M = \frac{1}{h}\sqrt{\frac{\ln(\sqrt{\pi}\epsilon/2h\sqrt{\Delta t})}{\Delta t}}$. This leads to

$$\left| \frac{h}{2\pi} \left(\sum_{m=-M-1}^{-\infty} + \sum_{m=M}^{\infty} \right) e^{-m^2 h^2 \Delta t + imhx} \right| \leq \frac{h}{\pi} \sum_{m=M}^{\infty} e^{-m^2 h^2 \Delta t} \leq \frac{2\epsilon}{\sqrt{4\pi \Delta t}}. \quad (4.17)$$

Combining (4.12), (4.15), (4.16), and (4.17), we obtain (4.7). □

We would like to remark here that it is well known that the trapezoidal rule achieves spectral accuracy for integrals involving infinitely long contours with rapidly decaying smooth integrand (see, for example, [82] for a comprehensive review on this topic).

To get a concrete feeling about the number of Fourier nodes needed in practice, we list the values of M in Table 4.1 for various precisions ϵ and time steps Δt for the approximation of $G_1(x, \Delta t)$ for $|x| \leq \pi$.

Table 4.1: Number of Fourier nodes needed to approximate the 1D heat kernel $G_1(x, \Delta t)$ for $x \in [-\pi, \pi]$.

ϵ \ Δt	10^{-1}	10^{-2}	10^{-3}	10^{-4}
10^{-3}	8	21	55	136
10^{-6}	11	34	100	296
10^{-9}	14	43	130	396
10^{-12}	17	50	154	475

Since $G_d(\mathbf{x}, \Delta t) = \prod_{i=1}^{d} G_1(x_i, \Delta t)$, we simply use the tensor product to construct Fourier spectral approximation of the heat kernel in higher dimensions. That is,

$$G_d(\mathbf{x}, \mathbf{y}; \Delta t) \approx \frac{h^d}{(2\pi)^d} \prod_{i=1}^{d} \left(\sum_{m_i=-M}^{M-1} e^{-m_i^2 h^2 \Delta t + i h m_i \cdot (x_i - y_i)} \right), \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \tag{4.18}$$

Obviously $G_d(\mathbf{x}, \mathbf{y}; \Delta t)$ decays exponentially fast and is almost negligible when $\|\mathbf{x} - \mathbf{y}\| > 10\sqrt{\Delta t}$ in physical space. Thus, if R is very large or Δt is very small, we should take advantage of the fact that the heat kernel is sharply peaked at the origin and design more efficient algorithms (say, via fast Gauss transform [44, 45]) accordingly.

4.1.2 Solving the pure initial value problem

We substitute (4.5) into (4.4) and exchange the order of integration to obtain

$$u(\mathbf{x}, \Delta t) = \frac{1}{(2\pi)^d} \int \cdots \int_{\mathbb{R}^d} e^{-\|\mathbf{k}\|^2 \Delta t + i \mathbf{k} \cdot \mathbf{x}} f(\mathbf{k}) d\mathbf{k}, \tag{4.19}$$

where $f(\mathbf{k})$ is given by the formula

$$f(\mathbf{k}) = \int \cdots \int_D e^{-i\mathbf{k}\cdot\mathbf{y}} d\mathbf{y}. \tag{4.20}$$

We note that $f(\mathbf{k})$ in (4.20) is a C^∞ function since D is a bounded domain in $\mathbb{R}^d$. One may also show that $f(\mathbf{k})$ tends to 0 as $\|\mathbf{k}\| \to \infty$ (see, for example, [74]).

For two dimensional problems, we may convert the double integral in (4.20) into a line integral using Green's theorem:

$$f(\mathbf{k}) = \iint_D e^{-i\mathbf{k}\cdot\mathbf{y}} d\mathbf{y} = \begin{cases} \dfrac{i}{k_1} \displaystyle\int_{\partial D} e^{-i\mathbf{k}\cdot\mathbf{y}} dy_2, & k_1 \neq 0, \\ \displaystyle\int_{\partial D} y_1 e^{-ik_2 y_2} dy_2, & k_1 = 0. \end{cases} \tag{4.21}$$

Since the boundary curve ∂D is smooth, the above integrals can be discretized via the trapezoidal rule to achieve spectral accuracy. To be more precise, let $s \in [0, 2\pi]$ be the parameter for the boundary curve and N be the total number of discretization points (equispaced in the parameter space but non-equispaced in physical space) on ∂D. Then we have

$$f(\mathbf{k}) \approx \begin{cases} \dfrac{2\pi i}{N k_1} \displaystyle\sum_{j=1}^{N} e^{-i\mathbf{k}\cdot\mathbf{y}(s_j)} y_2'(s_j), & k_1 \neq 0, \\ \dfrac{2\pi}{N} \displaystyle\sum_{j=1}^{N} y_1(s_j) e^{-ik_2 y_2(s_j)} y_2'(s_j), & k_1 = 0. \end{cases} \tag{4.22}$$

We note that for a smooth curve one may compute the derivatives $\mathbf{y}'(s_j)$ and the unit normal vectors $\mathbf{n}_y$ from $\mathbf{y}(s_j)$ $(j = 1, \ldots, N)$ using FFT with spectral accuracy (see, for example, [84]). Furthermore, the area bounded by ∂D can be computed as follows:

$$A = \iint_D 1 d\mathbf{y} = \int_{\partial D} y_1 dy_2 \approx \frac{2\pi}{N} \sum_{j=1}^{N} y_1(s_j) y_2'(s_j). \tag{4.23}$$

Similarly, for three dimensional problems, we may convert the volume integral in (4.20) into a surface integral using the divergence theorem:

$$f(\mathbf{k}) = \iiint_D e^{-i\mathbf{k}\cdot\mathbf{y}} d\mathbf{y} = \begin{cases} \dfrac{i}{k_1} \displaystyle\iint_{\partial D} e^{-i\mathbf{k}\cdot\mathbf{y}} (\hat{i}\cdot\mathbf{n}_y) ds_y, & k_1 \neq 0, \\ \displaystyle\iint_{\partial D} y_1 e^{-ik_2 y_2 - ik_3 y_3} (\hat{i}\cdot\mathbf{n}_y) ds_y, & k_1 = 0, \end{cases} \tag{4.24}$$

where $\hat{i}$ is the unit vector along x-axis and $\mathbf{n}_y$ is the unit normal vector at $\mathbf{y}$. Since we assume that ∂D is homomorphic to the unit sphere, we may parametrize via $[u, v] \in [0, \pi] \times [0, 2\pi]$ with u the polar angle and v the azimuthal angle. We discretize the integration in v via the trapezoidal rule and the integration in u via the Gauss-Chebyshev quadrature. Hence, the integrals in (4.24) are approximated by:

$$f(\mathbf{k}) \approx \begin{cases} \dfrac{i}{k_1} \displaystyle\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} e^{-i\mathbf{k}\cdot\mathbf{y}(u_i,v_j)} n_1(u_i, v_j) J(u_i, v_j) w_{ij}, & k_1 \neq 0, \\ \displaystyle\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} y_1(u_i, v_j) e^{-ik_2 y_2(u_i,v_j) - ik_3 y_3(u_i,v_j)} n_1(u_i, v_j) J(u_i, v_j) w_{ij}, & k_1 = 0, \end{cases} \tag{4.25}$$

where $n_1(u_i, v_j)$ is the x-component of the unit outward normal vector, $J(u_i, v_j) = |\mathbf{y}_u \times \mathbf{y}_v|$ is the Jacobian at the point (u_i, v_j), and w_{ij} is the quadrature weight at that point. Note again that we only need the coordinates $\mathbf{y}(u_i, v_j)$ and all other quantities can be computed from them efficiently since both $\mathbf{y}_u$ and $\mathbf{y}_v$ can be computed using FFT [84]. Furthermore, the volume bounded by ∂D can be computed as follows:

$$\begin{aligned} V &= \iiint_D d\mathbf{y} = \iint_{\partial D} x\hat{i} \cdot \mathbf{n}_y ds_y \\ &\approx \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} x(u_i, v_j) n_1(u_i, v_j) J(u_i, v_j) w_{ij}. \end{aligned} \tag{4.26}$$

After $f(\mathbf{k})$ has been computed, the solution $u(\mathbf{x}, \Delta t)$ in (4.19) can be evaluated by approximating the Fourier integral via the truncated trapezoidal rule as in Theorem 4.1.1. We have

$$u(\mathbf{x}, \Delta t) \approx \frac{h^d}{(2\pi)^d} \sum_{m_1=-M}^{M-1} \cdots \sum_{m_d=-M}^{M-1} e^{-\|\mathbf{m}\|^2 h^2 \Delta t + ih\mathbf{m}\cdot\mathbf{x}} f(h\mathbf{m}). \tag{4.27}$$

We now recall the definitions of NUFFTs. We use the convention in [42] to define type-1 and type-2 NUFFTs. The type-1 NUFFT evaluates sums of the form

$$f(\mathbf{k}) = \frac{1}{N} \sum_{j=1}^{N} c_j e^{\pm i\mathbf{k}\cdot\mathbf{x}_j}, \tag{4.28}$$

for "targets" $\mathbf{k}$ on a regular (equispaced) grid in $\mathbb{R}^d$ ($d = 1, 2, 3$ in our case), given function values c_j prescribed at arbitrary locations $\mathbf{x}_j$ in physical space. Here, N denotes the total number of source points.

The type-2 NUFFT evaluates sums of the form

$$F(\mathbf{x}_n) = \sum_{m_1=-M_1}^{M_1-1} \cdots \sum_{m_d=-M_d}^{M_d-1} f(h\mathbf{m}) e^{\pm ih\mathbf{m}\cdot\mathbf{x}_n}, \tag{4.29}$$

where the "targets" $\mathbf{x}_n$ are points irregularly located in $\mathbb{R}^d$, given the function values $f(h\mathbf{m})$ on a regular grid in the Fourier space.

It is now clear that we may apply type-1 NUFFT to evaluate $f(h\mathbf{m})$ defined in (4.22) or (4.25), then apply type-2 NUFFT to evaluate $u(\mathbf{x}, \Delta t)$ in (4.27).

We summarize the algorithm in Algorithm 1. The total cost of Algorithm 1 is $O\left(N_S + N_T + N_F \log N_F\right)$, where N_S is the total number of non-equispaced source points on the boundary, N_T is the total number of the target points near the boundary, and $N_F = (2M)^d$ is the total number of equispaced points in the Fourier space.

4.1.3 Extension to non-smooth boundary

When the boundary is piecewise smooth, for example, with corners, we only need to modify the discretization scheme for the boundary integrals in (4.21). The modification is straightforward, instead of using global trapezoidal rule, we divide the boundary into several chunks with corners belonging to the set of the end points of chunks (one can certainly divide the curve into as many as chunks as one would like to have). We then discretize the parameter space for each chunk using a high order (say, pth order) Gauss-Legendre rule. The discretization of the integrals in (4.21) will be of the pth order and the overall scheme is pth order.

Algorithm 1 NUFFT Based Solver for the Initial Value Problem (4.1)-(4.2)

Given the prescribed accuracy ϵ, the time step size Δt, the boundary ∂D, compute $u(\mathbf{x}, \Delta t)$ defined in (4.4) on a set of prescribed target points $\mathbf{x}_i$, $i = 1, \cdots, N_T$.

1: Discretize the boundary ∂D via N_S points in the parameter space (equispaced nodes for curves in 2D; scaled and shifted Chebyshev nodes along the polar direction and equispaced nodes along the azimuthal direction for surfaces in 3D), compute the source locations $\mathbf{y}_j$ ($j = 1, \ldots, N_s$) via the given parametrization of ∂D and the associated weights w_j.

2: Compute the derivatives $\mathbf{y}_j'$ and the Jacobians via FFT.

3: Compute the mesh spacing h and M for the Fourier spectral approximation of the heat kernel as in Theorem 4.1.1.

4: Use type-1 NUFFTs to evaluate $f(h\mathbf{m})$ defined in (4.22) or (4.25) for $m_i = -M, \cdots, M-1$ ($i = 1, \ldots, d$).

5: Use type-2 NUFFT to evaluate $u(\mathbf{x}_j, \Delta t)$ defined in (4.27) for $j = 1, \cdots, N_T$.

4.2 Thresholding

We now discuss how the thresholding is done in our algorithm. For the grid or mesh based methods, the accuracy of the thresholding step is generally of the first order. However, due to the smoothing effect of the diffusion process, the interface generally becomes smoother and thus it is possible to improve the accuracy of thresholding greatly. As pointed out in the preceding section, we use a carefully chosen set of points to represent the interface in both two and three dimensions and all other geometric quantities such as the tangential derivatives, unit normal vectors, and area elements can be computed efficiently from them. Hence, we only need to keep track of this set of points in the threshold dynamics.

We first discuss thresholding by finding the level set of a given solution value v to the initial value problem (4.1)-(4.2). The algorithm is as follows. We start from a set of source points $\mathbf{y}_j$ $(j = 1, \ldots, N_s)$. We then apply FFT to compute the unit normal vector on each source point $\mathbf{y}_j$ and allocate p target points along each normal direction on both sides of the source point. After applying Algorithm 1 to compute the solution to the initial value problem (4.1)-(4.2), we use a root finding algorithm (Müller's method in [69] to be more specific) to find a point whose solution value is equal to v along each normal direction. We summarize the whole scheme in Algorithm 2.

Remark 4.2.1. *We have several remarks on the details of the algorithm. First, the root finding procedure converges rapidly. It only takes* 5–6 *iterations to converge to* 12*-digit accuracy. Second, the total number of points in physical space is* pN_s*, where* p *is some number around* 10*. We set* p *to* 16 *in our implementation. Third, the original MBO method [64], i.e., the motion by mean curvature, corresponds to the case where* $v = \frac{1}{2}$*.*

When Algorithm 2 is applied to find the level set whose corresponding function value causes the interface to shrink, as in the original MBO method, numerical

Algorithm 2 Computing the level set at a given function value v

Given the prescribed accuracy ϵ, the time step size Δt, a set of source points $\mathbf{y}_j$ $(j = 1, \ldots, N_S)$ describing the interface at current time, and a specified function value v, compute the level set for the function value v, return a new set of N_S points representing the new interface after diffusion and thresholding, at which the solution value of the initial value problem is equal to v. Also return the area or the volume bounded by the new level set.

1: Use FFT to compute the derivatives $\mathbf{y}_j'$ and the unit normal vector $\mathbf{n}_j$ for $j = 1, \ldots, N_S$.

2: Compute a length L that is proportional to $\sqrt{\Delta t}$, say, $L = 5\sqrt{\Delta t}$.

3: Compute p Gauss-Legendre nodes on the standard interval $[-1, 1]$.

4: For each point $\mathbf{y}_j$ on the boundary, allocate along the normal direction p scaled Gauss-Legendre nodes on the interval $[-L, L]$ centred at $\mathbf{y}_j$; altogether we obtain pN_S target points $\mathbf{x}_i$ for $i = 1, \cdots, pN_S$.

5: Apply Algorithm 1 to compute the solution u_c to the intial value problem (4.1)-(4.2) at these target points.

6: For each point $\mathbf{y}_j$ on the boundary along its normal direction, approximate $u(\mathbf{x}(s), \Delta t)$ by a $p-1$th order Legendre polynomial, and use Müller's method [69] to find the parameter value $s \in [-L, L]$ at which the function value is equal to the given value v.

7: Calculate the coordinates of the points in the level set by setting $\mathbf{y}_j = \mathbf{y}_j + s \cdot \mathbf{n}_j$.

8: Use (4.23) or (4.26) to compute the area or the volume bounded by the new level set.

instability may occur for long time simulations due to the cross-over of the normal vectors. However, one can easily eliminate the instability by filtering out the high frequency content in the point set. For curves in two dimensions, this is done as follows: (1) compute the Fourier transform $\hat{f}_i$ $(i = 1, \ldots, M)$ of the coordinates; (2) multiply $\hat{f}_i$ by a smooth filter, say, $h(k) = \frac{1}{2}\text{erfc}\left(\frac{12(|k|-(M/2+M)/2)}{M-M/2}\right)$, to gradually filter out the high frequency part $\hat{f}_i$ for $i \in [M/2, M]$ to zero, where erfc is the complementary error function defined by the formula $\text{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}dt$; (3) apply inverse Fourier transform to obtain a smoother point set. For surfaces in three dimensions, the procedure is almost identical except that the (inverse) Fourier transform is replaced by the forward (backward) spherical harmonic transform.

In many applications, one needs to find the area or volume preserving level set. We may simply add another layer of iterations for this. The algorithm is summarized in Algorithm 3. Our numerical experiments show that about 4–6 iterations are required to achieve 12-digit accuracy and only 2–3 iterations are needed to reach single precision. Obviously, the cost of root finding procedure in Algorithms 2 and 3 is $O(N_s)$.

4.3 Numerical results

We have implemented the aforementioned algorithms in Fortran and MATLAB. We use the NUFFT library from [57]. We now illustrate the performance of our algorithm via several numerical examples. The timing results were obtained on a laptop with a 2.7GHz Intel(R) Core(TM) i5 processor and 8GB of RAM.

Algorithm 3 Calculating the area or volume preserving level set

Given the prescribed accuracy ϵ, the time step size Δt, a set of source points $\mathbf{y}_j$ $(j = 1, \ldots, N_S)$ describing the interface at current time, return a new set of N_S points representing the new interface after diffusion and thresholding, whose area or volume is equal to that bounded by the original set of points.

1: Use (4.23) or (4.26) to compute the area or the volume bounded by the original set of source points.

2: Set three initial guesses for the solution values v_0, v_1, v_2 somewhat randomly but close to, say, 0.5, and apply Algorithm 2 to find the level sets and the areas or the volumes associated with v_i $(i = 0, 1, 2)$.

3: Use Müller's method to find the solution value v and associated level set whose enclosed area or volume is equal to that bounded by the original set of points.

4: Return the new set of points and the solution value v.

4.3.1 Example 1: Accuracy of the NUFFT based heat solver

We first check the accuracy of the NUFFT based heat solver in Algorithm 1 in both two and three dimensions. For the 2D solver, we use the hexagram in Example 4 as the boundary and compute the solution to the initial value problem (4.1)-(4.2) at $\Delta t = 0.002$. The reference solution is obtained with $N_S = 1024$ points on the boundary. The numerical solutions are evaluated at $N_T = p \cdot N_S = 16 \times 1024$ fixed target points with $p = 16$ points along each normal direction of the source points. Table 4.2 lists the relative L^2 error of the numerical solution with various number of source points on the boundary.

Table 4.2: Relative L^2 error versus number of discretization points on the boundary for the 2D heat solver. The boundary curve is a smooth hexagram in Example 4.

N_S	30	60	90	120	150
E	5.738e-2	2.077e-4	2.373e-7	1.211e-10	3.5438e-14

For the 3D solver, we use a sphere of radius 0.5 as the boundary and compute the solution of the heat equation at $\Delta t = 0.001$. The reference solution is obtained with $N_S = 256 \times 256$ points on the boundary. We evaluate the numerical solution at $N_T = p \cdot N_S = 16 \times 256 \times 256$ fixed target points with $p = 16$ points along each normal direction. Table 4.3 lists the relative L^2 error of the numerical solution with various number of source points on the boundary.

From Tables 4.2 and 4.3, we observe that our heat solver is spectrally accurate in space for both two and three dimensional problems.

Table 4.3: Relative L^2 error versus number of discretization points on the boundary for the 3D heat solver. The boundary is a sphere of radius 0.5.

N_S	40×40	60×60	80×80	100×100	120×120
E	5.104e-4	2.519e-7	6.758e-10	2.929e-12	6.229e-15

4.3.2 Example 2: Efficiency and accuracy of the threshold dynamics method

In this example, we consider the motion by mean curvature of a circle with the radius 0.5 centred at the origin. As discussed in Remark 4.2.1, the motion by mean curvature is equivalent to finding the level set with the solution value equal to 0.5. We march to the final time $T = 0.01$ and compare the area loss with the exact answer $0.01 \times 2\pi$ [70]. Table 4.4 lists the relative error of the area loss and the total computational time of Algorithm 2 and the method in [95], which is an FFT based fast solver on a uniform mesh. Since Algorithm 2 only requires the discretization near the boundary and the spatial mesh size can be chosen more or less independently of the time step size, one needs much less number of discretization points in space to achieve same accuracy. And we observe from the table that Algorithm 2 is significantly more efficient than the uniform mesh method, with the computational time for the uniform mesh method is about 20 times more for the case $\delta t = 0.002$, and 60 times more for the case $\delta t = 0.001$.

We now examine the order of accuracy in time. From [62, 64, 77], we see that the MBO method is first order in time for smooth two-dimensional problems. To test the order of accuracy in time of our algorithm, we consider the same problem and march to $T = 0.02$ using Algorithm 2, compute the area loss, and compare it with the exact answer - 0.04π. Table 4.5 lists the relative error and convergence order of our algorithm, which shows that our algorithm has the predicted first order accuracy in time. Here we have used 400 points to discretize the boundary.

Table 4.4: Accuracy and timing results of Algorithm 2 and the uniform mesh method

(a) Algorithm 2

Δt	#(Source Points)	Error	Time	#(Fourier Modes)	TOL. of NUFFT
0.002	400	0.28%	0.49s	78×78	1d-12
0.001	400	0.14%	0.97s	112×112	1d-12

(b) The Uniform Mesh Method

Δt	Δx	Error	Time
0.002	$\frac{1}{2048}$	0.32%	10.63s
0.001	$\frac{1}{4096}$	0.15%	62.65s

Table 4.5: Relative error and convergence order in time of Algorithm 2 for the motion by mean curvature of a circle.

Δt	Relative Error	Convergence Order
0.004	0.005941	–
0.002	0.002938	1.02
0.001	0.001461	1.01
0.0005	0.000728	1.01

Finally, we consider the motion by mean curvature of a sphere of radius 0.5. We note that the sphere shrinks without changing its shape since it has constant curvature and the dynamic of the radius of the sphere is governed by [47]:

$$\frac{dr}{dt} = -\frac{2}{r}, \qquad r(0) = r_0 \tag{4.30}$$

where $r(t)$ is the radius of the sphere at time t. The exact solution to the above

ODE is

$$r(t) = \sqrt{r_0^2 - 4t}. \tag{4.31}$$

We apply Algorithm 2 to march to $T = 0.02$, calculate the volume loss and compare it against the exact value 0.229995 obtained from (4.31). We use 96×96 points to discretize the sphere. Table 4.6 lists the relative L^2 error, the estimated convergence order in time, and the total computational time in seconds for various time step size Δt. We observe that our algorithm has first order accuracy in time and is fairly efficient for three dimensional problems.

Table 4.6: Numerical Results for the Motion by Mean Curvature of a Sphere.

Δt	Error	Conv. Order	Time	#(Fourier Modes)	TOL. of NUFFT
0.004	1.365e-03	–	25.27s	72× 72×72	1d-10
0.002	6.741e-04	1.02	56.96s	90× 90×90	1d-10
0.001	3.350e-04	1.01	149.73s	120× 120×120	1d-10
0.0005	1.670e-04	1.01	472.57s	160× 160×160	1d-10

Remark 4.3.1. *While it is spectrally accurate in space, our algorithm is only first order accurate in time, just as the original MBO method. However, many researchers have proposed high-order schemes in time as well where many of these schemes use the extrapolation to achieve high order in time. We would like to refer the reader to [62, 75, 79] for these high-order temporal schemes. Our algorithm may be extended to high order in time in a similar fashion.*

4.3.3 Example 3: Interface motion by mean curvature in 2D and 3D

We now show that our algorithm is capable of handling various smooth geometries in both 2D and 3D. We first compute the motion by mean curvature of a crescent in 2D using a step size $\Delta t = 0.0005$. The dynamics at a sequence of times is shown in Fig.4.1. From these pictures, we observe that no topological changes arise, and that the boundary eventually becomes a circle as it shrinks.

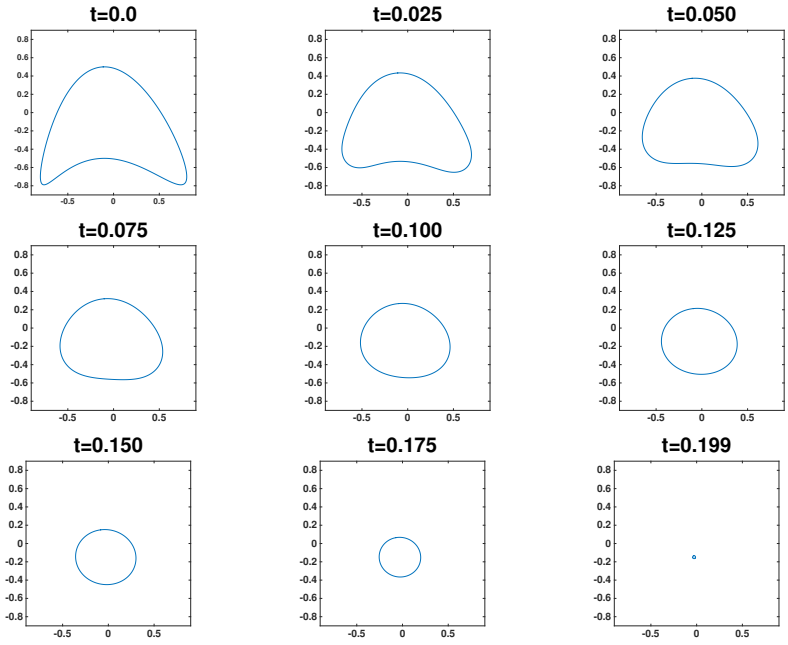


Figure 4.1: The Motion by Mean Curvature of a Crescent at Various Times. We set $\Delta t = 0.0005$ and use 400 points to discrete the interface.

We next compute dynamics of smooth surface in 3D. Fig. 4.2 shows the motion by mean curvature of a bowl shaped initial region. We set $\Delta t = 0.001$ and use 96×192 points to discretize the surface. The bowl evolves into a sphere and shrinks as t increases.

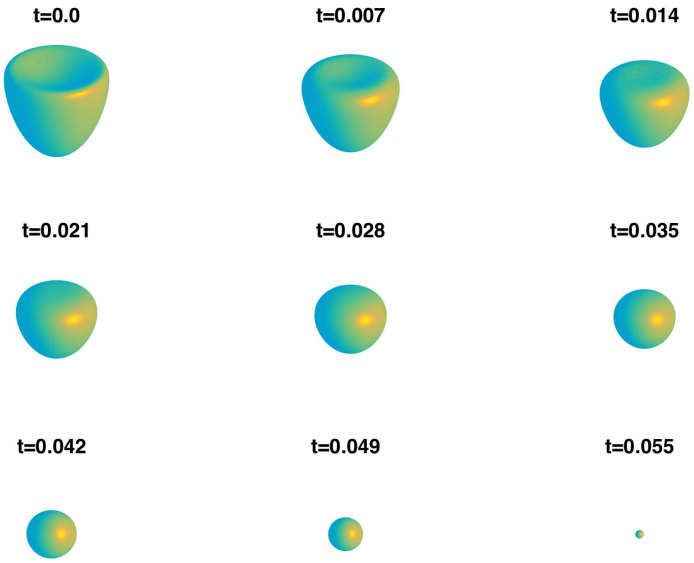


Figure 4.2: Snapshots of the Motion by Mean Curvature of a Bowl. $\Delta t = 0.001$.

4.3.4 Example 4: Area preserving motion of a hexagram in 2D

We consider the motion of a hexagram under mean curvature with the area preserving constraint. We set the time step $\Delta t = 0.001$ and discretize the boundary with 400 points. The number of points along each normal direction is set to $p = 16$. Fig. 4.3 shows subsets of the source and target points for the initial geometry. Fig. 4.4 shows the motion of the interface at various times. From

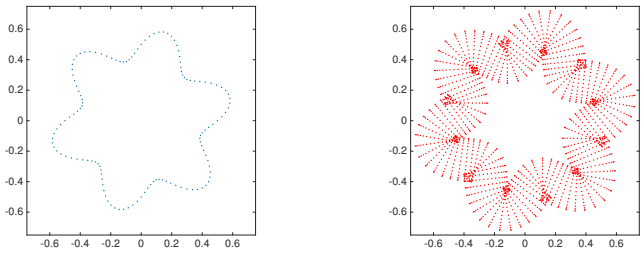


Figure 4.3: Left: subsets of the source points at $t = 0$. Right: subsets of the target points at $t = 0$.

these pictures, we observe that the shape eventually becomes a circle, which is in agreement with the exact solution at equilibrium. Furthermore, the initial area has numerical value $A = 0.801106\ldots$. From this, we may obtain the radius of the equilibrium circle $r = \sqrt{A/\pi} = 0.504975\ldots$. We have computed the relative L^∞ error of the shape at $t = 0.04$ by comparing it with the equilibrium circle and found it to be 0.000459. This shows that our algorithm is quite accurate to simulate the motion by mean curvature under the area preserving constraint.

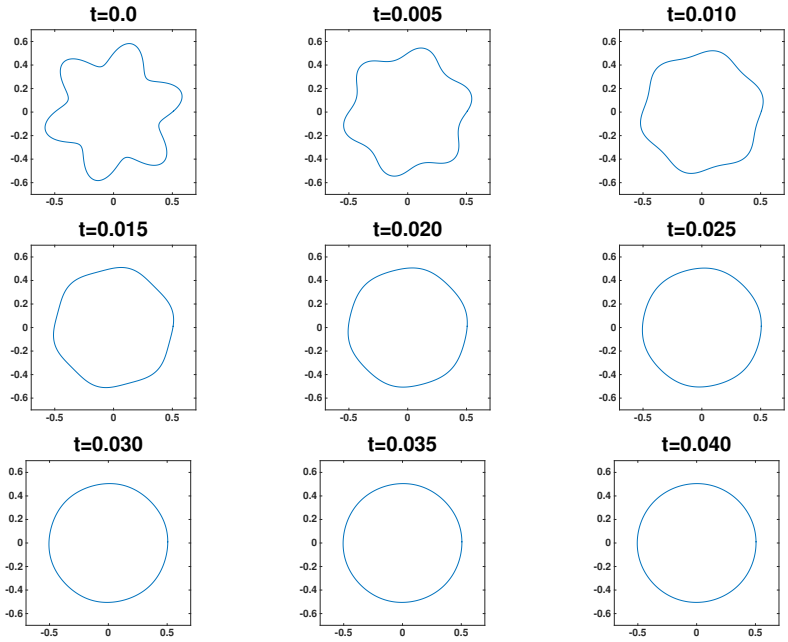


Figure 4.4: Snapshots of the Motion by Mean Curvature of a Hexagram under Area Preserving Constraint. $\Delta t = 0.001$.

4.3.5 Example 5: Volume preserving motion of a wedge in 3D

To show the capability and efficiency of our method to simulate the dynamics of the complicated surface in 3D, we compute the volume preserving motion of a wedge in three dimensions. Fig. 4.5 shows the initial shape of the wedge from various angles. We set $\Delta t = 0.001$ and use 96×96 points to discretize the surface.

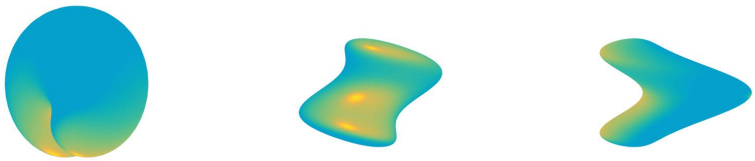


Figure 4.5: The front, side, and vertical views of the initial geometric profile in Example 5.

Fig.4.6 displays the snapshots at different times of the volume preserving motion of the wedge at various times. Again, the wedge eventually becomes a sphere when it reaches the equilibrium. The simulation takes only about 2068 seconds. We remark that for such 3D simulations, the uniform mesh method simulation with similar accuracy would require at least $1024 \times 1024 \times 1024$ grid points and takes roughly 200 hours to reach equilibrium.

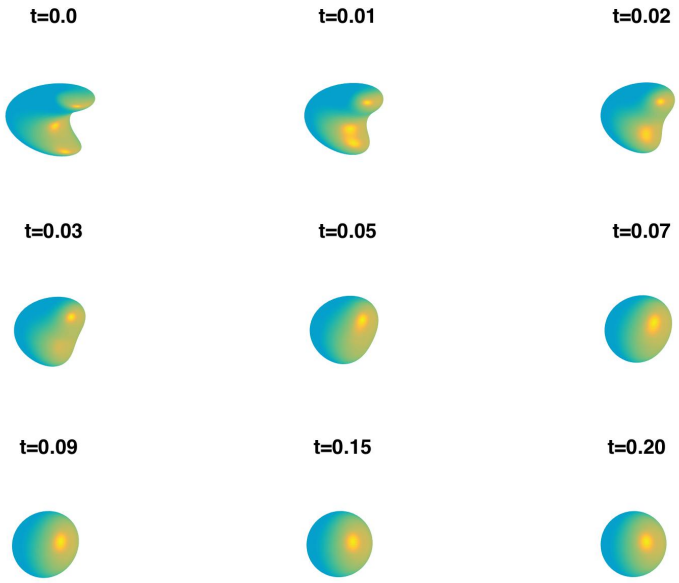


Figure 4.6: Snapshots of the Volume Preserving Motion of a Wedge. $\Delta t = 0.001$.

4.3.6 Example 6: Mean curvature motion of hexagram in 2D

To show the capability of our method to simulate the dynamics of non-smooth interface in 2D, we compute the mean curvature motion of a non-smooth Hexagram in two dimensions.

We first check the accuracy of the NUFFT based heat solver in Algorithm 1 in two dimensions for non-smooth case. For the 2D solver, we use the hexagram in Fig.4.7 as the boundary and compute the solution to the initial value problem (4.1)-(4.2) at $\Delta t = 0.0005$. The reference solution is obtained with $N_S = 120 \times 12$ points on the boundary with 12 chunks. The numerical solutions are evaluated at $N_T = p \cdot N_S = 16 \times 1440$ fixed target points with $p = 16$ points along each proper direction (See Remark 4.1) of the source points. Table 4.7 lists the relative L^2 error of the numerical solution with various number of source points on the boundary.

Table 4.7: Relative L^2 error versus number of discretization points on the boundary for the 2D heat solver. The boundary curve is a non-smooth hexagram.

N_S	30×12	40×12	50×12	60×12	70×12
E	3.396e-6	5.917e-8	5.847e-10	3.373e-12	1.629e-14

We set $\Delta t = 0.0005$ and use 360 points to discretize the interface with 12 chunks. Fig.4.7 displays the snapshots at different times of the mean curvature motion of hexagram at various times. The numerical example shows that our algorithm is also very robust for the non-smooth case as it shrinks.

Remark 4.3.2. *We note that the non-smooth initial interface will immediately evolve to a smooth interface under the mean curvature motion or the mean curvature motion with volume preserving. Hence, we simply use first kind Chebyshev*

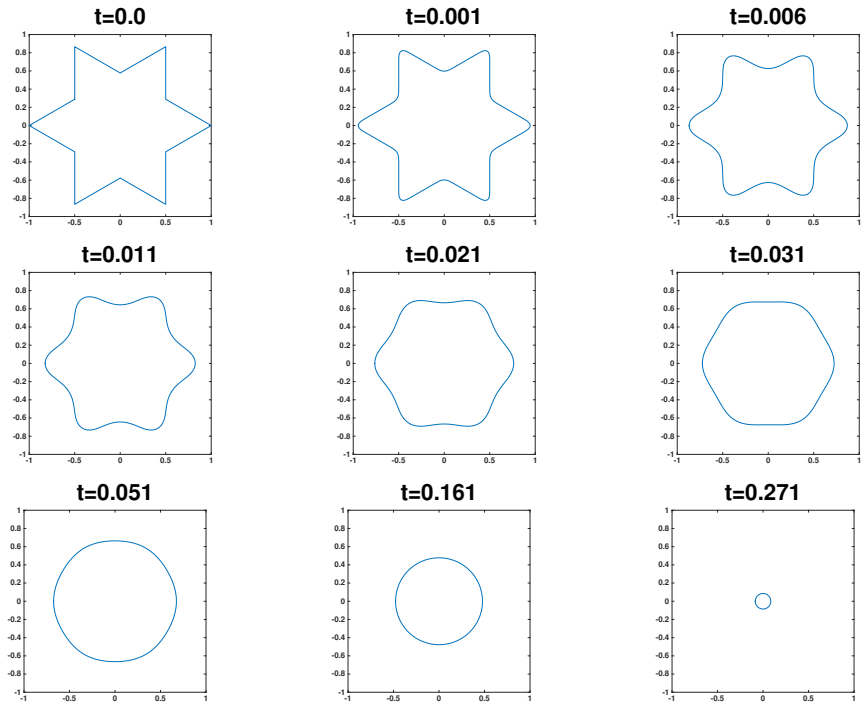


Figure 4.7: Snapshots of the Mean Curvature Motion of a Hexagram. $\Delta t = 0.0005$.

nodes(do not include end points) to discretize the parametric space in the first step. After then, we can easily map the parametric space(Chebyshev nodes) to uniform parametric space by inverse trigonometric transform. That is, we can treat the non-smooth case as a smooth case except when at the first step.

In the first step of the simulation, to avoid crossing of points, when generating the target points, we choose different direction(weighted sum of normal directions at neighbour points) at each point on the interface instead of normal directions (See Fig. 4.8). Note that different choices of target points does not affect the accuracy of our heat solver.

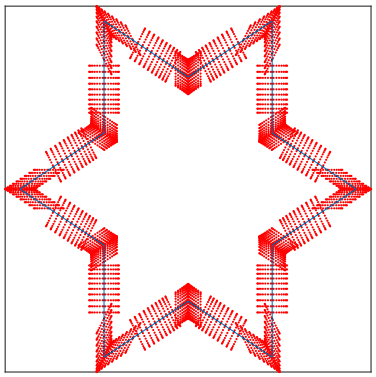


Figure 4.8: The Diagram of Target Points Generated Along a Non-smooth Hexagram Interface.

4.3.7 Example 7: Application in solid wetting problems in 2D

There is no essential difficulty to extend our algorithm into non-smooth motions with multiple junctions (e.g. motion of contact points in solid wetting problems). We only need to modify our algorithm by calculating the heat diffusion of each domain separately on the same target points and find the root of $\phi = v$. Here, v is a parameter under determination to keep volume preserving and

$$\phi = \sqrt{\Delta t}(1 - 2\widetilde{\chi_{D_1^k}} - (1 + \cos(\theta_Y))\widetilde{\chi_{D_3}}) \tag{4.32}$$

where θ_Y is Young's angle[96], $\widetilde{\chi_{D_1^k}}, \widetilde{\chi_{D_3}}$ is the heat diffusion after Δt with $\chi_{D_1^k}, \chi_{D_3}$ as the initial condition and χ_D represents the characteristic function of domain D.

Fig.4.9 displays the snapshots of the wetting process on a hydrophilic solid material from a half circle initial profile to a circular curve with contact angle as $\pi/3$. Fig.4.10 displays the snapshots of the wetting process on a hydrophobic solid material from a half circle initial profile to a circular curve with contact angle as $2\pi/3$.

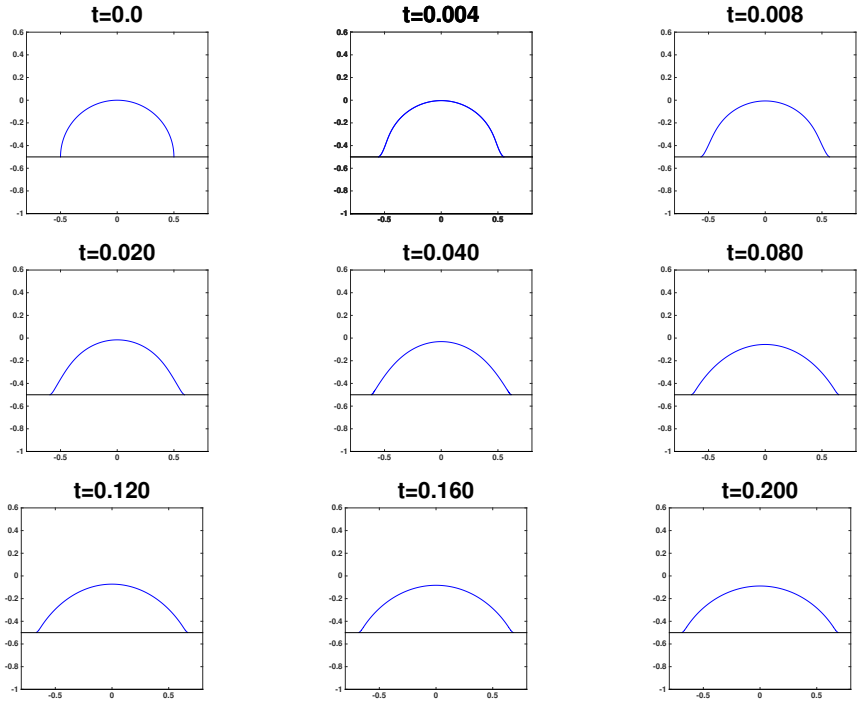


Figure 4.9: Snapshots of Wetting Process on Hydrophilic Solid Material with Young's Angle as $\pi/3$.

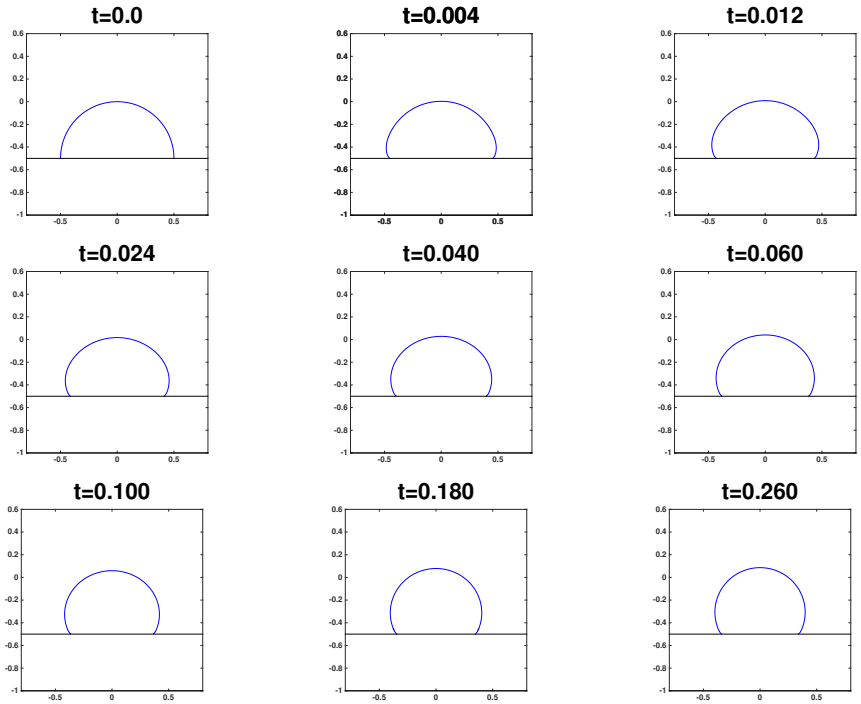


Figure 4.10: Snapshots of Wetting Process on Hydrophobic Solid Material with Young's Angle as $2\pi/3$.

Remark 4.3.3. *To keep the algorithm stable, the target points are generated along the direction parallel to the solid boundary for NT points (e.g. in our simulation, we set $NT = 40$) near the solid boundary.*

4.3.8 Example 8: Two droplets merging in 2D

To show the capability of the extension to cases of topological change, we simulate the process of two droplets merging together based on the model proposed in [95]. Initially, we have two droplets on the solid surface as in Fig. 4.11. We set $\theta_Y = \frac{\pi}{3}$ and two droplets will spread on the solid surface. Theoretically, two droplets will merge together when two contact points touch each other. We set $dt = 0.0002$ and use 200 points to descretize each interface. Fig. 4.12 displays the snapshots at different times of the volume preserving motion of two droplets on the solid surface. From these pictures, we see that two droplets merge together and the interface eventually becomes a part of circle.

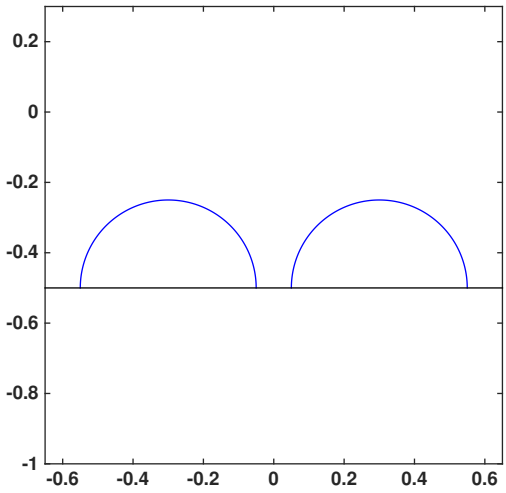


Figure 4.11: Initial Profile of Two Droplets on the Solid Surface.

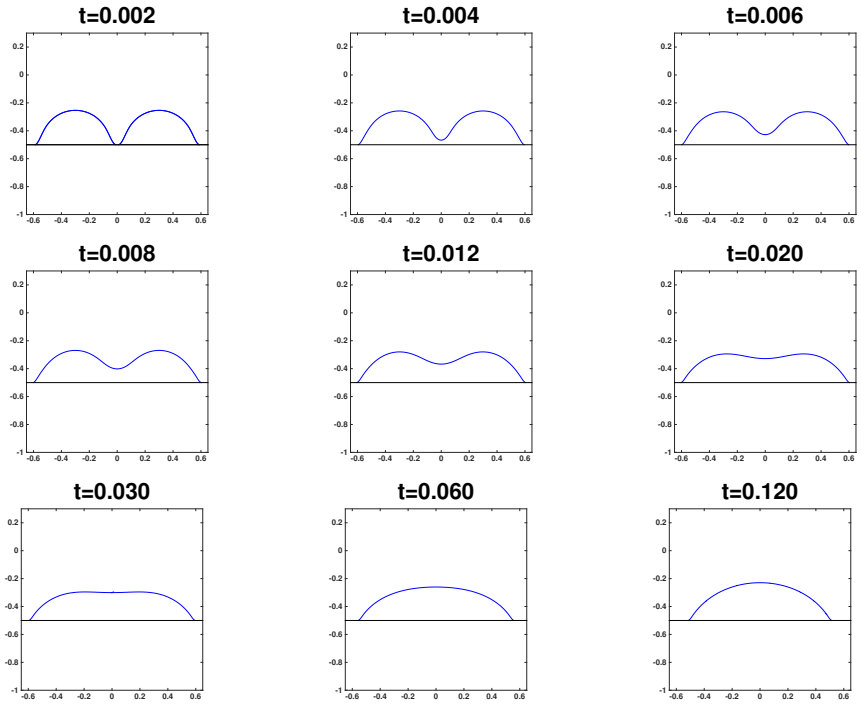


Figure 4.12: Snapshots of Two Droplets Merging Together on Solid Surface.

4.3.9 Example 9: Application in solid wetting problems in 3D

For a droplet on the solid surface in three dimensions, the NUFFT based heat solver is exactly same as the smooth one. In solid wetting problems, when the points is near the solid surface, we only need to modify in the direction of generating the target points perpendicular to the normal direction of solid surface for NT points near the solid surface. Fig. 4.13 displays the snapshots with 3D view and front view of the solid wetting process on hydrophobic solid surfaces. In this simulation, we set $dt = 0.002$ and use 96×96 points to descretize the surface of droplet with NT being chosen as 20×96. Fig. 4.14 displays the snapshots with 3D view and front view of the solid wetting process on hydrophilic solid surfaces. In this simulation, we set $dt = 0.0005$ and use 96×96 points to descretize the surface of droplet with NT being chosen as 10×96.

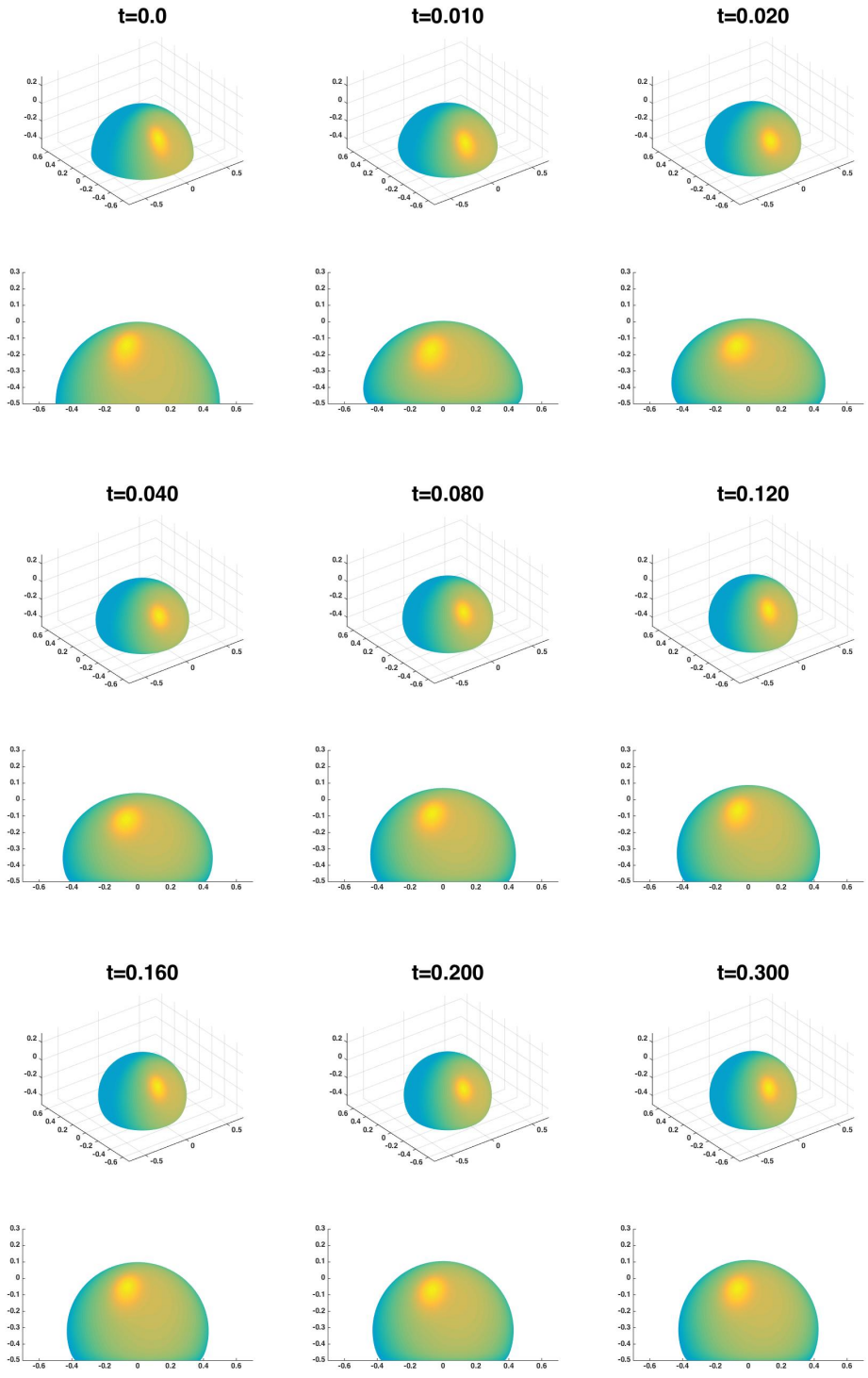


Figure 4.13: Snapshots of Wetting Process on Hydrophobic Solid Material (3D view and front view both) with Young's Angle as $2\pi/3$.

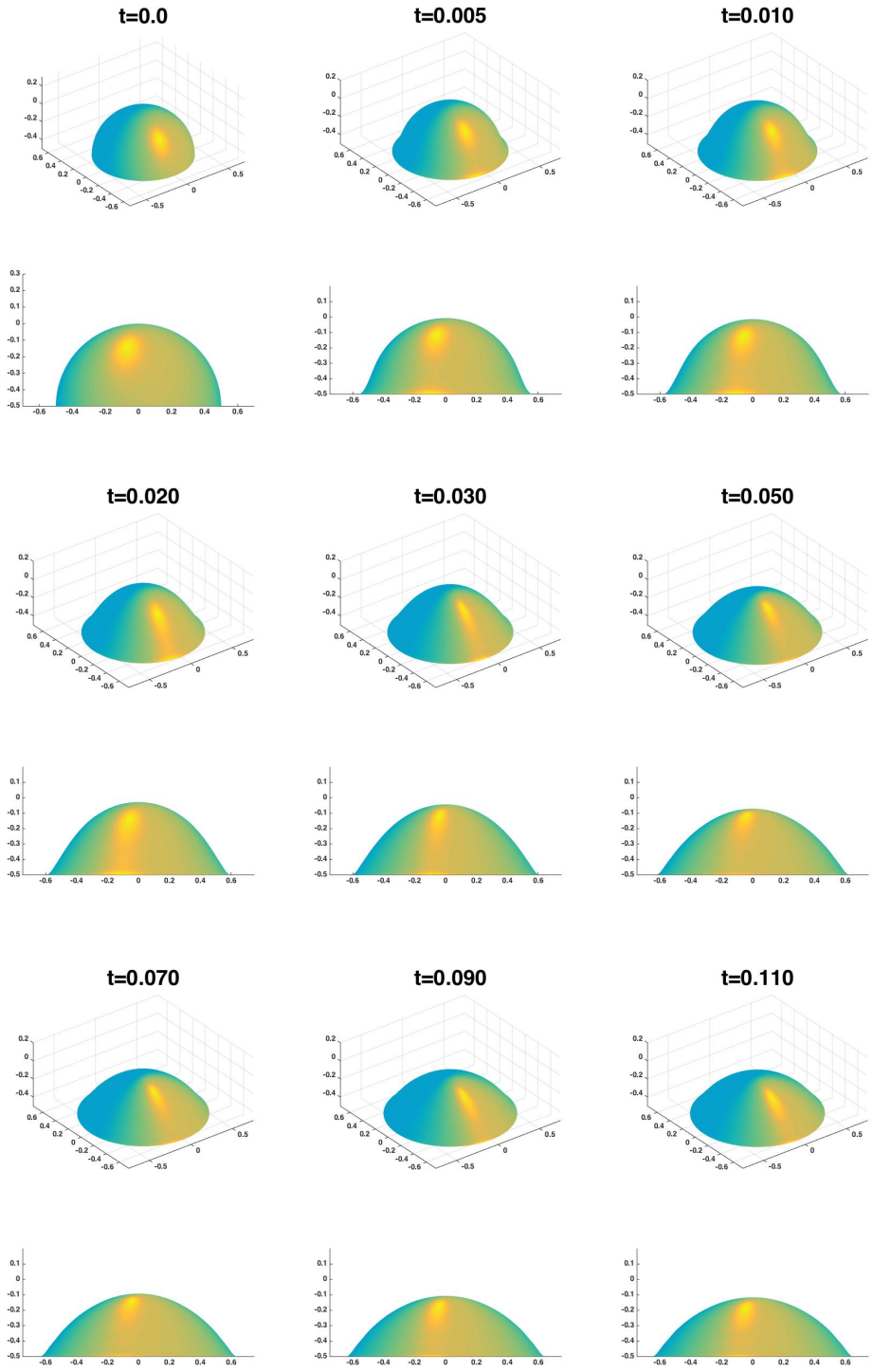


Figure 4.14: Snapshots of Wetting Process on Hydrophilic Solid Material (3D view and front view both) with Young's Angle as $\pi/3$.

Chapter 5

An efficient threshold dynamics method for image segmentation problems

Image segmentation is one of the fundamental tasks in image processing. In broad terms, it is the process of partitioning a digital image into many segments according to a characterization of the image. The motivation behind this is to auto-determine which part of an image is meaningful for analysis, which also makes it one of the fundamental problems in computer vision. Many practical applications require image segmentation, like content-based image retrieval, machine vision, medical imaging, object detection and traffic control systems [68].

Variational methods have enjoyed tremendous success in image segmentation. A typical variational method for image segmentation starts by choosing an energy functional over the space of all leagal segmentations, minimizing which gives a segmentation with desired properties. For instance, the celebrated Mumford-

Shah model [71] uses the following formulation of energy:

$$E_{MS}(u,\Gamma) = \lambda \int_D (u-f)^2 dx + \mu \int_{D\setminus\Gamma} |\nabla u|^2 dx + Length(\Gamma), \qquad (5.1)$$

where Γ is a closed subset of D given by the union of a finite number of curves representing the set of edges (i.e. boundaries of homogeneous regions) in the image f, u is a piecewise smooth approximation to f, and μ, λ are positive constants. Despite its descriptiveness, non-convexity of (5.1) makes the minimization problem difficult to analyse and solve numerically [7].

To address this issue, a useful simplification of (5.1) is to restrict the minimization to functions (i.e. segmentations) that take a finite number of values. The resulting model is commonly referred to as the piecewise constant Mumford-Shah model. In particular, there is the following two-phase Chan-Vese functional [17, 86]:

$$E_{CV}(\Sigma, C_1, C_2) = \lambda Per(\Sigma; D) + \int_\Sigma (C_1 - f)^2 dx + \int_{D\setminus\Sigma} (C_2 - f)^2 dx \qquad (5.2)$$

where Σ is the interior of a closed curve and $Per(.)$ denotes the perimeter. C_1 and C_2 are averages of f within Σ and $D \setminus \Sigma$ respectively:

$$C_1 = \frac{\int_\Sigma f dx}{\int_\Sigma dx} \quad \text{and} \quad C_2 = \frac{\int_{D\setminus\Sigma} f dx}{\int_{D\setminus\Sigma} dx}$$

The level set method was used here to solve the minimization problem: Let $\phi(x) : D \to R$ be a Lipschitz continuous function with $\Sigma = \{x \in D : \phi(x) > 0\}$ and $D \setminus \Sigma = \{x \in D : \phi(x) < 0\}$. We can rewrite (5.2) as

$$E_{CV}(\phi, C_1, C_2) = \int_D \{\lambda |\nabla H(\phi)| + H(\phi)(C_1 - f)^2 + (1 - H(\phi))(C_2 - f)^2\} dx \qquad (5.3)$$

where $H(\cdot) : R \to R$ is the Heaviside function

$$H(\xi) = \begin{cases} 0 & \text{if } \xi < 0, \\ 1 & \text{if } \xi \geq 0. \end{cases}$$

In practice, a regularized version of H denoted by H_ε is used. Then the Euler-Lagrange equation of (5.3) with respect to ϕ is given by

$$\frac{\partial \phi}{\partial t} = -H'_\varepsilon(\phi)\{-\{(C_1 - f)^2 - (C_2 - f)^2\} + \lambda \nabla \cdot (\frac{\nabla \phi}{|\nabla \phi|})\} \tag{5.4}$$

Equation (5.4) is nonlinear and requires regularization when $|\nabla \phi| = 0$.

Over the years, various modifications [7, 12, 85, 86] are used in order to solve the equations more efficiently. For example, in [8], the authors instead of solving the optimal problem directly. They solved a dual fomulation of the continuous Potts model based on its convex relaxation. In [14], a two-stage segmentation method is proposed. In the first stage, the authors apply the split Bregman method [41] to find the minimizer of a convex variant of the Mumford-Shah functional. In the second stage, a K-means clustering algorithm is used to choose $k-1$ thresholds automatically to segment the image into k segments. One of the advantages of this method is that there is no need to specify the number of segments before finding the minimizer. Any k-phase segmentation can be obtained by choosing $k-1$ thresholds after the minimizer is found. In [25], a frame-based model was introduced in which the perimeter term was approximated by a term involving framelets. The framelets were used to capture key features of biological structures. The model can also be fast implemented using split Bregman method [41].

Chan et al. [16] considered a convex reformulation to part of the Chan-Vese model. Given fixed values of C_1 and C_2, a global minimizer can be found. It is then demonstrated in [97] that this convex variant can be regarded as a continuous min-cut (primal) problem, and a corresponding continuous max-flow problem can be formulated as its dual. Efficient algorithms are developed by taking advantage of the strong duality between the primal and the dual problem, using the augmented Lagrangian method or the primal-dual method (see [90, 97] and references therein).

Esedoglu et al. [32] proposed a phase-field approximation of (5.2) in which the Ginzburg-Landau functional is used to approximate the perimeter:

$$E^{\varepsilon}_{MS}(u, C_1, C_2)$$
$$= \int_D \left\{ \lambda \left(\varepsilon |\nabla u|^2 + \frac{1}{\varepsilon} W(u) \right) + u^2 (C_1 - f)^2 + (1-u)^2 (C_2 - f)^2 \right\} dx \qquad (5.5)$$

where $\varepsilon > 0$ is the approximate interface thickness and $W(\cdot)$ is a double-well potential. Variation of (5.5) with respect to u yields the following gradient descent equation:

$$u_t = \lambda \left(2\epsilon \Delta u - \frac{1}{\epsilon} W'(u) \right) - 2\{u(C_1 - f)^2 + (u-1)(C_2 - f)^2\}$$

which can be solved efficiently by an MBO based threshold dynamic method that works by alternating the solution of a linear (but non-constant coefficient) diffusion equation with thresholding.

The idea of approximating the perimeter of a set by a non-local energy (using heat kernel) [2][67] is used by Esedoglu and Otto [33] to design an efficient threshold dynamics method for multi-phase problems with arbitrary surface tensions. The method is also generalized to wetting on rough surfaces in [95]. In this chapter, we propose an efficient iterative thresholding method for minimizing the piecewise constant Mumford-Shah functional based on the similar approach. The perimeter term in (5.2) is approximated by a non-local multi-phase energy constructed based on convolution of the heat kernel with the characteristic functions of regions. An iterative algorithm is then derived to minimize the approximate energy. The procedure works by alternating the convolution step with the thresholding step. The convolution can be implemented efficiently on a uniform mesh using the fast Fourier transform (FFT) with the optimal complexity of $O(N \log N)$ per iteration. We also show that the algorithm has the total energy decaying property.

The rest of the chapter proceeds as follows. In Section 5.1, we first give the approximate piecewise constant Mumford-Shah functional. We then derive the

iterative thresholding scheme based on the linearization of the approximate functional. The monotone decrease of energy at each iteration is proved (with details given in the appendix). In Section 5.2, we present some numerical examples to show the efficiency of the method.

5.1 An efficient threshold dynamics method for image segmentation

In this section, we introduce an efficient threshold dynamics method for multi-phase image segmentation based on the Chan-Vese model [86]. The perimeter terms will be approximated by a non-local multi-phase energy constructed based on convolution of the heat kernel with the characteristic functions of regions. The iterative algorithm is then derived as an optimization procedure for the approximate energy.

5.1.1 The approximate Chan-Vese functional

Let Ω denote the domain of an input image f given by a d-dimensional vector. Our task is to find an n-phase partition $\{\Omega_i\}_{i=1}^n$ of Ω which minimizes

$$\mathcal{E} = \sum_{i=1}^{n} \left[\int_{\Omega_i} g_i d\Omega_i + \lambda |\partial \Omega_i| \right] \tag{5.6}$$

where Ω_i represents the region of the i^{th} phase; $|\partial\Omega_i|$ is the length of a boundary curve of the region Ω_i; $g_i = ||C_i - f||_2^2$ ($||.||_2$ denotes the l^2 vector norm) and

$$C_i = \frac{\int_\Omega u_i f d\Omega}{\int_\Omega u_i d\Omega}. \tag{5.7}$$

Let $u = (u_1(x), \cdots, u_n(x))$ where $\{u_i(x)\}_{i=1}^n$ are the characteristic functions of the regions $\{\Omega_i\}_{i=1}^n$. We then look for u such that

$$u = \underset{u \in \mathcal{S}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[\int_\Omega u_i(x) g_i(x) d\Omega + \lambda |\partial \Omega_i| \right], \tag{5.8}$$

where $\mathcal{S} = \left\{ u = (u_1, \cdots, u_n) \in BV(\Omega) : u_i(x) = 0, 1, \text{and} \sum_{i=1}^{n} u_i = 1 \right\}$. It is shown in [2][67], that when $\delta t \ll 1$, the length of $\partial\Omega_i \cap \partial\Omega_j$ can be approximated by

$$|\partial\Omega_i \cap \partial\Omega_j| \approx \sqrt{\frac{\pi}{\delta t}} \int_\Omega u_i G_{\delta t} * u_j d\Omega, \tag{5.9}$$

where $*$ represents convolution and

$$G_{\delta t}(x) = \frac{1}{4\pi\delta t} exp(-\frac{|x|^2}{4\delta t})$$

is the heat kernel.

The above integral measures the amount of heat that escapes from Ω_j to Ω_i. That can estimate the size of the boundary between Ω_i and Ω_j after normalization. Therefore,

$$|\partial\Omega_i| \approx \sum_{j=1, j\neq i}^{n} \sqrt{\frac{\pi}{\delta t}} \int_\Omega u_i G_{\delta t} * u_j d\Omega. \tag{5.10}$$

Hence the total energy can be approximated by

$$\mathcal{E}^{\delta t}(u_1, \cdots, u_n) = \sum_{i=1}^{n} \int_\Omega \left(u_i g_i + \lambda \sum_{j=1, j\neq i}^{n} \frac{\sqrt{\pi}}{\sqrt{\delta t}} u_i G_{\delta t} * u_j \right) d\Omega. \tag{5.11}$$

Now, (5.8) becomes

$$u = \underset{(u_1, \cdots, u_n) \in \mathcal{S}}{\operatorname{argmin}} \mathcal{E}^{\delta t}(u_1, \cdots, u_n) \tag{5.12}$$

Notice that the problem is to minimize a non-convex energy functional defined on a non-convex admissible set. However, we can relax it to a problem defined on a convex admissible set by finding $u = (u_1, \cdots, u_n)$ such that

$$u = \underset{(u_1, \cdots, u_n) \in \mathcal{K}}{\operatorname{argmin}} \mathcal{E}^{\delta t}(u_1, \cdots, u_n) \tag{5.13}$$

where $\mathcal{K}$ is the convex hull of $\mathcal{S}$:

$$\mathcal{K} = \left\{ u = (u_1, \cdots, u_n) \in BV(\Omega) : 0 \leq u_i(x) \leq 1, \text{and} \sum_{i=1}^{n} u_i = 1 \right\}. \tag{5.14}$$

The following lemma shows that the relaxed problem (5.13) is equivalent to the original problem (5.12). Therefore we can solve the relaxed problem (5.13) instead.

Lemma 5.1.1. *Let $u = (u_1, \cdots, u_n)$. Then*

$$\operatorname*{argmin}_{u\in\mathcal{S}} \mathcal{E}^{\delta t}(u) = \operatorname*{argmin}_{u\in\mathcal{K}} \mathcal{E}^{\delta t}(u). \tag{5.15}$$

Proof. Let $v = (v_1, \cdots, v_n) \in \mathcal{K}$ be a minimizer of $\mathcal{E}^{\delta t}(u)$ on $\mathcal{K}$. Since $\mathcal{S} \subset \mathcal{K}$, we have

$$\begin{aligned}\mathcal{E}^{\delta t}(v) &= \min_{v\in\mathcal{K}} \mathcal{E}^{\delta t}(u)\\ &\leq \min_{v\in\mathcal{S}} \mathcal{E}^{\delta t}(u).\end{aligned}$$

Therefore, we only need to prove that $v \in \mathcal{S}$.

We prove it by contradiction. If $v \notin \mathcal{S}$, then there exists a set $A \subseteq \Omega$ ($|A| > 0$) and a constant $0 < \epsilon < \frac{1}{2}$ such that for some $k, l \in \{1, \cdots, n\}$ with $k \neq l$,

$$v_k(x), v_l(x) \in (\epsilon, 1-\epsilon), \quad \forall x \in A.$$

Denote

$$u_m^t(x,t) = v_m(x) + t(\delta_{m,l} - \delta_{m,k})\chi_A(x)$$

for $m = 1, \cdots, n$ where $\chi_A(x)$ represents the characteristic function of region A and

$$\delta_{m,l} = \begin{cases} 1 & m = l \\ 0 & m \neq l. \end{cases}$$

When $-\epsilon \leq t \leq \epsilon$, we have $u_m^t(x,t) \geq 0$ and $\sum_{m=1}^{n} u_m^t(x,t) = 1$ so that $u^t(x,t) = (u_1^t(x,t), \cdots, u_n^t(x,t)) \in \mathcal{K}$. Now denote

$$f^m = \int_\Omega v_m f d\Omega, \quad V^m = \int_\Omega v_m d\Omega, \quad f^A = \int_\Omega \chi_A f d\Omega. \tag{5.16}$$

Then

$$\begin{aligned}\int_\Omega u_m^t f d\Omega &= \int_\Omega v_m f d\Omega + t\int_\Omega (\delta_{ml} - \delta_{mk})\chi_A f d\Omega\\ &= f^m + t(\delta_{ml} - \delta_{mk}) f^A \qquad (5.17)\\ \int_\Omega u_m^t d\Omega &= \int_\Omega v_m d\Omega + t\int_\Omega (\delta_{ml} - \delta_{mk})\chi_A d\Omega\\ &= V^m + t(\delta_{ml} - \delta_{mk})|A| \qquad (5.18)\end{aligned}$$

Let

$$C_m = \frac{\int_\Omega u_m^t f d\Omega}{\int_\Omega u_m^t d\Omega}.$$

It is easy to see that C_m depends on t only when $m = l$ or k. We have

$$C_l = \frac{f^l + tf^A}{V^l + t|A|} \quad \text{and} \quad C_k = \frac{f^k - tf^A}{V^k - t|A|}.$$

Then, we calculate the first and second order derivatives of C_l and C_k with respect to t as follows:

$$\begin{aligned}
\frac{dC_l}{dt} &= \frac{f^A}{V^l + t|A|} - \frac{|A|(f^l + tf^A)}{(V^l + t|A|)^2} \\
\frac{dC_k}{dt} &= -\frac{f^A}{V^k - t|A|} + \frac{|A|(f^k - tf^A)}{(V^k - t|A|)^2} \\
\frac{d^2C_l}{dt^2} &= -\frac{2|A|f^A}{(V^l + t|A|)^2} + \frac{2|A|^2(f^l + tf^A)}{(V^l + t|A|)^3} \\
\frac{d^2C_k}{dt^2} &= -\frac{2|A|f^A}{(V^k - t|A|)^2} + \frac{2|A|^2(f^k - tf^A)}{(V^k - t|A|)^3}
\end{aligned} \tag{5.19}$$

A direct calculation then gives

$$\begin{aligned}
\frac{d^2\mathcal{E}^{\delta t}}{dt^2} =& \int_\Omega \sum_{i=1}^n \left(4\frac{du_i^t}{dt}\langle C_i - f, \frac{dC_i}{dt}\rangle + 2u_i^t\langle C_i - f, \frac{d^2C_i}{dt^2}\rangle + 2u_i^t||\frac{dC_i}{dt}||_2^2 \right) d\Omega. \\
& - 4\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} \int_\Omega \chi_A G_{\delta t} * \chi_A d\Omega \\
=& 4\int_\Omega \chi_A \langle C_l - f, \frac{dC_l}{dt}\rangle d\Omega - 4\int_\Omega \chi_A \langle C_k - f, \frac{dC_k}{dt}\rangle d\Omega \\
& + 2\int_\Omega u_l^t \langle C_l - f, \frac{d^2C_l}{dt^2}\rangle d\Omega + 2\int_\Omega u_k^t \langle C_k - f, \frac{d^2C_k}{dt^2}\rangle d\Omega \\
& + 2\int_\Omega u_l^t ||\frac{dC_l}{dt}||_2^2 d\Omega + 2\int_\Omega u_k^t ||\frac{dC_k}{dt}||_2^2 d\Omega \\
& - 4\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} \int_\Omega \chi_A G_{\delta t} * \chi_A d\Omega
\end{aligned} \tag{5.20}$$

where $\langle \alpha, \beta \rangle = \sum_{i=1}^n \alpha_i \beta_i$ for $\alpha, \beta \in R^n$. Evaluating at $t = 0$ and substituting

(5.19) into (5.20), we have

$$\left.\frac{d^2\mathcal{E}^{\delta t}}{dt^2}\right|_{t=0} = \int_\Omega 4\chi_A\langle\frac{f^l}{V^l} - f, \frac{f^A}{V^l} - \frac{|A|f^l}{(V^l)^2}\rangle + 4\chi_A\langle\frac{f^k}{V^k} - f, \frac{f^A}{V^k} - \frac{|A|f^k}{(V^k)^2}\rangle d\Omega \tag{5.21}$$

$$+ 2\int_\Omega v_l\langle\frac{f^l}{V^l} - f, -\frac{2|A|f^A}{(V^l)^2} + \frac{2|A|^2 f^l}{(V^l)^3}\rangle d\Omega \tag{5.22}$$

$$+ 2\int_\Omega v_k\langle\frac{f^k}{V^k} - f, -\frac{2|A|f^A}{(V^k)^2} + \frac{2|A|^2 f^k}{(V^k)^3}\rangle d\Omega \tag{5.23}$$

$$+ 2\int_\Omega v_l||\frac{f^A}{V^l} - \frac{|A|f^l}{(V^l)^2}||_2^2 d\Omega \tag{5.24}$$

$$+ 2\int_\Omega v_k||\frac{f^A}{V^k} - \frac{|A|f^k}{(V^k)^2}||_2^2 d\Omega \tag{5.25}$$

$$- 4\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}\int_\Omega \chi_A G_{\delta t} * \chi_A d\Omega. \tag{5.26}$$

Then, using (5.16) and the definition of $|A|$, we can calculate the above integrals (note that f^l, f^k, f^A, V^l, V^k and $|A|$ in the integrand are all independent of Ω). Therefore,

$$\begin{aligned}&(5.21) + (5.24) + (5.25)\\ =& -\frac{2}{V^l}||\frac{|A|f^l}{V^l} - f^A||_2^2 - \frac{2}{V^k}||\frac{|A|f^k}{V^k} - f^A||_2^2 < 0.\end{aligned} \tag{5.27}$$

Similarly, direct calculations show that (5.22) $= 0$ and (5.23) $= 0$. It is obvious that

$$-4\frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}}\int_\Omega \chi_A G_{\delta t} * \chi_A d\Omega < 0.$$

Combining the above, we have

$$\left.\frac{d^2\mathcal{E}^{\delta t}}{dt^2}\right|_{t=0} < 0.$$

Thus, $v(x) = u(x, 0)$ cannot be a minimizer. This contradicts the assumption.

□

5.1.2 Derivation of the threshold dynamics method

In the following, we show that the minimization problem (5.12) can be solved by an efficient threshold dynamics method. Suppose that we have the k^{th} iteration $(u_1^k, \cdots, u_n^k) \subset \mathcal{S}$. Let $g_i^k = ||C_i^k - f||_2^2$ with

$$C_i^k = \frac{\int_\Omega u_i^k f d\Omega}{\int_\Omega u_i^k d\Omega}.$$

Then the energy functional $\mathcal{E}^{\delta t}(u_1, \cdots, u_n)$ with $g_i = g_i^k$ given above can be linearized near the point $(u_1^k, \cdots, u_n^k)$ by

$$\begin{aligned} &\mathcal{E}^{\delta t}(u_1, \cdots, u_n) \approx \mathcal{E}^{\delta t}(u_1^k, \cdots, u_n^k) \\ &+ \mathcal{L}(u_1 - u_1^k, \cdots, u_n - u_n^k, u_1^k, \cdots, u_n^k) + h.o.t \end{aligned} \tag{5.28}$$

where

$$\begin{aligned} \mathcal{L}(u_1, \cdots, u_n, u_1^k, \cdots, u_n^k) &= \sum_{i=1}^n \int_\Omega \left(u_i g_i^k + \sum_{j=1, j\neq i}^n \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} u_i G_{\delta t} * u_j^k \right) d\Omega \\ &= \sum_{i=1}^n \int_\Omega u_i \left(g_i^k + \sum_{j=1, j\neq i}^n \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} G_{\delta t} * u_j^k \right) d\Omega. \end{aligned} \tag{5.29}$$

We can now determine the next iteration $(u_1^{k+1}, \cdots, u_n^{k+1})$ by minimizing the linearized functional

$$\min_{(u_1, \cdots, u_n) \in \mathcal{K}} \mathcal{L}(u_1, \cdots, u_n, u_1^k, \cdots, u_n^k). \tag{5.30}$$

Denote

$$\phi_i^k := g_i^k + \sum_{j=1, j\neq i}^n \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} G_{\delta t} * u_j^k. \tag{5.31}$$

$$= g_i^k + \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} (1 - G_{\delta t} * u_i^k). \tag{5.32}$$

We have

$$\mathcal{L}(u_1, \cdots, u_n, u_1^k, \cdots, u_n^k) = \sum_{i=1}^n \int_\Omega u_i \phi_i^k d\Omega. \tag{5.33}$$

The optimization problem (5.30) becomes minimizing a linear functional over a convex set. It can be carried out at each $x \in \Omega$ independently. By comparing the coefficients $\phi_i^k(x)$ (non-negative) of $u_i(x)$ in the integrand of (5.33), it is easy to see that the minimum is attained at

$$u_i^{k+1}(x) = \begin{cases} 1 & \text{if } \phi_i^k(x) = \min_l \phi_l^k(x), \\ 0 & \text{otherwise.} \end{cases} \tag{5.34}$$

The following theorem shows that the total energy $\mathcal{E}^{\delta t}$ decreases in the iteration for any $\delta t > 0$. Therefore, our iteration algorithm always converges to a minimum for any initial partition.

Theorem 5.1.1. *Let $(u_1^{k+1}, \cdots, u_n^{k+1})$ be the $k+1^{th}$ iteration derived above, we have*

$$\mathcal{E}^{\delta t}(u_1^{k+1}, \cdots, u_n^{k+1}) \leq \mathcal{E}^{\delta t}(u_1^k, \cdots, u_n^k) \tag{5.35}$$

for all $\delta t > 0$.

Proof. From (5.29), we have

$$\begin{aligned}
&\mathcal{E}^{\delta t}(u_1^k, \cdots, u_n^k) + \sum_{i=1}^n \int_\Omega \sum_{j\neq i, j=1}^n \frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} u_i^k G_{\delta t} * u_j^k d\Omega = \mathcal{L}(u_1^k, \cdots, u_n^k, u_1^k, \cdots, u_n^k) \\
&\geq \mathcal{L}(u_1^{k+1}, \cdots, u_n^{k+1}, u_1^k, \cdots, u_n^k) = \mathcal{E}^{\delta t}(u_1^{k+1}, \cdots, u_n^{k+1}) \\
&+ \sum_{i=1}^n \int_\Omega \left(u_i^{k+1}(g_i^k - g_i^{k+1}) + \sum_{j=1, j\neq i}^n \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} u_i^{k+1} G_{\delta t} * u_j^k \right) d\Omega \\
&- \sum_{i=1}^n \int_\Omega \sum_{j\neq i, j=1}^n \frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} u_i^{k+1} G_{\delta t} * u_j^{k+1} d\Omega.
\end{aligned}$$

That leads to

$$\mathcal{E}^{\delta t}(u_1^k, \cdots, u_n^k) \geq \mathcal{E}^{\delta t}(u_1^{k+1}, \cdots, u_n^{k+1}) + I \tag{5.36}$$

with

$$
\begin{aligned}
I =& \sum_{i=1}^{n} \int_{\Omega} \left(u_i^{k+1}(g_i^k - g_i^{k+1}) + \sum_{j=1, j\neq i}^{n} \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} u_i^{k+1} G_{\delta t} * u_j^k \right) d\Omega \\
& - \sum_{i=1}^{n} \int_{\Omega} \sum_{j\neq i, j=1}^{n} \frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} u_i^{k+1} G_{\delta t} * u_j^{k+1} d\Omega \\
& - \sum_{i=1}^{n} \int_{\Omega} \sum_{j\neq i, j=1}^{n} \frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} u_i^{k} G_{\delta t} * u_j^{k} d\Omega \\
=& I_1 + I_2
\end{aligned}
$$

where

$$
\begin{aligned}
I_1 =& \sum_{i=1}^{n} \int_{\Omega} u_i^{k+1}(g_i^k - g_i^{k+1}) d\Omega \\
I_2 =& \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \int_{\Omega} \frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} u_i^{k+1} G_{\delta t} * (u_j^k - u_j^{k+1}) d\Omega \\
& - \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \int_{\Omega} \frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} (u_i^k - u_i^{k+1}) G_{\delta t} * u_j^k d\Omega.
\end{aligned}
$$

Now, we only need to prove that $I_1 \geq 0$ and $I_2 \geq 0$. From the definition of C_i^{k+1} and using the fact that $\int_\Omega u_i^{k+1} f d\Omega = \int_\Omega u_i^{k+1} d\Omega C_i^{k+1}$, we have

$$
\begin{aligned}
I_1 =& \sum_{i=1}^{n} \int_{\Omega} u_i^{k+1}(||C_i^k - f||_2^2 - ||C_i^{k+1} - f||_2^2) d\Omega \\
=& \sum_{i=1}^{n} \int_{\Omega} u_i^{k+1}(||C_i^k||_2^2 - ||C_i^{k+1}||_2^2 - 2\langle C_i^k - C_i^{k+1}, f\rangle) d\Omega \\
=& \sum_{i=1}^{n} \left\{ \int_{\Omega} u_i^{k+1} d\Omega (||C_i^k||_2^2 - ||C_i^{k+1}||_2^2 - 2\langle C_i^k - C_i^{k+1}, C_i^{k+1}\rangle) \right\} \qquad (5.37) \\
=& \sum_{i=1}^{n} \left\{ \int_{\Omega} u_i^{k+1} d\Omega ||C_i^k - C_i^{k+1}||_2^2 \right\} \geq 0.
\end{aligned}
$$

By changing the order of the two summations in the second part of I_2 and using

the fact that $\sum_{i=1}^{n} u_i^k = 1$ for any k, we obtain

$$\begin{aligned} I_2 =& \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \int_\Omega \frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} (u_i^{k+1} - u_i^k) G_{\delta t} * (u_j^k - u_j^{k+1}) d\Omega \\ =& \sum_{i=1}^{n} \int_\Omega \frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} (u_i^{k+1} - u_i^k) G_{\delta t} * \left(\sum_{j=1, j\neq i}^{n} (u_j^k - u_j^{k+1}) \right) d\Omega \\ =& \sum_{i=1}^{n} \int_\Omega \frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} (u_i^{k+1} - u_i^k) G_{\delta t} * (1 - u_i^k - (1 - u_i^{k+1})) d\Omega \qquad (5.38) \\ =& \sum_{i=1}^{n} \int_\Omega \frac{\lambda\sqrt{\pi}}{\sqrt{\delta t}} (u_i^{k+1} - u_i^k) G_{\delta t} * (u_i^{k+1} - u_i^k) d\Omega \geq 0. \end{aligned}$$

Combining (5.36), (5.37) and (5.38) gives (5.35). □

We are then led to the following iterative thresholding algorithm:

Algorithm:

Step 0. *Given an initial partition* $\Omega_1^0, ..., \Omega_n^0 \subset \Omega$ *and the corresponding* $u_1^0 = \chi_{\Omega_1^0}, ..., u_n^0 = \chi_{\Omega_n^0}$. *Set a tolerance parameter* $\tau > 0$.

Step 1. *Given* k^{th} *iteration* $(u_1^k, \cdots, u_n^k) \subset \mathcal{S}$, *we compute* g_i^k *and the following convolutions for* $i = 1, \cdots, n$*:*

$$\phi_i^k := g_i^k + \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}} (1 - G_{\delta t} * u_i^k) \qquad (5.39)$$

Step 2. *Thresholding: Let*

$$\Omega_i^{k+1} = \left\{ x : \phi_i^k(x) < \min_{j\neq i} \phi_j^k(x) \right\} \qquad (5.40)$$

and define $u_i^{k+1} = \chi_{\Omega_i^{k+1}}$ *where* $\chi_{\Omega_i^{k+1}}$ *represents the charecteristic function of region* Ω_i^{k+1}

Step 3. *Let the normalized* L^2 *difference between successive iterations be*

$$e^{k+1} = \frac{1}{|\Omega|} \int_\Omega \sum_{i=1}^{n} |u_i^{k+1} - u_i^k|^2 d\Omega.$$

If $e^{k+1} \leq \tau$, *stop. Otherwise, go back to step 1.*

Remark 5.1.1. *The convolutions in Step 1 are computed efficiently using FFT with a computational complexity of $O(Nlog(N))$, where N is the total number of pixels. Therefore the total computational cost at each iteration is also $O(Nlog(N))$.*

Remark 5.1.2. *In Step 3, e^k measures the percentage of pixels on which $u_i^{k+1} \neq u_i^k$. Therefore the tolerance τ specifies the threshold of the percentage of pixels changing during the iteration below which the iteration stops.*

5.2 Numerical Results

We now present numerical examples to illustrate the performance of our algorithm. We implement the algorithm in MATLAB. All the computations are carried out on a MacBook Pro laptop with a 3.0GHz Intel(R) Core(TM) i7 processor and 8GB of RAM.

5.2.1 Example 1: Cameraman

We first test our algorithm on the standard cameraman image using two-phase segmentation. Figure 5.1(a) is the original image. We start with the initial contour given in Fig. 5.1(b). We choose $\delta t = 0.03$ and $\lambda = 0.01$. Our algorithm takes only 15 iterations to converge to a complete steady state, i.e. $e^k = 0$ (for $k = 15$) with a total computation time of only 0.1188 seconds. Fig. 5.1(c) gives the final segmentation contour. We also plot the normalized energy $\mathcal{E}^{\delta t}/|\Omega|$ as a function of the iteration number k in Fig.5.2, which verifies the monotone decay of the energy. In fact, the energy decays quickly in the first few iterations and almost reaches steady state in less than 10 iterations.

To study the effect of the parameter λ in the energy (5.11), we run our algorithm

(a) Given Image. (b) Initial Contour. (c) Final Contour.

Figure 5.1: Segmentation results for the classic cameraman image with $\delta t = 0.03$ and $\lambda = 0.01$. The algorithm converges in 15 iterations with a computational time of 0.1188 seconds

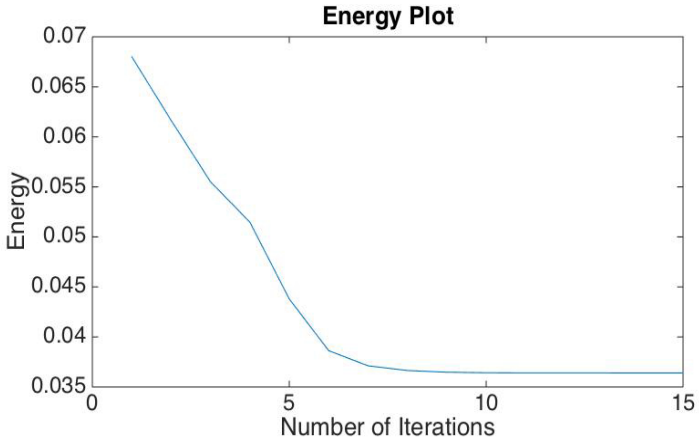
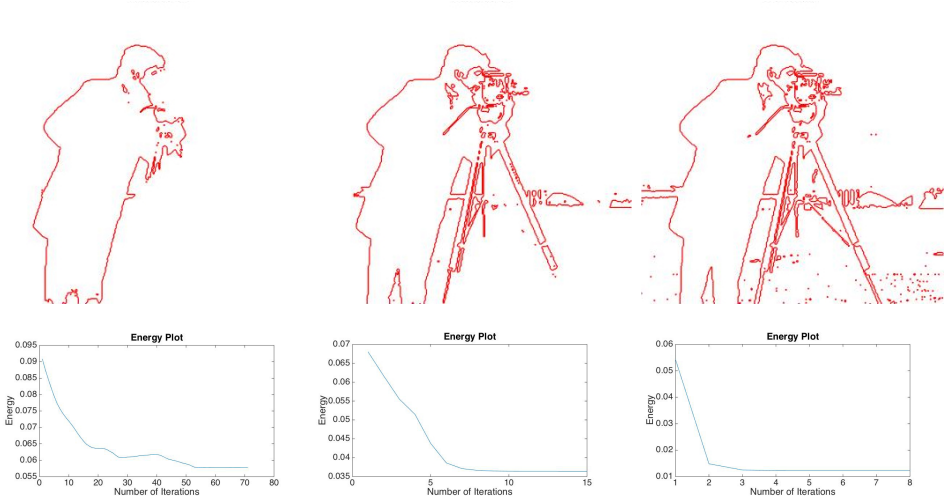


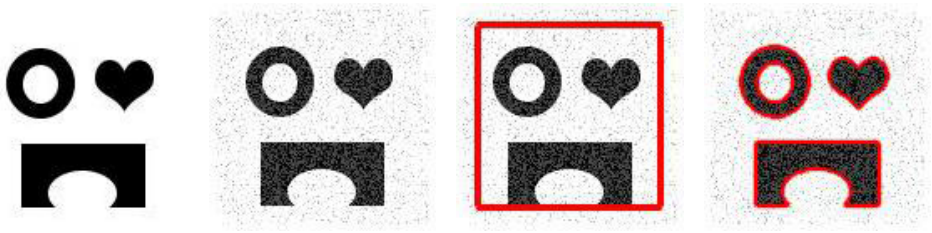
Figure 5.2: Energy curve for the iteration algorithm with $\delta t = 0.03$ and $\lambda = 0.01$.

on the same test image for three different values of $\lambda = 0.001, 0.01$ and 0.025 but with a fixed $\delta t = 0.03$. The final segmentation contours together with the energy curves are shown in Fig. 5.3. As the figure shows, larger $\lambda = 0.025$ turns to smooth out the small-scale structures while smaller $\lambda = 0.001$ would pick up more noisy regions. This is easy to understand since λ measures the relative importance of the contour length and the data term in the Chan-Vese functional to be minimized. A larger λ tends to shorten the total contour length and therefore does not favor small-scale structures. On the other hand, convergence is much faster for a smaller λ while a larger λ would require more iterations to converge as shown by the energy curves.



(d) $\lambda = 0.025$. (e) $\lambda = 0.01$. (f) $\lambda = 0.001$.

Figure 5.3: Segmentation contours and energy curves for $\delta t = 0.03$ and different λ values.



(a) Given Image. (b) Noised Image. (c) Initial Contour. (d) Final Contour.

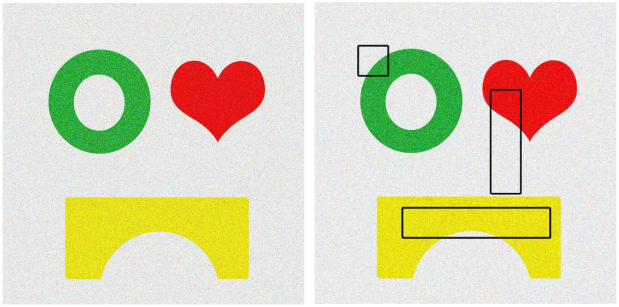
Figure 5.4: Noisy Image Segmentation with $\delta t = 0.03$, $\lambda = 0.1$.

5.2.2 Example 2: Image with heavy noise

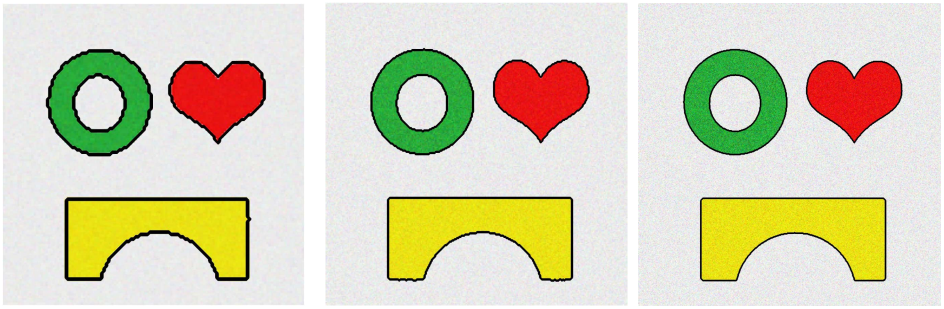
Now, we apply our algorithm to a heavy noised image Fig. 5.4(a). The original image was a clear synthetic one. Gaussian noise with mean 0.6 and variance 0.5 was added to the image to give Fig. 5.4(b). The initial contours are given in Fig. 5.4(c). We apply our two-phase algorithm to the image with $\delta t = 0.03$ and $\lambda = 0.1$. The algorithm converges in 11 iterations with runtimes of 0.019 seconds. Figures. 5.4(d) show the final segmentation result.

5.2.3 Example 3: A synthetic four-phase image

We next use a synthetic color image given in Fig. 5.5(a). The image f is a vector-valued function. Gaussian noise is added with mean 0 and variance 0.04 to each component of image f. The initial contours are given in Fig. 5.5(b). We apply our four-phase algorithm to the image with three different resolutions from 128×128 to 512×512. In each case, $\delta t = 0.01$ and $\lambda = 0.003$. The algorithm converges in $7 \sim 8$ iterations for all resolutions with runtimes of $0.0444, 0.1333, 0.6706$ seconds respectively, which demonstrates good stability of and robustness of our method. Figures. 5.5(c)-5.5(e) show the final segmentation result.



(a) Image with Noise. (b) Initial Contour.

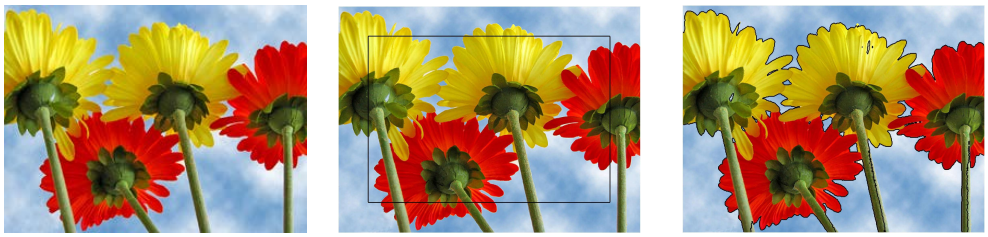


(c) 128×128. (d) 256×256. (e) 512×512.

Figure 5.5: Segmentation for images with different resolutions and with the parameters $\delta t = 0.01$ and $\lambda = 0.003$

5.2.4 Example 4: Flower color image

We now consider an image containing flowers of different colors in Fig. 5.6(a). We first use a two-phase segmentation algorithm with $\delta t = 0.01$ and $\lambda = 0.005$ and the initial contour in Fig. 5.6(b). The algorithm converges in 20 iterations with a runtime of 0.6751 seconds. The final segmentation result is given in Fig. 5.6(c). We also use a four-phase segmentation algorithm with $\delta t = 0.01$ and $\lambda = 0.003$ and the initial contour in Fig. 5.7(a). The algorithm converges in 18 iterations with a runtime of 1.1007 seconds. The final segmentation result is given in Fig. 5.7(b) and 5.7(c)



(a) Given Color Image. (b) Initial Contour. (c) Final Contour.

Figure 5.6: Two-phase segmentation for a 375×500 RGB image and with parameters $\delta t = 0.01$ and $\lambda = 0.005$.

(a) Initial Contour. (b) Final Contour. (c) Four Segments.

Figure 5.7: Four phase segmentation for a 375×500 RGB image with $\delta t = 0.01$ and $\lambda = 0.003$.

Chapter 6

Conclusion and future work

Conclusion: This thesis has discussed efficient threshold dynamics methods for free interface problems. We developed an efficient threshold dynamics method for wetting on rough surfaces. The method is based on minimization of the weighted surface area functional over an extended domain that includes the solid phase. The method is simple, stable with the complexity $O(NlogN)$per time step and is not sensitive to the inhomogeneity or roughness of the solid boundary.

We rigorously analysed the convergence of the threshold dynamics method proposed for wetting dynamics.

We presented an algorithm for the threshold dynamics of interface motion. The algorithm discretizes physical space only in a neighborhood of the interface and applies NUFFT to solve the initial value problem of the heat equation. Unlike many grid based methods where the spatial mesh size is required to be of the same order as the time step size, our numerical experiments show that the spatial mesh size can be chosen based upon the accuracy consideration only and thus is more or less independent of the time step size for our algorithm. Hence, our algorithm is efficient and robust for smooth interfaces if the time step is not very small. When the time step is very small, the number of Fourier modes

in the spectral approximation of the heat kernel becomes excessively large and our algorithm becomes less efficient, especially for three dimensional problems. However, one may use other fast algorithms such as the fast Gauss transform or even asymptotic analysis to obtain the solution of the heat equation for the diffusion stage in this regime. For two dimensional problems, our algorithm has been easily extended to treat piecewise smooth boundaries such as polygons, or multiphase flows such as the wetting problems on a solid surface with given contact angles. For three dimensional problems, it is straightforward to modify our algorithm to deal with the wetting problems on a at solid plane. Our algorithm has also been extended to treat the cases involving topological changes.

We proposed an efficient threshold dynamics method for the Chan-Vese model for multi-phase image segmentation. The algorithm works by alternating the convolution step with the thresholding step and has the optimal computational complexity of $O(NlogN)$ per iteration. We prove that the iterative algorithm has the property of total energy decay. The numerical results show that the method is stable and the number of iterations before convergence is independent of the spacial resolution (for a given image). The relative importance of the different effects in the energy functional is studied by tuning the parameter λ.

Future work: There are many works can be done related to this thesis. Adaptivity is a way to further improve the efficiency and can be done in the framework of structured grids or un-structured grids. And parallelization among different computers is also worthwhile to do due to many fast Fourier transforms in these numerical methods. To further simplify the threshold dynamics method we proposed, we can replace the Gaussian kernel by a piecewise constant disc kernel, then the convolution can be more efficiently calculated by narrow band algorithm (i.e. for each discrete point, select the nearby points in a very narrow band). Meanwhile, we are working on the generalization of the threshold dynamics method to model the anisotropic interface motion and apply it into

many physical problems (e.g. solid de-wetting). Also, we are working on the development of fast solvers for non-local diffusion equations (e.g. fractional diffusion) with applications in the non-local threshold dynamics method for non-local motions. In addition, we are using distance function to replace the characteristic function to generate the motion by arbitrary interface motions including anisotropic motions, high order geometric motions, volume preserving motions, non-local motions, and so on.

Bibliography

[1] G. Alberti and A. DeSimone. Wetting of rough surfaces: A homogenization approach. *Proc. R. Soc. A*, 451:79–97, 2005.

[2] Giovanni Alberti and Giovanni Bellettini. A non-local anisotropic model for phase transitions: asymptotic behaviour of rescaled energies. *Euro. J. Appl. Math.*, 9(03):261–284, 1998.

[3] Samuel M Allen and John W Cahn. A correction to the ground state of fcc binary ordered alloys with first and second neighbor pairwise interactions. *Scripta Metallurgica*, 7(12):1261–1264, 1973.

[4] Samuel Miller Allen and John W Cahn. Ground state structures in ordered binary alloys with second neighbor interactions. *Acta Metallurgica*, 20(3):423–433, 1972.

[5] Fred Almgren, Jean E Taylor, and Lihe Wang. Curvature-driven flows: a variational approach. *SIAM J. Control. Optim.*, 31(2):387–438, 1993.

[6] Fred Almgren and Lihe Wang. Mathematical existence of crystal growth with Gibbs-Thomson curvature effects. *J. Geom. Anal.*, 10(1):1–100, 2000.

[7] Luigi Ambrosio and Vincenzo Maria Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. *Comm. Pure Appl. Math.*, 43(8):999–1036, 1990.

[8] Egil Bae, Jing Yuan, and Xue-Cheng Tai. Global minimization for continuous multiphase partitioning problems using a dual approach. *International journal of computer vision*, 92(1):112–129, 2011.

[9] G. Barles and C. Georgelin. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. *SIAM J. Numer. Anal.*, 32(2):484–500, 1995.

[10] A. L. Bertozzi, S. Esedoglu, and A. Gillette. Inpainting of binary images using the cahn-hilliard equation. *IEEE Transactions on image processing*, 16(1):285–291, 2007.

[11] Eric Bonnetier, Elie Bretin, and Antonin Chambolle. Consistency result for a non monotone scheme for anisotropic mean curvature flow. *arXiv preprint arXiv:1005.4794*, 2010.

[12] Andrea Braides. *Approximation of free-discontinuity problems*. Number 1694. Springer Science & Business Media, 1998.

[13] John W Cahn and John E Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of chemical physics*, 28(2):258–267, 1958.

[14] Xiaohao Cai, Raymond Chan, and Tieyong Zeng. A two-stage image segmentation method using a convex variant of the mumford–shah model and thresholding. *SIAM J. Imaging Sci.*, 6(1):368–390, 2013.

[15] A. Chambolle and M. Novaga. Convergence of an algorithm for the anisotropic and crystalline mean curvature flow. *SIAM J. Math. Anal.*, 37(6):1978–1987, 2006.

[16] Tony F. Chan, Selim Esedoḡlu, and Mila Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.*, 66(5):1632–1648, 2006.

[17] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE-IP*, 10(2):266–277, 2001.

[18] Long-Qing Chen. Phase-field models for microstructure evolution. *Annual review of materials research*, 32(1):113–140, 2002.

[19] X. Chen, X.-P. Wang, and X. Xu. Effective contact angle for rough boundary. *Physica D*, 242:54–64, 2013.

[20] X. Chen, X.-P. Wang, and X. Xu. Analysis of the cahn-hilliard equation with relaxation boundary condition modelling contact angle. *Arch. Rational Mech. Anal.*, 213:1–24, 2014.

[21] Xinfu Chen, Xiao-Ping Wang, and Xianmin Xu. Effective contact angle for rough boundary. *Physica D: Nonlinear Phenomena*, 242(1):54–64, 2013.

[22] P.G. de Gennes, F. Brochard-Wyart, and D. Quere. *Capillarity and Wetting Phenomena*. Springer Berlin, 2003.

[23] K. Deckelnick, G. Dziuk, and C. M. Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta numerica*, 14:139–232, 2005.

[24] Jesús Ildefonso Diaz, Juan Luis Vazquez, MA Herrero, and Amable Linan. *Free boundary problems: theory and applications*, volume 323. CRC Press, 1995.

[25] Bin Dong, Aichi Chien, and Zuowei Shen. Frame based segmentation for medical images. *Commun. Math. Sci.*, 9(2):551–559, 2010.

[26] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM J. Sci. Comput.*, 14:1368–1393, 1993.

[27] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. II. *Appl. Comput. Harmon. Anal.*, 2:85–100, 1995.

[28] H. Dym and H. P. McKean. *Fourier Series and Integrals*. Academic Press, 1972.

[29] C. M. Elliott and V. Styles. Computations of bidirectional grain boundary dynamics in thin metallic films. *J. Comput. Phys.*, 187(2):524–543, 2003.

[30] Matt Elsey and Selim Esedoglu. Threshold dynamics for anisotropic surface energies. Technical report, Technical report, UM, 2016. Under review, 2016.

[31] H. Y. Erbil. The debate on the dependence of apparent contact angles on drop contact area or three-phase contact line: a review. *Surface Science Reports*, 69(4):325–365, 2014.

[32] Selim Esedog, Yen-Hsi Richard Tsai, et al. Threshold dynamics for the piecewise constant mumford–shah functional. *J. Comput. Phys.*, 211(1):367–384, 2006.

[33] Selim Esedoglu and Felix Otto. Threshold dynamics for networks with arbitrary surface tensions. *Comm. Pure Appl. Math.*, 2015.

[34] Selim Esedoglu, Steven Ruuth, and Richard Tsai. Threshold dynamics for high order geometric motions. *Interfaces and Free Boundaries*, 10(3):263–282, 2008.

[35] S. Esodoglu and P. Smereka. A variational formulation for a level set representation of multiphase flow and area preserving curvature flow. *Comm. Math. Sci.*, 6(1):125–148, 2008.

[36] L. C. Evans. Convergence of an algorithm for mean curvature motion. *Indiana Math. J.*, 42(2):533–557, 1993.

[37] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2nd edition, 2010.

[38] C. W. Extrand. Model for contact angles and hysteresis on rough and ultraphobic surfaces. *Langmuir*, 18:7991–7999, 2002.

[39] David J Eyre. An unconditionally stable one-step scheme for gradient systems. *Unpublished article*, 1998.

[40] Min Gao and X.P. Wang. A gradient stable scheme for a phase field model for the moving contact line problem. *J. Comput. Phys.*, 231:1372–386, 2012.

[41] Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM J. Imaging. Sci.*, 2(2):323–343, 2009.

[42] L. Greengard and J. Y. Lee. Accelerating the nonuniform fast Fourier transform. *SIAM Rev.*, 46(3):443–454, 2004.

[43] L. Greengard and P. Lin. Spectral approximation of the free-space heat kernel. *Appl. Comput. Harmon. Anal.*, 9(1):83–97, 2000.

[44] L. Greengard and J. Strain. The fast Gauss transform. *SIAM J. Sci. Statist. Comput.*, 12(1):79–94, 1991.

[45] L. Greengard and X. Sun. A new version of the fast Gauss transform. *Proc. Int. Cong. Math.*, III:575–584, 1998.

[46] Charles AR Hoare. Quicksort. *The Computer Journal*, 5(1):10–16, 1962.

[47] T. Ilmanen. Lectures on mean curvature flow and related equations, 1998.

[48] K. Ishii. Optimal rate of convergence of the Bence-Merriman-Osher algorithm for motion by mean curvature. *SIAM J. Math. Anal.*, 37(3):841–866, 2005.

[49] Shidong Jiang, Dong Wang, and Xiaoping Wang. An efficient boundary integral scheme for the MBO threshold dynamics method via NUFFT. *To appear in J. Sci. Comput.*

[50] David Kay and Richard Welford. A multigrid finite element solver for the cahn–hilliard equation. *J. Comput. Phys.*, 212(1):288–304, 2006.

[51] Junseok Kim, Kyungkeun Kang, and John Lowengrub. Conservative multigrid methods for cahn–hilliard fluids. *J. Comput. Phys.*, 193(2):511–543, 2004.

[52] Robert V Kohn and Peter Sternberg. Local minimisers and singular perturbations. *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, 111(1-2):69–84, 1989.

[53] Catherine Kublik, Selim Esedoglu, and Jeffrey A Fessler. Algorithms for area preserving flows. *SIAM J. Sci. Comput.*, 33(5):2382–2401, 2011.

[54] JS Langer. Models of pattern formation in first-order phase transitions. *Directions in Concensed Matter Physics: Memorial Volume in Honor of Shang-Keng Ma. Edited by GRINSTEIN S ET AL. Published by World Scientific Publishing Co. Pte. Ltd., 1986. ISBN# 9789814415309, pp. 165-186*, pages 165–186, 1986.

[55] Tim Laux and Felix Otto. Convergence of the thresholding scheme for multi-phase mean-curvature flow. *arXiv preprint arXiv:1602.05857*, 2016.

[56] Tim Laux and Drew Swartz. Convergence of thresholding schemes incorporating bulk effects. *arXiv preprint arXiv:1601.02467*, 2016.

[57] J. Y. Lee, L. Greengard, and Z. Gimbutas. NUFFT Version 1.3.2 Software Release. http://www.cims.nyu.edu/cmcl/nufft/nufft.html, 2009.

[58] S. Leung and H. Zhao. A grid based particle method for moving interface problems. *J. Comput. Phys.*, 228(8):2993–3024, 2009.

[59] J. R. Li and L. Greengard. On the numerical solution of the heat equation. I. Fast solvers in free space. *J. Comput. Phys.*, 226(2):1891–1901, 2007.

[60] Chun Liu and Jie Shen. A phase field model for the mixture of two incompressible fluids and its approximation by a fourier-spectral method. *Physica D: Nonlinear Phenomena*, 179(3):211–228, 2003.

[61] Stephan Luckhaus and Thomas Sturzenhecker. Implicit time discretization for the mean curvature flow equation. *Calculus of variations and partial differential equations*, 3(2):253–271, 1995.

[62] P. Mascarenhas. *Diffusion generated motion by mean curvature*. Department of Mathematics, University of California, Los Angeles, 1992.

[63] Ekaterina Merkurjev, Tijana Kostic, and Andrea L Bertozzi. An MBO scheme on graphs for classification and image processing. *SIAM J. Imag. Sci.*, 6(4):1903–1930, 2013.

[64] B. Merriman, J. K. Bence, and S. Osher. *Diffusion generated motion by mean curvature*. UCLA CAM Report 92-18, 1992.

[65] B. Merriman, J. K. Bence, and S. J. Osher. Motion of multiple junctions: A level set approach. *J. Comput. Phys.*, 112(2):334–363, 1994.

[66] Barry Merriman and Steven J Ruuth. Convolution-generated motion and generalized Huygens' principles for interface motion. *SIAM J. on Appl. Math.*, 60(3):868–890, 2000.

[67] Michele Miranda, Diego Pallara, Fabio Paronetto, and Marc Preunkert. Short-time heat flow and functions of bounded variation in r^n. In *Annales-Faculte des Sciences Toulouse Mathematiques*, volume 16, page 125. Université Paul Sabatier, 2007.

[68] Amar Mitiche and Ismail Ben Ayed. *Variational and level set methods in image segmentation*, volume 5. Springer Science & Business Media, 2010.

[69] D. E. Müller. A method for solving algebraic equations using an automatic computer. *Math. Tables Aids Comput.*, 10:208–215, 1956.

[70] W. W. Mullins. Two-dimensional motion of idealized grain boundaries. *J. Appl. Phys.*, 27(8):900–904, 1956.

[71] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42(5):577–685, 1989.

[72] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[73] D. Quere. Wetting and roughness. *Annu. Rev. Mater. Res.*, 38:71–99, 2008.

[74] B. Randol. On the Fourier transform of the indicator function of a planar set. *Trans. Amer. Math. Soc.*, 139:271–278, 1969.

[75] S. J. Ruuth. *Efficient Algorithms for Diffusion-Generated Motion by Mean Curvature*. PhD thesis, University of British Columbia, 1996.

[76] S. J. Ruuth. A diffusion-generated approach to multiphase motion. *J. Comput. Phys.*, 145(1):166–192, 1998.

[77] S. J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *J. Comput. Phys.*, 144(2):603–625, 1998.

[78] S. J Ruuth and B. TR Wetton. A simple scheme for volume-preserving motion by mean curvature. *J. Sci. Comput.*, 19(1-3):373–384, 2003.

[79] Steven Ruuth. An algorithm for generating motion by mean curvature. *ICAOS'96*, pages 82–91, 1996.

[80] Steven J Ruuth and Barry Merriman. Convolution–thresholding methods for interface motion. *J. Computat. Phys.*, 169(2):678–707, 2001.

[81] Y. Shi and X.-P. Wang. Modeling and simulation of dynamics of three-component flows on solid surface. *Japan J. Indust. Appl. Math.*, 31(3):611–631, 2014.

[82] F. Stenger. Numerical methods based on Whittaker cardinal, or sinc functions. *SIAM Rev.*, 23(2):165–224, 1981.

[83] K. Svadlenka, E. Ginder, and S. Omata. A variational method for multiphase volume-preserving interface motions. *J. Comput. Appl. Math.*, 257:157–179, 2014.

[84] L. N. Trefethen. *Spectral Methods in MATLAB*. SIAM, Philadelphia, 2000.

[85] Andy Tsai, Anthony Yezzi, and Alan S Willsky. Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE-IP*, 10(8):1169–1186, 2001.

[86] Luminita A Vese and Tony F Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *Int',I J.Computer vision*, 50(3):271–293, 2002.

[87] Benjamin P Vollmayr-Lee and Andrew D Rutenberg. Fast and accurate coarsening simulation with an unconditionally stable time step. *Physical Review E*, 68(6):066703, 2003.

[88] Dong Wang, Haohan Li, Xiaoyu Wei, and Xiaoping Wang. An efficient iterative thresholding method for image segmentation. *arXiv preprint arXiv:1608.01431*, 2016.

[89] Dong Wang, Xiaoping Wang, and Yaguang Wang. The dynamics of three-phase triple junction and contact points. *to appear in SIAM J. Appl. Math.*, 2017.

[90] Ke Wei, Xue-Cheng Tai, Tony F. Chan, and Shingyu Leung. Primal-dual method for continuous max-flow approaches. In *Computational Vision and*

Medical Image Processing V: Proceedings of the 5th Eccomas Thematic Conference on Computational Vision and Medical Image Processing (VipIMAGE 2015, Tenerife, Spain, October 19-21, 2015), page 17. CRC Press, 2015. 00000.

[91] G. Whyman, E. Bormashenko, and T. Stein. The rigorous derivative of young, cassie-baxter and wenzel equations and the analysis of the contact angle hysteresis phenomenon. *Chem. Phys. Letters*, 450:355–359, 2008.

[92] David E Womble. A front-tracking method for multiphase free boundary problems. *SIAM J. Numer. Anal.*, 26(2):380–396, 1989.

[93] X. Xu and X. P. Wang. Analysis of wetting and contact angle hysteresis on chemically patterned surfaces. *SIAM J. Appl. Math.*, 71:1753–1779, 2011.

[94] X. Xu and X. P. Wang. The modified Cassie's equation and contact angle hysteresis. *Colloid Polym. Sci.*, 291:299–306, 2013.

[95] Xianmin Xu, Dong Wang, and Xiao-Ping Wang. An efficient threshold dynamics method for wetting on rough surfaces. *J. Comput. Phys.*, 330:510–528, 2017.

[96] T. Young. An essay on the cohesion of fluids. *Philos. Trans. R. Soc. London*, 95:65–87, 1805.

[97] Jing Yuan, Egil Bae, and Xue-Cheng Tai. A study on continuous max-flow and min-cut approaches. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2217–2224. IEEE, 2010. 00148.

[98] P. Yue, J. Feng, C. Liu, and J. Shen. A diffuse-interface method for simulating two-phase flows of complex fluids. *J. Fluid Mech.*, 515:293–317, 2004.

[99] H. Zhao, T. Chan, B. Merriman, and S. J. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127(1):179–195, 1996.