ABSTRACT

Title of dissertation: MODELING, SIMULATING, AND CONTROLLING THE FLUID DYNAMICS OF ELECTRO-WETTING ON DIELECTRIC

Shawn W. Walker, Doctor of Philosophy, 2007

Dissertation directed by: Professor Benjamin Shapiro
Department of Aerospace Engineering
and
Professor Ricardo H. Nochetto
Department of Mathematics

This work describes the modeling and simulation of a parallel-plate Electrowetting On Dielectric (EWOD) device that moves fluid droplets through surface tension effects. The fluid dynamics are modeled by Hele-Shaw type equations with a focus on including the relevant boundary phenomena. Specifically, we include contact angle saturation, hysteresis, and contact line pinning into our model. We show that these extra boundary effects are needed to make reasonable predictions of the correct shape and time scale of droplet motion. We compare our simulation to experimental data for five different cases of droplet motion that include splitting and joining of droplets. Without these boundary effects, the simulation predicts droplet motion that is much faster than in experiment (up to 10-20 times faster).

We present two different numerical implementations of our model. The first uses a level set method, and the second uses a variational method. The level set method provides a straightforward way of simulating droplet motion with topological changes. However, the variational method was pursued for its robust handling of curvature and mass conservation, in addition to being able to easily include a phenomenological model of contact line pinning using a variational inequality. We are also able to show that the variational form of the time-discrete model satisfies a well-posedness result. Our numerical implementations are fast and are being used to design algorithms for the precise control of micro-droplet motion, mixing, and splitting.

We demonstrate micro-fluidic control by developing an algorithm to steer individual particles inside the EWOD system by control of actuators already present in the system. Particles are steered by creating time-varying flow fields that carry the particles along their desired trajectories. Results are demonstrated using the model given above. We show that the current EWOD system [29] at the University of California in Los Angeles (UCLA) contains enough control authority to steer a single particle along arbitrary trajectories and to steer two particles, at once, along simple paths. We also show that particle steering is limited by contact angle saturation and by the small number of actuators available in the EWOD system.

MODELING, SIMULATING, AND CONTROLLING THE FLUID DYNAMICS OF ELECTRO-WETTING ON DIELECTRIC

by

Shawn W. Walker

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2007

Advisory Committee:

Dr. Benjamin Shapiro, Chair/Advisor
Dr. Ricardo H. Nochetto, Co-Advisor
Dr. Robert Sanner
Dr. Christopher Cadou
Dr. Howard Elman, Dean's Representative

DEDICATION

I dedicate this work to the memory of my brother Steven Wayne Walker, Jr. (1970 - 2003)

ACKNOWLEDGMENTS

I would like to thank my parents, Steve and Dale, for being supportive during my time in graduate school, and especially for helping me move from house to house through the years. I also thank my girlfriend, Marilyn Perry, for all her love, support, and dinner making during my PhD studies. I would not have been nearly as productive without her.

Next, I want to thank my advisor, Ben Shapiro, for giving me the freedom to pursue different research directions, debating ideas, and funding me. I appreciate very much his help in writing this dissertation and our journal papers. And I want to thank my co-advisor, Ricardo Nochetto, for the time he has invested in my development, his input on research issues, and for helping me get a job at Courant. He has also been a great help in writing this dissertation. I thank my PhD committee, Robert Sanner, Chistopher Cadou, and Howard Elman for reading my dissertation and providing comments. I also thank Andrea Bonito for reading the dissertation and making comments, and for our many discussions on the "nightmare".

I thank the UCLA research group, namely CJ Kim, Robin Garrell, Sung Kwon Cho, Jian Gong, and Hsiang-Wei Lu for providing information on the EWOD device and experimental data for splitting and moving droplets. In particular, I thank Jian Gong for his efforts in providing much needed experimental data.

I want to thank Steve Jordan whom I met at Yale while taking a Linear

Systems course. He helped give me the confidence to stay in graduate school and continue my studies. Lastly, I thank Serge Lang for teaching me the basics of mathematical analysis. His two semester course (at Yale) has proved invaluable during my time at Maryland.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

In 1875, Gabriel Lippmann demonstrated, through rigorous theory and experiments, a relationship between electrical and surface tension phenomena [94] (see [103] for an English translation). This relationship allows for controlling the shape and motion of a liquid meniscus (i.e. liquid-gas interface) through the use of an applied voltage. The liquid surface changes shape when a voltage is applied in order to minimize the total energy of the system (i.e. the sum of the surface tension energy and electrical energy). In his seminal paper, he showed applications of this effect ranging from allowing sensitive voltage measurements to a working electro-capillary motor. Today, this effect is known as *electrowetting* and has seen a resurgence in modern applications in the area of Micro-Electro-Mechanical Systems (MEMS). Some of these applications include cell phone camera lenses [15], video speed electronic paper [65], and 'lab-on-a-chip' devices [62].

The main topic of this thesis concerns the fluid dynamics of water droplets actuated through electrowetting in a specific system configuration. To begin, we review the current literature and applications of electrowetting. Next, we give a description of the particular electrowetting configuration in which we are interested. Then, we give an overview of various numerical methods for simulating two-phase flow which is important for our application. Finally, we give a thesis outline.

1.1 Literature Overview and Applications of Electro-Wetting

Well designed MEMS devices take advantage of the large surface-to-volume ratios found at the micro-scale. In particular, micro-fluidic devices often exploit surface tension forces to actuate or control liquids [66], [33], [53]. Electro-wetting refers to using electrical fields to effectively modify surface tension effects [94], [103], [34] (see [82] and [81] for some fascinating experimental demonstrations). This allows for the manipulation of fluid droplets at the micro-scale.

Applications for these devices range from micro-fluid transport [118], mixing [108], dispensing [113], and 'lab-on-a-chip' devices that automate functions, such as sensing and testing of biological samples [48], [131], to tunable optical fiber devices [25], [75], reflective displays [115], and light valves [68], [67]. See Figure 1.1 for more.

This thesis is concerned with modeling a specific variant of electrowetting called Electrowetting-On-Dielectric (EWOD) [29], which has an extra insulating layer to enhance its operation (see Section 1.2 for a full description of the EWOD device). See [99] for an initial experimental demonstration and [14] for an analysis of the advantage of using a dielectric insulating layer in an electrowetting system.

Similar applications exist for the EWOD device as well, such as mass spectrometry [144], [101], mixing [55], 'lab-on-a-chip' [62], micro-injection [73], and particle separation and concentration control [27]. Potential uses of these technologies could be for controlled mixing of chemicals and automated DNA testing.

Ultimately, these applications will need accurate fluid dynamical control in order to execute their many subtasks (i.e. particle control, precise droplet motion,

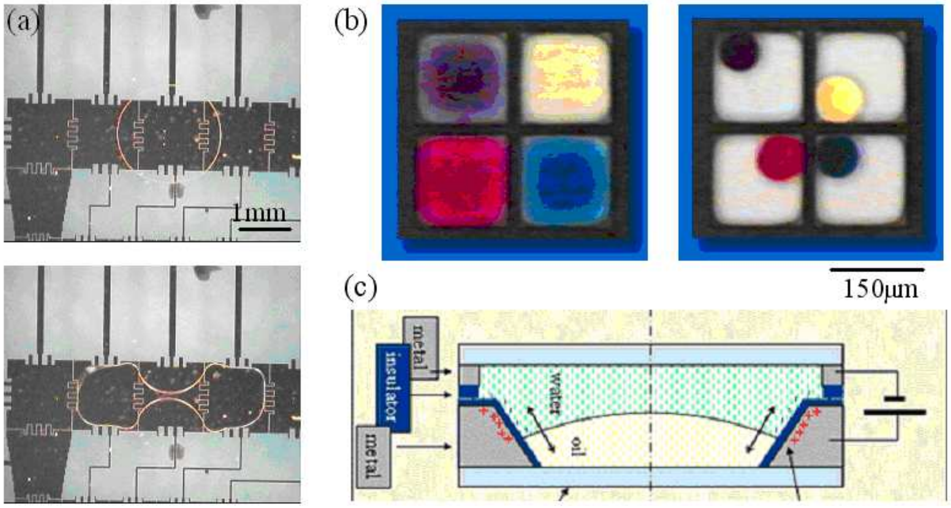


Figure 1.1: Electrowetting applications. (a) The UCLA ‘lab-on-a-chip’ system. View is from above through a transparent top electrode. The six bottom electrodes are the black squares between the jagged white lines. (b) The Phillips/Liquavista liquid color pixels. Left: colored oils are wetted; right: voltage actuation has de-wetted the oils revealing a reflective white background. The 30 micrometer diameter beaded oil drops are too small to be seen by the naked eye, hence the pixels on the left appear to be colored and the ones on the right look bright white. The small size and fast switching speed of these pixels allows them to be fabricated on thin and flexible substrates and to function faster and brighter than regular LCD pixels. This allows, for example, viewing of movies on a laptop in bright sunlight. (c) A schematic diagram of the first liquid cell-phone lens by Varioptic. (Figures (a), (b), and (c) courtesy of Chang-Jin (‘CJ’) Kim at UCLA, Hans Feil and Rob Hayes at Liquavista/Phillips, and Bruno Berge at Varioptic.)

splitting, optimal mixing, etc.). But this will also require accurate models to help design robust controllers as well as guide device optimization. Fabricating EWOD systems usually takes more than eight months per device. Hence, having a predictive tool is clearly desirable. Furthermore, these models must be convenient and cheap to use in order to fit within available control design and optimization methodologies.

Other modeling efforts of EWOD include [29], which gives a basic model of how the device parameters affect droplet splitting. Equilibrium models for the shape of sessile drops on a charging dielectric plate are given in [139] and [124]. In particular, [139] considers a conducting liquid on top of an insulating layer and the effect of charge trapping at high voltage on contact angle saturation. In [124], Shapiro, et al use an energy minimization framework to show that liquid resistance can lead to contact angle saturation in the EWOD devices. Other energy based methods include [10], which they use to ascertain performance parameters for electrowetting-induced droplet actuation. An alternative, lumped parameter, electro-mechanical model for a one dimensional (1-D) liquid column actuated by electrowetting is given in [77] for the equilibrium case and in [143] for the dynamics. In addition, a dynamic model of the contact angle variation for a spreading axisymmetric drop is given in [36], and in [97], a diffuse interface model and simulation of droplet motion is compared to experiments on a scaled-up version of the electrowetting device.

This thesis presents a distributed parameter model of EWOD fluid dynamics that is able to approximately capture the evolution of a droplet's liquid-gas interface in two dimensions. Our model includes a rough approximation of contact angle hysteresis, which is different than, though analogous to, the contact line friction model discussed in [143] and [36]. Also, we include a phenomenological model of contact line pinning that can account for ‘sticking’ droplets, and is similar in spirit to static (Coulomb) friction.

The simulation of our model is done using two different techniques, both of which are sufficiently fast and low dimensional to use in controller design. The first method uses the level set method [105] and is advantageous for capturing droplet split and merge events (i.e. topological changes). The second method uses explicit front tracking of the interface with a variational formulation of the governing equations. Furthermore, we augment the variational method with the work in [140] to allow for topological changes. See Section 1.3 for more discussion on different general techniques for simulating two-phase droplet motion.

Other computational models of electrowetting exist, such as [92], which assumes quasi-static behavior of the droplet (i.e. no internal fluid effects) and uses Surface Evolver [19] to compute droplet shapes for device optimization. In [13], they developed a CFD model for transport of biological species inside an EWOD-driven droplet. Their simulation uses a Volume of Fluid (VoF) method to track the droplet motion but does not give precise information about the liquid-gas interface shape. In [100], they also use a VoF method to simulate a zero-leakage micro-valve actuated by electrowetting. And in [135], they perform a finite element analysis for deforming dielectric droplets in an electric field.

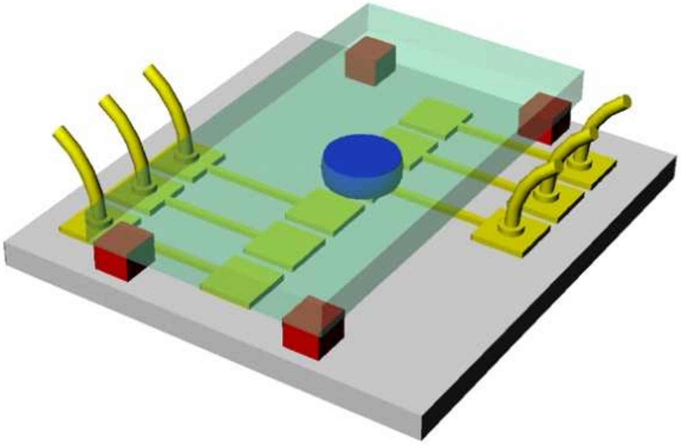


Figure 1.2: Schematic of sample EWOD device (courtesy of CJ Kim at UCLA). This EWOD system consists of two parallel plates with the top plate (transparent) acting as a ground electrode and the bottom plate containing a grid of embedded electrodes. Wires are shown leading off of the device and connecting to an external voltage source (not shown). The small cubes near the edges are 'spacers' to ensure the channel height is uniform. In practice, two pieces of cellophane tape (one on each side) are used as spacers. See Figure 1.3 for a cross-sectional view.

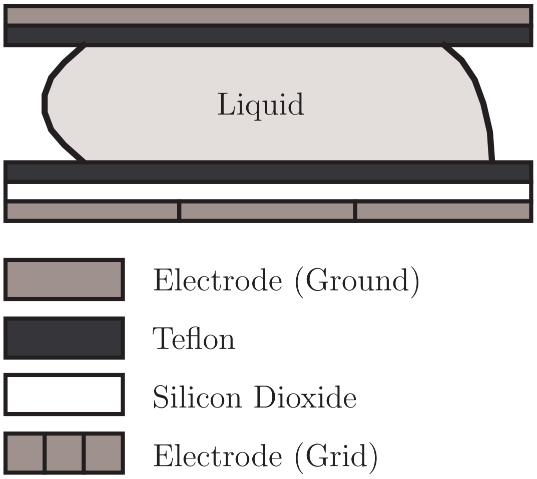


Figure 1.3: Cross-sectional view of the EWOD device. The top layer is a (transparent) ground electrode, followed by a coating of Teflon and a liquid droplet. Underneath the droplet is a coating of Teflon, followed by a layer of silicon dioxide (SiO_2) that acts as a dielectric and improves device performance. On the bottom is a grid of electrodes that can be actuated individually to induce droplet motion.

1.2 Description Of The EWOD System

A schematic of an EWOD device is given in Figure 1.2, while Figure 1.3 shows a cross-sectional view. The device consists of a sandwich of various layers listed from top to bottom as: top (transparent) electrode, hydrophobic Teflon coating, droplets of water (here only one droplet is shown), another Teflon coating, a layer of solid dielectric silicon dioxide, and an underlying grid of electrodes. There are also spacers to ensure that the channel height is uniform.

The basic principle of operation is that the liquid-gas interface of the droplet can be locally deformed by capacitively charging the silicon dioxide layer underneath it. The induced motion of the droplet is due to competing effects of energy storage between the dielectric layer (and external charging source) and the surface energy of the liquid-gas interface [124]. In other words, the EWOD system attempts to minimize its total energy, which is the sum of the electrical energy and surface tension energy. The stored electrical energy of the system is proportional to the applied voltage squared and linearly proportional to the contact area of the droplet with the solid substrate. Furthermore, the electrical energy has a negative sign because the voltage source does *work* in moving charge to and from the dielectric layer. The surface tension energy is directly proportional to the surface area of the liquid-gas interface of the droplet and is positive. Hence, it is favorable for the droplet's substrate contact area to increase when a voltage is applied because it leads to a reduction in total energy (because the electrical energy is negative). This causes the contact angle of the droplet to decrease and increases its wettability with increasing voltage. Note that the change in electrical energy is more than enough to compensate for the increased surface tension area (because the area of the liquid-gas interface has increased). Essentially, the applied voltage acts to modify the surface tension properties of the solid-liquid surface between the dielectric and the droplet.

This electro-wetting effect can be done locally about the three-phase contact line of the droplet (i.e. the line where the liquid, gas, and solid phases intersect). Thus, each electrode can change the surface tension properties immediately above it. This change can be used to move droplets from electrode to electrode, to split droplets (by pulling on either side using three electrodes), to join droplets by making

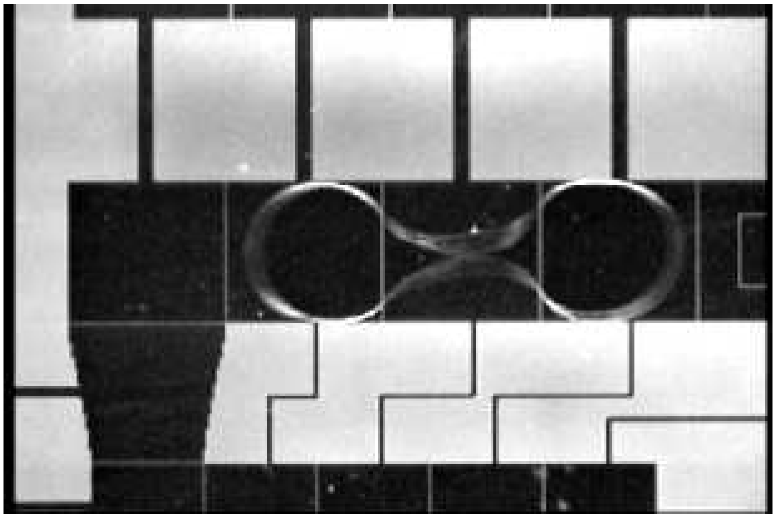


Figure 1.4: Overhead view (through transparent electrode) of experimental EWOD device (courtesy of CJ Kim at UCLA). A droplet is shown undergoing a topological change (i.e. it is pinching). The voltage actuation is 25, 0, and 25 volts (from left to right) for the three square electrodes that the droplet overlays. The long thin pieces are wires that lead to voltage sources.

them collide, and to mix fluid in droplets by making the droplets execute complex paths.

An experimental device with a splitting droplet is shown in Figure 1.4 (the view is through the top transparent electrode). The actuation voltages of the three electrodes from left to right have been turned on to 25, 0, and 25 volts.

In [124], a model was developed for the equilibrium shape of droplets under applied electric fields. This thesis further considers the non-equilibrium fluid dynamics. Specifically, we focus on modeling and simulating motion, splitting, and joining of the liquid droplets, as well as accounting for contact line friction. Before this work, most models and simulation tools have focused on the equilibrium behavior of EWOD, or simplified versions such as 1-D flow dynamics of a plug of fluid driven by EWOD, or a modified EWOD system where contact angle hysteresis is negligible. Furthermore, our simulation tool is able to handle pinching and merging of droplets, which is readily observed in EWOD. And our model has enabled the possibility of controlling fluid flow in an EWOD system such as for controlled mixing or particle control (i.e. for controlling the trajectory of particles or agents floating in a droplet).

The next section gives a general overview of numerical methods for simulating two-phase flow and states why we chose the methods described in this thesis.

1.3 Overview of Numerical Methods for Two-Phase Flow

Given a model of two-phase flow, there is still the problem of discretization and implementing a computational tool to simulate it. Many techniques exist for simulating and computing solutions to partial differential equations (PDE), especially those with a moving boundary. But the foundation of these methods are the standard numerical schemes: Finite Difference (FDM), Finite Element (FEM), and Boundary Integral methods (BIM). FDM [102], [46] works by replacing all derivative terms in the PDE by finite difference approximations (i.e. by approximating the differential operator of the PDE) and partitioning the domain (usually) into a cartesian grid. FEM [18], [20], [76], on the other hand, never discretizes the differential operator. It only approximates the solution of the PDE over an unstructured grid (e.g. a triangulation) that represents the domain. Furthermore, FEM uses an integral representation, which makes it more robust than FDM. BIM [57], [74] also uses an integral representation, but the PDE is captured through the use of a convolution kernel. And BIM only requires a partitioning of the boundary of the domain, which is advantageous in 3-D because no bulk interior mesh is required.

Problems that involve a moving boundary and large deformations require special enhancements to the standard techniques [52] and use implicit and explicit front tracking methods to simulate the motion of the interface (see Figure 1.5 and Table 1.1). One popular implicit method for capturing free surface motion is the level set method [105], [123], which advects a scalar field function whose zero level set represents the interface. The numerical implementation can be done using either FDM or FEM. Level set methods have the advantage of being completely Eulerian and can automatically handle topological changes, though the physics underlying such changes is often left ill-understood. One drawback of the level set method is enforcing boundary conditions because the interface is not known explicitly. This is commonly addressed by including a Dirac-delta type source term into the governing equations that is active at the two-phase boundary [105], [136], [64], but this still has some accuracy issues. In addition, for curvature-driven flows, level set methods typically use an explicit calculation of the interface curvature which can create numerical artifacts and noise [112]. And mass conservation requires special handling [44] or refinement [96]. The phase-field method is also implicit and uses a 'phase' variable to represent the fluid domain [3], [40], [134], [147], where the interface is represented by a thin diffuse region (see Figure 1.5). Phase field methods

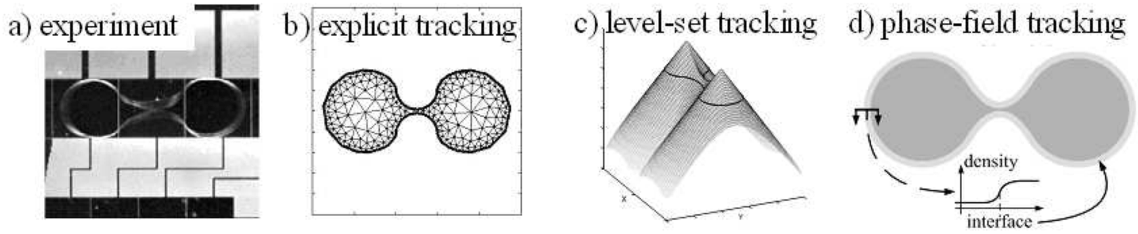


Figure 1.5: Moving interfaces can be tracked by explicit and implicit methods. a) Overhead view of the water/air interface in the UCLA experiment, the droplet is just about to split. b) Explicit tracking of the interface by a FEM over a mesh that conforms to the interface and is convected by the flow. c) Implicit level-set tracking of an interface. The scalar function shown is the signed distance function to the interface; it is convected by the flow. The intersection of the scalar function and the $z = 0$ horizontal plane (here shown as the thick 'figure 8' curve) tracks the moving interface. d) Implicit phase-field tracking of the interface. Here a phase variable, such as the fluid density, varies smoothly from one phase (liquid, dark gray) to the other (gas, white) and is convected by the flow. The interface is the thin smeared gray layer. An inset shows the smoothed density function as it goes from gas to liquid. There is also the Volume of Fluid (VoF) method (not shown) that tracks volume fractions for each phase inside each cell of a computational grid and then reconstructs the interface. And the Coupled Level Set-VoF (CLS-VoF) method combines VoF and level-sets.

have similar advantages and drawbacks as the level set method. Another technique, similar in spirit, is the Volume of Fluid (VoF) method [71], which tracks volume fractions for each phase inside each cell of a computational grid and is designed to conserve mass exactly. However, VoF requires a non-trivial reconstruction algorithm [95] to obtain the interface shape from the mass fraction in each grid cell and also has accuracy concerns for curvature driven flows.

Alternatively, one can use an explicit interface tracking method (Figure 1.5b), which uses a computational grid that conforms to the two-phase boundary (Lagrangian approach) and discretize the PDE using either FDM, FEM, or BIM. FDM is less appealing because the interface mesh will not be cartesian, whereas FEM and BIM make use of unstructured grids naturally and can enforce boundary conditions easily [120], [49] and mass conservation accurately. In addition, there exist finite element and boundary integral methods that take advantage of the intrinsic representation of the interface [9], [43], [74] to compute curvature accurately. However, the main disadvantage to these explicit surface representations is the difficulty in handling large deformations of the mesh and topological changes (pinching/reconnection), especially in 3-D. This requires a non-trivial adaptive mesh algorithm to adjust the mesh and avoid distortion [31], [2], [149], [56], [51], [121], in addition to a way of handling topological changes (droplet splitting and merging) [31], [88]. Some of this difficulty is alleviated with BIM, which requires no bulk interior mesh.

This thesis describes two methods for simulating the free surface motion of a droplet in an EWOD device. The first uses a level set method and finite differences,

Table 1.1: Front-Tracking Method Comparison

	Advantages	*Disadvantages*
Implicit Front-Tracking	topological changes are automatic, can use stationary uniform grid	accuracy issues, conservation of mass
Explicit Front-Tracking	interface position known exactly, high accurate methods available	mesh distortion due to moving grid

and the second method uses an explicit representation of the free surface and a variational formulation with finite elements. Originally, the level set method was chosen for its ability to handle topological changes. But because of the mass conservation and curvature issues, we adopted the second approach. Topological changes for the explicit approach are handled using the method in [140]. The two methods are compared in Chapters 5 and 6.

1.4 Thesis Outline

This thesis is organized as follows. Chapter 2 describes the governing fluid equations and boundary conditions, along with our model of contact angle hysteresis and contact line pinning. Chapter 3 describes our level set method for simulating droplet motion in the EWOD system and its numerical implementation. Chapter 4 describes our variational method, which includes our phenomenological contact line pinning model. We also show the well-posedness of our numerical FEM scheme. Chapter 5 presents numerical results, using both numerical methods, in comparison with experiments that exhibit droplet splitting and bulk droplet motion. In Chapter 6, we describe our algorithm for controlling particle motion in the EWOD system, along with some simulation test cases. Finally, Chapter 7 gives some discussion about the trade-offs of our methods and a list of the contributions of this work. An appendix is also included which contains some supplemental material and a list of symbol definitions (see Appendix D for nomenclature).

Chapter 2

EWOD Modeling

This section describes the EWOD modeling approach. In particular, our main assumptions, derivation of the fluid equations, proper boundary conditions, voltage actuation, contact angle saturation, and hysteresis effects are discussed in detail. A list of the physical parameters for the geometry of the EWOD device, and the fluid parameters for distilled water at standard temperature and pressure (assumed in our model), is given in Table 2.1.

2.1 Governing Equations of the Liquid Flow

In the following sections, the main assumptions and governing equations for the flow of liquid inside an EWOD device are described (see Figure 2.1). In particular, we obtain a model similar to Hele-Shaw type flow with pressure boundary conditions at the liquid-gas interface proportional to its mean curvature.

2.1.1 Navier-Stokes Equations

We start by considering the Knudsen number of the EWOD device, which provides a measure of how accurate the continuum hypothesis is for a fluid system [78]. For our case, we can assume the flow physics to be a continuum because the

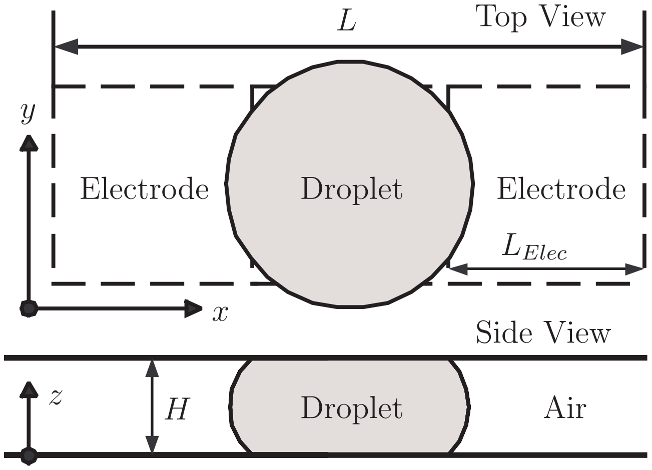


Figure 2.1: EWOD device geometry. The coordinate axes are defined such that the top and bottom plates of the device lie in planes parallel to the x-y plane. The physical parameters of the device are listed in Table 2.1.

Knudsen number is

$$\mathrm{Kn} = \frac{\lambda_{air}}{H} = \frac{6.111 \times 10^{-8}\mathrm{m}}{70 \times 10^{-6}\mathrm{m}} = 8.73 \times 10^{-4}$$

where λ_{air} is the mean free path of air molecules at standard temperature and pressure, and H is the channel height of the device. Clearly, this is within the continuum regime defined to be $\mathrm{Kn} < 10^{-2}$ [78]. The Knudsen number associated with the liquid flow is even smaller because the mean free path of water is much lower than that of air.

Since the flow is a continuum, the dimensional Navier-Stokes equations are applicable. Because we are modeling the flow of water, incompressibility and Newtonian fluid assumptions may be used [109]. This gives

$$\rho(\partial_t \vec{V} + \vec{V} \cdot \nabla \vec{V}) = -\nabla p + \mu \nabla^2 \vec{V} \tag{2.1}$$

$$\nabla \cdot \vec{V} = 0 \tag{2.2}$$

in the bulk liquid, where $\vec{V} = (u, v, w)$ is the three dimensional velocity, p is the pressure, ∂_t denotes the partial derivative with respect to time, and ρ and μ are the density and dynamic viscosity, respectively. Equations (2.2) and (2.1) represent conservation of mass and momentum, respectively, with gravity ignored because the potential energy change in the z direction is negligible when the channel height, H, is small.

Next, we have the boundary conditions for a liquid droplet between two parallel plates. On the top and bottom plates, we have the usual no-slip condition for velocity (i.e. all velocity components are zero). Because the air surrounding the droplet is not being forced, it does not significantly affect any droplet motion. Therefore, by ignoring the airflow, we have the following conditions for the free surface of an incompressible, Newtonian liquid (i.e. the liquid-gas interface) [11]

$$\vec{n} \cdot \mathrm{T}\vec{n} = -\sigma_{lg}(\kappa_1 + \kappa_2) \tag{2.3}$$

$$\vec{t} \cdot \mathrm{T}\vec{n} = 0 \tag{2.4}$$

where σ_{lg} denotes surface tension coefficient, κ_1 and κ_2 are the principal curvatures of the interface [37], T is the stress tensor, $\vec{n}$ is the unit normal vector to the interface, and $\vec{t}$ is any tangent vector to the interface. Physically, (2.3) states that the normal stress across the liquid-gas interface is balanced by surface tension, whereas (2.4)

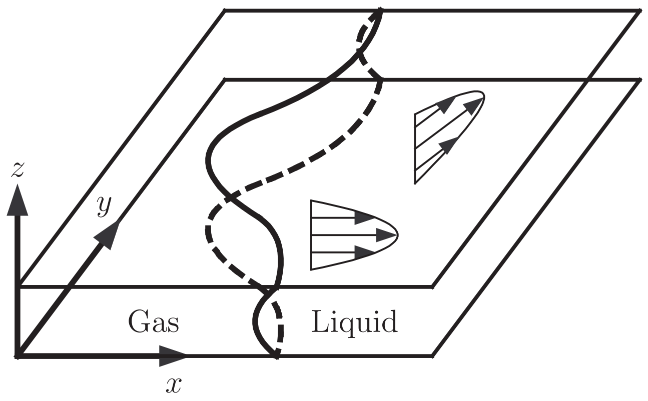


Figure 2.2: Velocity profile: the fluid velocity field is assumed to have a quadratic profile in the z direction.

says the tangential stress vanishes because the shear stress at the liquid-gas interface is negligible.

2.1.2 Hele-Shaw Type Flow

The Reynolds number is small (approximately 1 to 5) because we have pressure-driven flow in a slot with channel height much smaller than the diameter of the droplet [109]. Therefore, we assume the flow can be modeled by a two-dimensional field. By making the additional assumption that the x and y fluid velocity components u and v have a quadratic profile in the z direction (i.e. local Poiseuille flow; see Figure 2.2), equations (2.1) and (2.2) can be non-dimensionalized and reduced to a form similar to Hele-Shaw flow [11]:

$$\nabla^2 p = 0 \tag{2.5}$$

$$\left(\frac{L}{H}\mathrm{Re}\right)\partial_t\vec{u} + 12\left(\frac{L}{H}\right)^2\vec{u} = -\frac{P_0 L}{U_0 \mu}\nabla p \tag{2.6}$$

where $\vec{u} = (u, v)$ is the vector velocity field. The term on the far left of (2.6) is the extra term beyond the usual Hele-Shaw equations. This time derivative term is included because it may have a large magnitude due to rapidly varying pressure boundary conditions if high frequency voltage actuation is used to modulate the droplet's contact angles.

Parameter	**Symbol Definition**
Surface Tension	$\sigma_{lg} = 0.07199$ J/m^2
Dynamic Viscosity	$\mu = 0.89$ g/m $\cdot$ s
Density	$\rho = 996.93$ Kg/m^3
Channel Height	$H = 70\ \mu$m
Electrode Length	$L_{Elec} = 1.4$ mm
Length Scale	$L \approx 3 \times L_{Elec}$
Velocity Scale	U_0 (see Sec. 5.1)
Time Scale	$t_0 = L/U_0$
Pressure Scale	$P_0 = \sigma_{lg}/L$
Reynolds Number	$\mathrm{Re} = \rho U_0 H/\mu$
Capillary Number	$\mathrm{Ca} = \mu U_0/\sigma_{lg}$

Table 2.1: Physical Parameters of the EWOD device for the experiments described in Sections 5.1 and 5.2. See Appendix D for a list of symbols.

The boundary conditions for equation (2.5) are then given by the Young-Laplace relation [11], which says (in non-dimensional form) that the pressure on the liquid-gas interface is equal to the total curvature of the interface (i.e. the sum of the principal curvatures). Because the channel spacing is so small, this can be approximated by

$$p = \kappa_{xy} + \frac{L}{H}\kappa_z, \text{ at the liquid/gas interface} \tag{2.7}$$

where κ_{xy} is the non-dimensional curvature of the droplet in the x-y plane, κ_z is the non-dimensional curvature of a cross-section of the droplet along the z axis (see Figure 2.3), and L is the x-y length scale of the device. Given that (2.5) has been posed in two dimensions, equation (2.7) is evaluated at each point of the boundary of the two-dimensional (2-D) droplet and is discussed in Section 2.2.1.

2.2 Physics of the Droplet Boundary

Above, we described the governing equations of liquid droplet motion. We now discuss the geometry and different physical phenomena happening at the liquid-gas interface, such as voltage actuation, contact angle saturation, and hysteresis. We show how these effects are modeled and how they affect the computation of the boundary conditions.

2.2.1 Interface Curvature

The interface mean curvature is approximated using the individual curvatures κ_{xy} and κ_z in equation (2.7). We compute the z curvature by assuming the interface has a circular cross-section (see Figure 2.3). The x-y curvature computation requires a representation of the shape of the two dimensional droplet boundary. This is accomplished by using either 1) a level set method to implicitly capture the interface (described in more detail in Section 3.4.2), or 2) an explicit representation (described

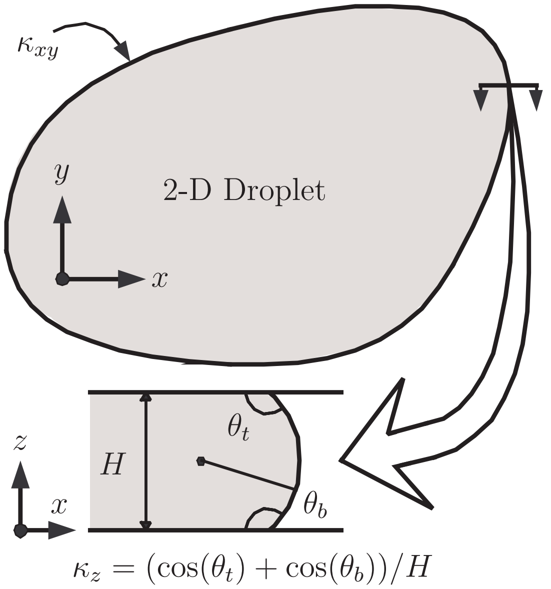


Figure 2.3: Overhead view of a 2-D droplet with side view zoom-in of the interface. The liquid-gas interface is assumed to have a circular cross-section, which gives an estimate of the z curvature, κ_z, in dimensional form. The x-y curvature, κ_{xy}, is just the curvature of the boundary of the two dimensional droplet.

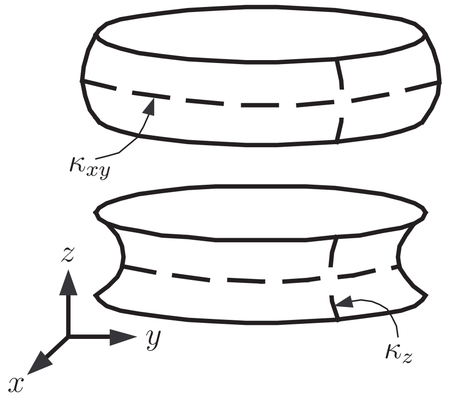


Figure 2.4: Curvature note: κ_{xy} and κ_z are both positive for the bulging droplet on top. For the inward bending droplet, only κ_z is negative.

in Section 4.2.1).

To use the circular approximation for computing the z curvature, we must know the slope of the liquid-gas interface cross-section at the floor and ceiling of the EWOD device. This is given by the top and bottom contact angles, θ_t and θ_b respectively (see Figure 2.3). After some basic geometry, the dimensional z curvature is given by

$$\kappa_z = -\frac{1}{H}[\cos(\theta_t) + \cos(\theta_b)],$$

which gives the non-dimensional curvature as

$$\kappa_z = -[\cos(\theta_t) + \cos(\theta_b)] \tag{2.8}$$

Through the use of EWOD, contact angles can be voltage controlled, thereby allowing the droplet pressure field to be actuated through the boundary condition (2.7). The details of voltage actuation and contact angles are described in the next sections 2.2.2 and 2.2.3.

2.2.2 EWOD Charging Time

We analyze the electrical charging time of the EWOD device to determine if it must be included in our model. Consider the circuit diagram shown in Figure 2.5. Using transfer function theory [104], which reduces the ordinary differential equation associated with the electric circuit to an algebraic problem, we can obtain an estimate for the time constant of the device. The transfer function for this circuit is given by

$$T(s) = \frac{\alpha_1 s + \alpha_2}{\alpha_3 s^2 + \alpha_4 s + \alpha_5} \tag{2.9}$$

where the coefficients are given by

$$\begin{aligned}
\alpha_1 &= (C_T + C_O)R_O R_T \\
\alpha_2 &= R_O + R_T \\
\alpha_3 &= R_L R_O C_O R_T C_T \\
\alpha_4 &= R_O R_T(C_T + 2C_O) + R_L(R_O C_O + R_T C_T) \\
\alpha_5 &= R_O + R_L + 2R_T
\end{aligned}$$

Using the parameters for water, silicon dioxide, Teflon, and the dimensions of the EWOD device [29], equation (2.9) becomes

$$T(s) = \frac{1927.5(s + 3.957 \times 10^{-6})}{(s + 3.132 \times 10^{-6})(s + 2439.7)}$$

After approximately canceling the two near-identical terms in the numerator and denominator, we are left with a transfer function describing a first order differential

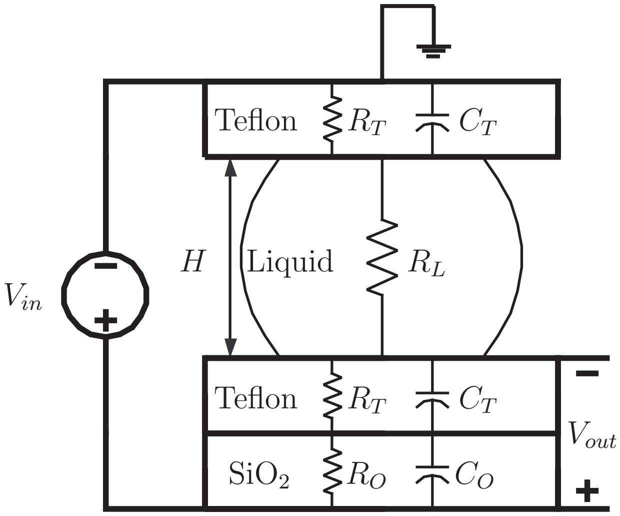


Figure 2.5: EWOD lumped circuit schematic. A voltage source is connected across the device by connecting with the ground electrode (top) and electrode grid (bottom); the electrodes are assumed to be perfect conductors. Each layer of Teflon, and silicon dioxide layer, can be modeled as a parallel resistor and capacitor. The liquid is assumed to only be resistive. Values for the circuit components can be obtained from [91]. The output voltage V_{out} is only taken across the bottom plate of the device because that is where the majority of the electrical energy is stored (due to the silicon dioxide layer).

equation. The defining parameter of any first order system is the time constant, which in this case is 0.41 milliseconds. Using this, the electrical charging time is estimated as four times the time constant, or 1.64 milliseconds.

For the splitting droplet experiment in Section 5.1, the time to split is 0.167 seconds. Because the majority of the voltage drop occurs across the bottom SiO_2 and Teflon layer and the charging time is more than 100 times faster than the *bulk* fluid motion we are interested in (i.e. droplet splitting), we assume the output voltage is instantaneously equal to the input voltage. Therefore, given that there is a direct relation between contact angle and applied voltage (see Section 2.2.3), EWOD is capable of changing the contact angle very quickly.

2.2.3 Contact Angles and Saturation

There is a considerable amount of literature on contact angles and wetting phenomena; see the following references for a sampling: [23], [110], [39], [35], [146], [119], [41], [69], [98]. In this section, we are concerned with how the contact angle varies with respect to the applied voltage.

According to [94], [17], and [36], for a sessile drop on a single dielectric plate, the Young-Lippmann equation predicts a near parabolic curve relating contact angle to the capacitive voltage across the plate (see Figure 2.6). However, if Young-Lippmann is used to simulate droplet splitting, it predicts an incorrect shape for the motion of the droplet. This is because electrowetting, in reality, deviates from the Young-Lippmann theory at high voltages and reaches a saturation limit (also shown in Figure 2.6). In Section 5.1.1, we present simulations using the Young-Lippmann equation and saturation to illustrate the importance of modeling the latter. For more information on the causes of contact angle saturation of sessile droplets, see [124].

The available literature only discusses a sessile droplet on a single plate. For this thesis, we need data on contact angle variations of a droplet sandwiched between two plates. In [29], experimental contact angle data for the EWOD device is given at an applied voltage of 0 and 25 volts. The top contact angle remains approximately the same at 117 degrees regardless of voltage actuation. This is because most of the dielectric energy is stored in the bottom plate due to the presence of the SiO_2 layer. Therefore, we assume the nominal contact angle on the top plate is fixed at 117 degrees. The bottom contact angle varies between 117 and 90 degrees at 0 and 25 volts respectively.

In order to model contact angle variations on the bottom plate for any voltage, we must combine the two experimental data points for the parallel plate case with the six data points for the single plate case (shown in Figure 2.6). In other words, we must transform the contact angle versus voltage data for the single plate case into useable data for the parallel plate case. Because there is an overlap between two of the data points for both cases, we define a linear mapping that represents this correspondence

$$\theta_2 = \frac{117^\circ - 90^\circ}{119^\circ - 80.4^\circ}(\theta_1 - 80.4^\circ) + 90^\circ \tag{2.10}$$

where θ_1 is the contact angle for a single plate and θ_2 is the contact angle for two parallel plates. This equation maps 119.0° to 117.0° and 80.4° to 90.0°. By taking the six data points and their piecewise interpolating polynomial for saturation on a single plate as input to this linear map, we obtain the saturation curve for two plates as output (see Figure 2.6). Due to the scarcity of data on contact angle variation for the parallel plate EWOD device, we assume the two plate saturation curve in Figure 2.6 is true for our model.

2.2.4 Hysteresis

Contact angle hysteresis is another piece of the boundary physics we need to complete our model of droplet motion using EWOD forces. Hysteresis refers to the difference in contact angles between the advancing and receding ends of sessile drops. It is a direct consequence of contact line pinning, which acts as a force that resists any sliding motion, and it can be seen when water droplets stick to the side of a solid surface (see Figure 2.7). For more information on contact angle hysteresis and line pinning, see [23], [39], [35], [146], [119], [41], [69].

From Figure 2.7, for a sessile drop on a single plate, it can be seen that the advancing and receding contact angles are greater and smaller, respectively, than the nominal contact angle. This is also true for a droplet inside the EWOD device (shown in Figure 2.8). Ideally, if there were no hysteresis, the nominal contact angle at the interface of the droplet should be determined by the two plate saturation curve in Figure 2.6 and the applied voltage at the interface. But in the presence of hysteresis, the contact angles deflect from their nominal values which affects the

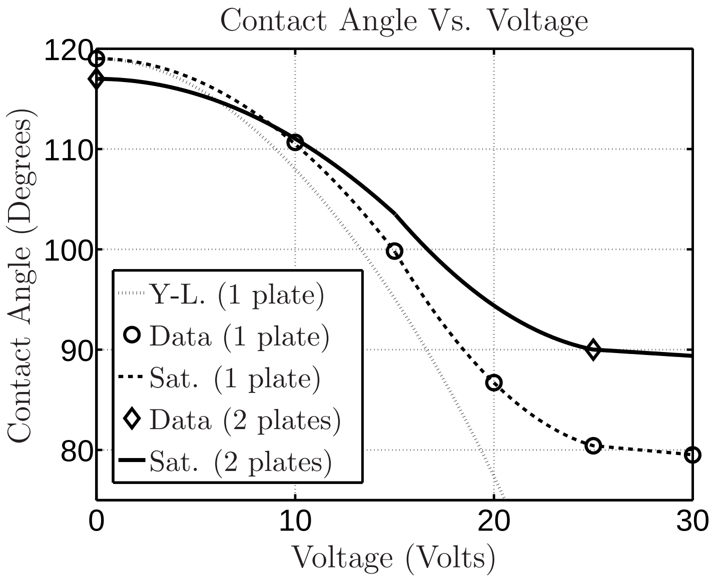


Figure 2.6: Contact Angle Versus Voltage Curves: theoretical and experimental data for contact angle variations using electrowetting on dielectric. The dotted line denoting the Young-Lippmann (Y-L) curve is theoretical [94], [17], [36]. The single plate saturation curve has six experimental data points (given in [29]) with a piecewise interpolating polynomial (see dashed line and '○' data points). The two plate saturation curve has two experimental data points [29] with an interpolating curve derived from the single plate case in Section 2.2.3 using a linear map (see solid line and '◇' data points). This curve is used in our simulation to predict the correct droplet motion and splitting time (see Section 5.1.3).

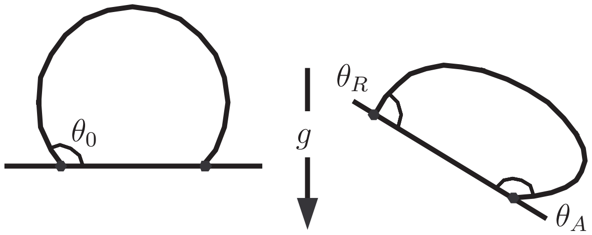


Figure 2.7: Illustration of contact line pinning and hysteresis. The contact line of the droplet is where the liquid-gas interface meets the solid surface. Line pinning simply means the contact line (and the droplet) is stuck to the surface. A direct result of this is contact angle hysteresis, which refers to the situation where the receding angle θ_R is less than the nominal (equilibrium) angle θ_0 while the advancing angle θ_A is greater than θ_0. In the diagram above, θ_0 is the contact angle of the droplet on a horizontal surface, whereas θ_R and θ_A are the contact angles when the surface is tilted. The droplet can slide by using a large enough tilt angle, but the motion will be limited by the static frictional effect of line pinning and contact angle hysteresis will still be present. A similar situation happens in EWOD, where hysteresis also acts as a retarding effect by deforming the liquid-gas interface shape in an unfavorable way (see Section 2.2.4).

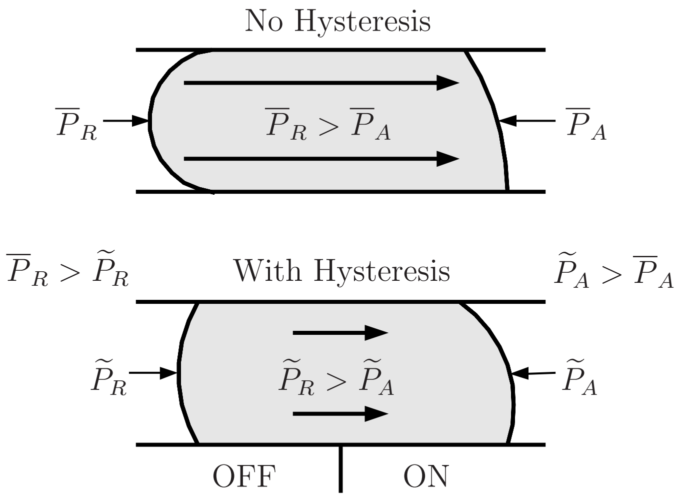

No Hysteresis
$\overline{P}_R$
$\overline{P}_R > \overline{P}_A$
$\overline{P}_A$
$\overline{P}_R > \widetilde{P}_R$
With Hysteresis
$\widetilde{P}_A > \overline{P}_A$
$\widetilde{P}_R$
$\widetilde{P}_R > \widetilde{P}_A$
$\widetilde{P}_A$
OFF
ON

Figure 2.8: Effects of contact angle hysteresis in the EWOD device. A droplet is shown moving from left to right due to voltage actuation (OFF/ON). When hysteresis is present, the contact angles differ from their nominal (non-hysteresis) values. The effects on the liquid-gas interface pressure are also shown. The presence of hysteresis causes the pressure gradient throughout the droplet to be weakened from the nominal case ($\overline{P}_R - \overline{P}_A > \widetilde{P}_R - \widetilde{P}_A$).

pressure on the boundary by the Young-Laplace relation (2.7).

To see how it is affected, consider a *circular* droplet in motion due to voltage actuation (see Figure 2.8). Let $\overline{P}_A$ and $\overline{P}_R$ denote the pressures at the advancing and receding ends of the droplet, respectively, when no hysteresis is present. And let $\widetilde{P}_A$ and $\widetilde{P}_R$ denote the same pressures with hysteresis. It is clear from Figure 2.8 that the z curvatures at the receding and advancing ends of the droplet are larger and smaller, respectively, for no hysteresis than with hysteresis. From equation (2.7), it can be seen that the curvature change implies that $\overline{P}_R > \widetilde{P}_R$ and $\overline{P}_A < \widetilde{P}_A$.

This change in boundary pressure weakens the pressure gradient throughout the droplet from what it would be without hysteresis because its magnitude is proportional to the pressure difference

$$|\nabla \overline{P}| \propto |\overline{P}_R - \overline{P}_A|$$

$$|\nabla \widetilde{P}| \propto |\widetilde{P}_R - \widetilde{P}_A|$$

where $\nabla \widetilde{P}$ and $\nabla \overline{P}$ are the pressure gradients with and without hysteresis, respectively. Using the fact that $\overline{P}_R > \widetilde{P}_R > \widetilde{P}_A > \overline{P}_A$, we obtain the inequality

$$|\nabla \widetilde{P}| < |\nabla \overline{P}|.$$

Hence, the driving force of the droplet motion is decreased when hysteresis is present. This is why our simulation (with just saturation modeled) predicts a split time over ten times faster than the experiment shows (see Section 5.1.2).

From the discussion above, an obvious way to model hysteresis is to modify the contact angle of the interface based on which way it is moving. However, from our own numerical experiments with the level set method, this is not very robust. Therefore, we opted for a simpler model by assuming that

$$\nabla \widetilde{P} = K_{hys} \nabla \overline{P}$$

where K_{hys} is a constant smaller than 1. In other words, we scale down the pressure gradient in (2.6) to account for hysteresis. This is analogous to the contact line friction model in [36] and [143], which also acts as a retarding effect on liquid motion.

Scaling the pressure gradient introduces one fitting parameter into the simulation, but is straightforward and capable of approximately capturing the droplet motion and time scale observed in the experiments (see Section 5.1.3). However, we do stress that it is not exact nor does it capture the effect of line pinning, which is observable in EWOD as demonstrated in [29] by the fact that droplets do not move unless sufficient voltage actuation is used. Contact angle hysteresis is not completely understood, so we opted for a model that is computationally quick but still captures the lossy effect of droplet motion that is induced by line pinning; namely hysteresis. In Section 2.3, we describe an alteration of the EWOD pressure boundary conditions that is able to account for contact line pinning.

We now estimate the hysteresis constant in terms of contact angles. From the relations given above, we have

$$K_{hys} = \frac{|\widetilde{P}_R - \widetilde{P}_A|}{|\overline{P}_R - \overline{P}_A|}$$

where the pressure terms are given by (2.7) and (2.8). Using these, K_{hys} can be estimated by

$$\begin{aligned} G_{num} = \Big[& \cos(\theta_{t,0V} - \Delta_{hys}) + \cos(\theta_{b,0V} - \Delta_{hys}) \\ & - \cos(\theta_{t,25V} + \Delta_{hys}) - \cos(\theta_{b,25V} + \Delta_{hys}) \Big] \\ G_{den} = \Big[& \cos(\theta_{t,0V}) + \cos(\theta_{b,0V}) \\ & - \cos(\theta_{t,25V}) - \cos(\theta_{b,25V}) \Big] \\ K_{hys} = & \frac{G_{num}}{G_{den}} \end{aligned} \tag{2.11}$$

where Δ_{hys} is the extra amount of contact angle deflection from the nominal angle due to hysteresis, and θ_t and θ_b are the top and bottom contact angles, respectively. The voltage subscripts specify the actuation strength of the contact angles, with the top angle fixed at 117.0° regardless of voltage and the bottom angle obeys the two plate saturation curve in Figure 2.6. Due to the lack of data on hysteresis of EWOD driven droplets, we assume the contact angle deflection to be the same on the top and bottom of the advancing and receding ends of the droplet. In Section 5.1.3, we use (2.11) to estimate the hysteresis angle deflection that corresponds to the appropriate constant, K_{hys}, that ensures the simulated splitting time matches the experiment.

2.3 Alternate Contact Line Pinning Model

The model described above does not accurately account for contact line pinning (see Section 2.2.4). Therefore, an alternative model is described in the following section and is based upon the assumption that contact line pinning can be modeled in an analogous way to static (Coulomb) friction.

2.3.1 Phenomenological Approach

Contact line pinning (or sticking) is a readily observed phenomenon in most wetting applications (see [23], [39], [35], [146], [119], [41], [69]). It is not a fluid viscous effect but rather a kind of molecular adhesion that occurs at the three-phase contact line of the droplet. This can prevent motion of a droplet, even under an applied force (see Figure 2.7) and is observable in the EWOD system. As of today, the nature of this effect is still somewhat controversial. Most modeling of contact line pinning uses molecular dynamics simulations [137], [63], which can be computationally very expensive.

But for our purposes, a molecular dynamics simulation is not necessary or desirable. Thus, to improve the agreement between simulation and experiment, we use a phenomenological approach to incorporate this effect into our continuum model. This is done to avoid a molecular/atomistic description that would be impractical for enabling useful simulations of fluid flow in an EWOD device. Recently [16], some new models for contact line dynamics have been proposed that avoid an atomistic description and are designed to be incorporated into a continuum model.

Macroscopic experiments [146], [103] indicate that the line pinning force scales with the length of the contact line (see Figure 2.9). In other words,

$$F_{max} = c_{pin} L_{cl} \tag{2.12}$$

where F_{max} is the maximum total force that can resist motion of the droplet, L_{cl}

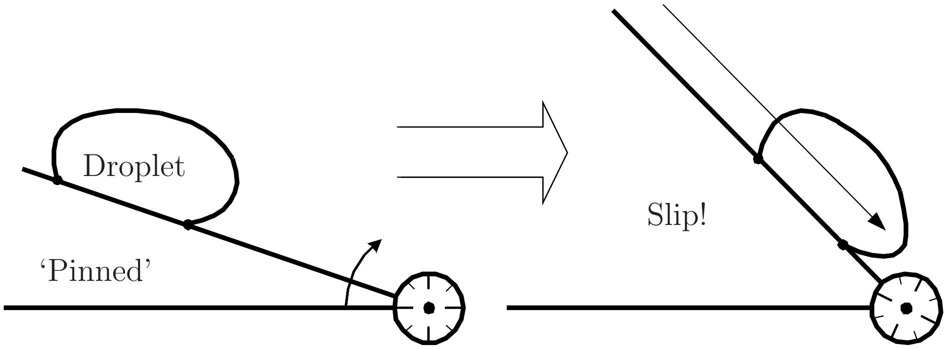


Figure 2.9: Diagram of simple contact line pinning experiment. An adjustable inclined plane is shown with a droplet resting on it. On the left, the droplet slightly deflects towards the right because of gravity but does not move because of the contact line pinning effect. When the angle of the inclined plane is increased to a high enough value, the force due to gravity is strong enough to overcome the pinning effect, and the droplet slides down the plane. Given the angle at which the droplet first slips, one can compute the maximal pinning force given the droplet volume, density, and gravitational force. After performing this experiment several times for droplets of varying volumes, it is found that F_{max} scales linearly with the contact line length L_{cl} [146], [103]. This relation is written as $F_{max} = c_{pin} L_{cl}$, where c_{pin} is the line pinning coefficient with the same units as surface tension.

is the total contact line length, and c_{pin} is the line pinning coefficient with units of force per length (units of surface tension). So the constant c_{pin} represents the maximum force per unit length that a piece of contact line can exert against the droplet to prevent its motion. Therefore, we model contact line pinning similarly to static (or Coulombic) friction, in which case the friction force always opposes motion and cannot exceed a certain threshold value (see Figure 2.11).

2.3.2 Including Line Pinning into the Governing Equations

Since the EWOD governing equations are posed in 2-D, we must 'average' the above line friction model in order to incorporate it into the pressure boundary condition. This is done by averaging the maximal line friction coefficient c_{pin} over the channel height H of the EWOD device (see Figure 2.10). This gives a maximal 'pinning pressure' $\widetilde{P_{pin}} = 2c_{pin}/H$ (in dimensional form), which represents the maximum opposing force per liquid-gas interface area that the contact line can apply *against* motion of the interface. The factor of '2' accounts for the interface contact line pinning at the floor and ceiling of the EWOD device. The non-dimensional pinning pressure is given by

$$P_{pin} = \frac{1}{P_0}\frac{2c_{pin}}{H}, \tag{2.13}$$

where P_0 is the reference pressure scale. This allows us to introduce a variable pinning pressure λ to the boundary condition (in non-dimensional form)

$$p = \kappa_{xy} + \frac{L}{H}\kappa_z + \lambda, \tag{2.14}$$

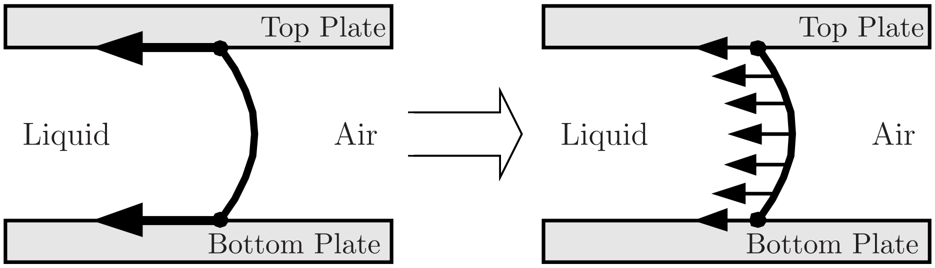


Figure 2.10: Contact line force averaging. Contact line friction is a force that is active along the three-phase contact line. On the left, the contact line pinning force is shown concentrated at the contact line (at both the floor and ceiling). On the right, the pinning force has been redistributed over the channel height. Since the governing EWOD fluid equations have been averaged along the channel height, we average the contact line friction force across the channel height. This redistributes the force from a length of contact line (at floor and ceiling), to a vertical strip along the liquid-gas interface. This allows the line friction to be included in the pressure boundary conditions as an additional pressure term (see equation (2.14)).

where $\lambda = P_{pin} \operatorname{sgn}(\vec{u} \cdot \vec{n})$ (see Figure 2.11). In other words, if the normal velocity of the liquid-gas interface is positive, then the pinning pressure will push back with maximum positive pressure $+P_{pin}$ to limit the motion. Likewise, if the normal velocity is negative, the pinning pressure will push back in the opposite direction $-P_{pin}$. And if the normal velocity is zero, then λ takes on a value between $\pm P_{pin}$ and acts as a lagrange multiplier to enforce the constraint that the interface does not move (see Figure 2.12). This also means that $|\lambda| \leq P_{pin}$, which agrees with the above experimental observation.

It is important to note that this line pinning model must be implemented exactly in order to correctly capture the pinning effect. Replacing the 'sign' function by a smoothed version, and plugging it into the pressure boundary condition gives poor results and does not simulate pinning. This model also introduces a coupling between the velocity and pressure boundary condition, which makes it non-linear. Despite this, we are able to include this model into our variational formulation of Chapter 4, and we have a method of solution. Unfortunately, it is not clear how to include this model into our level set method. Hence, it is not used in Chapter 3.

2.4 Final Equation Summary

This section collects all the equations and simplifies some of the notation. The first section gives the model without any contact line pinning, and the next section indicates how the model changes when the contact line pinning model of Section 2.3 is included.

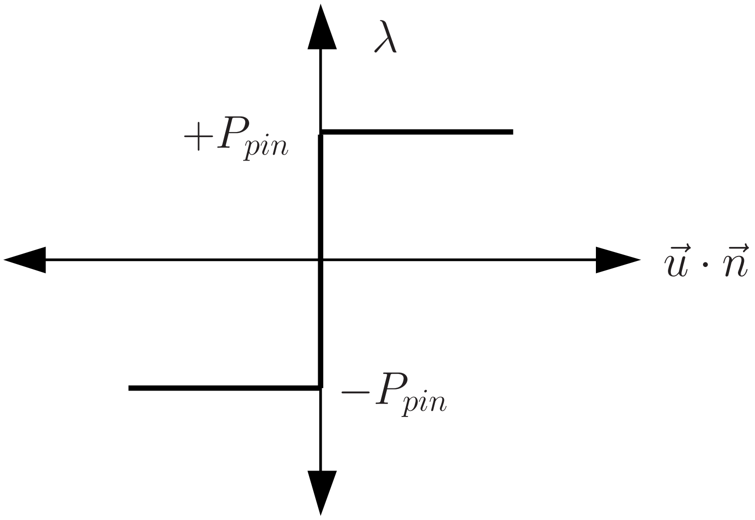


Figure 2.11: Line pinning variable versus normal velocity. The line pinning friction variable λ is defined by the 'sign' function of the normal velocity: $\lambda = P_{pin}\,\mathrm{sgn}(\vec{u}\cdot\vec{n})$. If the normal velocity ($\vec{u}\cdot\vec{n}$) is positive, the friction 'pressure' pushes against the interface with maximum $+P_{pin}$; vice-versa when $\vec{u}\cdot\vec{n}$ is negative (note that $\vec{n}$ is an outward pointing normal vector). When $\vec{u}\cdot\vec{n} = 0$, the pinning variable takes on a value between $\pm P_{pin}$ (i.e. $-P_{pin} \leq \lambda \leq +P_{pin}$) and acts as a lagrange multiplier to enforce the constraint that the interface does not move. Also see Figure 2.12.

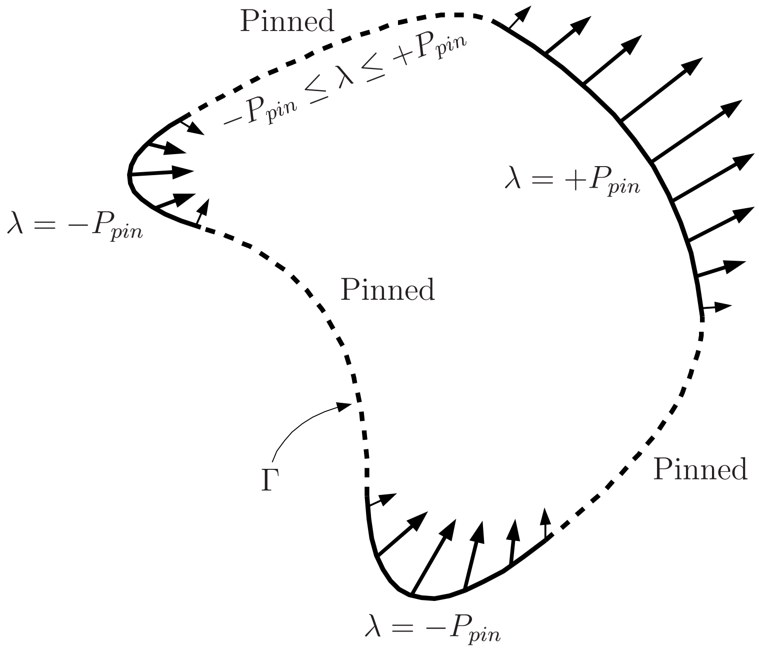


Figure 2.12: A 2-D droplet with parts of the boundary pinned. The pinned regions are denoted by a dashed line; unpinned regions are shown as a solid line with velocity arrows indicating direction of motion. An outward motion is considered positive ($\vec{u} \cdot \vec{n} > 0$), and an inward motion is negative ($\vec{u} \cdot \vec{n} < 0$). The pinning variable λ is defined on the boundary Γ of the droplet. On the unpinned regions, the value of λ saturates to $\pm P_{pin}$. On the pinned regions ($\vec{u} \cdot \vec{n} = 0$), λ continuously varies between $-P_{pin}$ and $+P_{pin}$. In our simulations (see Chapter 5), λ is used to indicate where the boundary is pinned, i.e. the boundary is pinned wherever $|\lambda| < P_{pin}$.

2.4.1 Model Without Contact Line Pinning

We now write the final model equations (non-dimensionalized) describing the fluid flow of a liquid droplet inside an EWOD device. The equations for the pressure field are

$$\nabla^2 p = 0, \qquad \text{in } \Omega \tag{2.15}$$

$$p = \kappa + E, \text{ on } \Gamma \tag{2.16}$$

where $\nabla^2 := \nabla \cdot \nabla$ is the Laplacian operator, Ω denotes the domain of the liquid droplet in two dimensions with boundary labeled Γ (see Figure 2.13), p is the pressure, and κ is the curvature in the x-y plane (note: we have dropped the subscript xy for convenience). The symbol E is the electrowetting forcing

$$E = \frac{L}{H}\kappa_z, \tag{2.17}$$

where L is a chosen length scale, H is the channel height, and κ_z is the z curvature given by

$$\kappa_z = -[\cos(\theta_t) + \cos(\theta_b)] \tag{2.18}$$

where θ_t and θ_b are the contact angles on the top and bottom of the EWOD device, respectively. The top angle is assumed to be 117.0° regardless of the applied voltage. The variations of the bottom angle are given by the two plate saturation curve in Figure 2.6.

The equation for the velocity field is

$$\alpha \partial_t \vec{u} + \beta \vec{u} = -\nabla p, \text{ in } \Omega \tag{2.19}$$

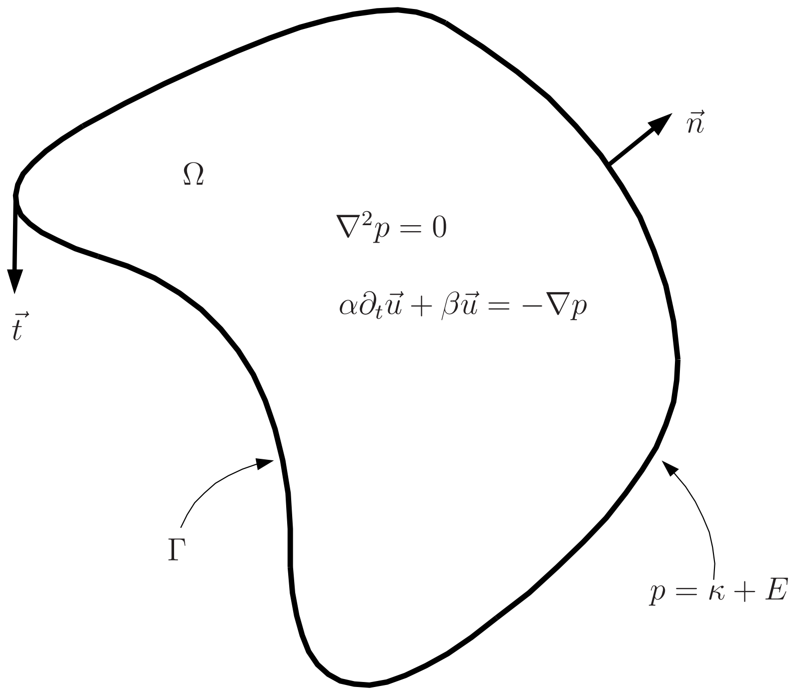


Figure 2.13: Diagram of 2-D droplet domain with notation. The domain of the droplet is denoted by Ω and its liquid-gas interface is labeled Γ. The unit vectors $\vec{n}$ and $\vec{t}$ denote the outward pointing normal vector and 'right-hand' oriented tangent vector. Inside the droplet, the pressure field obeys Laplace's equation, and the velocity field is governed by a Hele-Shaw type equation. The pressure boundary condition depends on the curvature of Γ (denoted κ) and the EWOD forcing E.

where $\vec{u} = (u, v)$ is the vector velocity with x and y components denoted by u and v. The constants α and β are given by

$$\alpha = \left(\frac{\rho U_0 L}{\mu}\right)\left(\frac{\mathrm{Ca}}{K_{hys}}\right) \qquad\qquad \beta = 12\left(\frac{L}{H}\right)^2\left(\frac{\mathrm{Ca}}{K_{hys}}\right) \tag{2.20}$$

where U_0 is a chosen velocity scale, ρ is the fluid density, μ is the dynamic viscosity, K_{hys} is the hysteresis constant, and Ca is the capillary number.

Because we have two-phase flow, we need an equation to describe the motion of the droplet boundary Γ. For each point $\vec{x}$ on the boundary, the change in position is given by

$$\partial_t \vec{x} = (\vec{u}(\vec{x}, t) \cdot \vec{n}(\vec{x}, t))\vec{n}(\vec{x}, t) \tag{2.21}$$

where $\vec{n}$ is the unit outer normal vector of the boundary. Basically, the droplet boundary moves with the normal velocity of the fluid. The next two chapters discuss the numerical simulation of these equations.

2.4.2 Model With Contact Line Pinning

The governing equations inside the domain Ω are the same as in the previous section, except the pressure boundary condition is modified by the following relation

$$p = \kappa + E + \lambda, \tag{2.22}$$

where λ is a pinning variable and is defined by

$$\lambda = P_{pin}\,\mathrm{sgn}(\vec{u} \cdot \vec{n}). \tag{2.23}$$

The λ encodes the contact line pinning effect and is able to partially account for the loss of EWOD forcing available to move or split a droplet. The other loss mechanism is due to contact angle hysteresis when the droplet is in motion and is accounted for by the hysteresis constant K_{hys}. Chapter 4 discusses the simulation of EWOD fluid dynamics with this pinning effect included.

Chapter 3

Implicit Front Tracking with The Level Set Method

In this chapter, we describe our level set implementation of the EWOD governing equations. This was the first method we experimented with and does not include the contact line pinning model. In chapters 5 and 6 we show comparisons between our level set method and the variational formulation given in Chapter 4.

3.1 Introduction

The most crucial part of simulating the fluid equations in Section 2.4 is in handling the time-varying two-phase droplet boundary modeled by (2.21). Various methods for simulating two-phase flow are available (see Section 1.3 for a survey). For this thesis, the method needs to be capable of simulating splitting and joining of droplets without excessive computation. Therefore, we first chose the level set method [105], [123], [24], which implicitly tracks the droplet boundary as the zero level set of a scalar function defined over the x-y plane (see Figure 3.3). This scalar function deforms and changes shape based on the fluid velocity field, which causes the zero level set to also change. Hence, the motion of the droplet boundary is captured through the evolution of the scalar function.

This level set function is used to define the domain of the droplet at each instant of time, allowing the pressure and velocity fields to be computed from finite

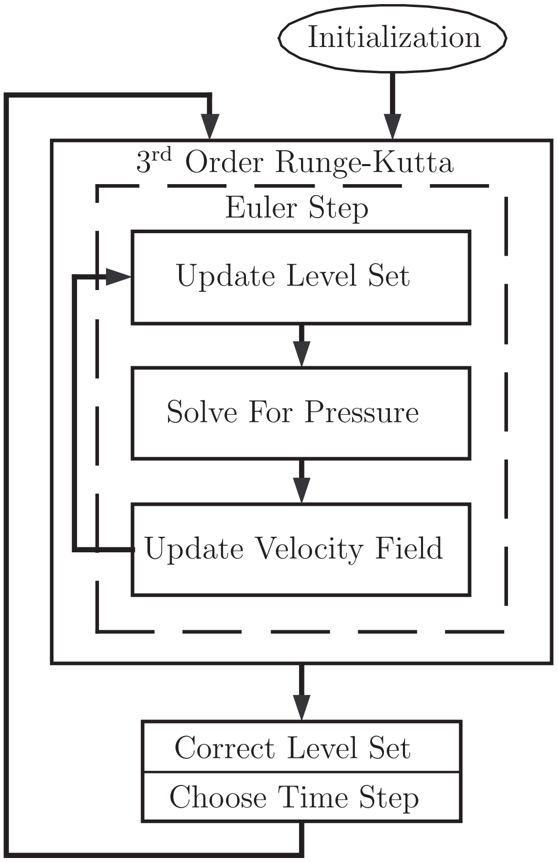


Figure 3.1: Algorithm flowchart.

difference approximations to equations (2.15) and (2.19). We combine these methods in a third order Runge-Kutta time-stepping algorithm that specifies an order to the computation of the pressure field, velocity field, and level set update (see Figure 3.1). The following sections give the details of our algorithm, which is based on the methods used in [50] and [24].

All simulations were performed with MATLAB on a Pentium 4, 3.6 GHz with 2 GB of RAM running Windows XP. The computing time of each simulation varied between 3 and 6 minutes for a 108x108 mesh, which shows the speed of the method.

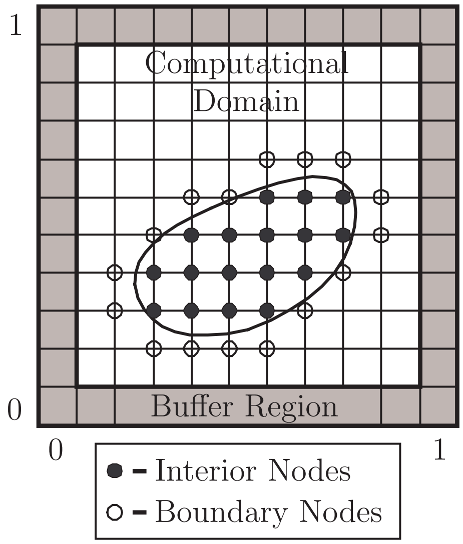


Figure 3.2: Computational domain layout. Liquid region corresponds to interior nodes. Boundary nodes are adjacent to the interior nodes.

3.2 Discretization

The computational domain is defined to be the unit square and is discretized by a regular cartesian grid (see Figure 3.2). For the simulations given in Chapter 5, we used a 108x108 mesh. On this grid, the level set function, ϕ, and the fluid variables $\vec{u}$ and p are sampled. A small buffer region, three grid nodes thick, is defined at the sides of the computational domain. No droplet motion is allowed inside the buffer region because of potential problems with computing second order spatial derivatives there.

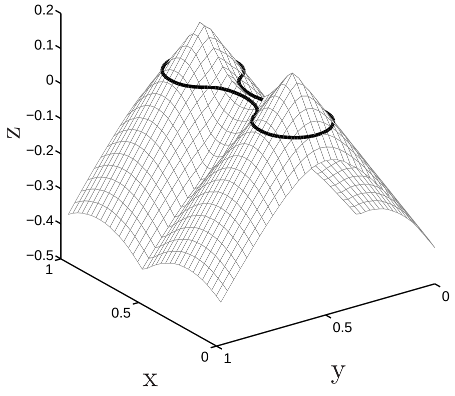


Figure 3.3: Example of a level set function. The zero level contour is highlighted with a thick black line and shows a droplet about to split.

3.3 Initialization

The level set function, ϕ, is initialized to a signed distance function with the zero level contour corresponding to the initial interface shape (see Figure 3.3). By distance function, we mean that the value of ϕ at a grid point in the computational domain corresponds to the shortest distance that the grid point is from the interface. Signed distance means that ϕ is positive inside the droplet and negative outside. Next, the velocity field, $\vec{u} = (u, v)$, is set to zero. And finally, we choose a small initial time step before entering the main update routine discussed in Section 3.4.

3.4 Main Update Routine

At each time step of our simulation, the fluid variables and level set function are updated by computing a convex combination of three forward Euler steps. This method is a third order Runge-Kutta method, and is detailed in [105] and [112].

In each Euler step, the level set is updated first, followed by the pressure, and then velocity. The updated level set is used in computing the pressure field for the new time step, which is then used to update the velocity field (see Figure 3.1). In the following sections, we give the details of each of these subroutines.

3.4.1 Update Level Set

The level set function is updated through a standard convection equation:

$$\partial_t \phi + \vec{u} \cdot \nabla \phi = 0 \tag{3.1}$$

This equation represents *conservation* of the quantity, ϕ, while being transported by the velocity field, $\vec{u}$. Hence, the zero level set 'quantity,' $\phi = 0$, is transported by the local fluid velocity around the droplet boundary. The numerical implementation uses an upwinded third order Hamilton-Jacobi weighted essentially non-oscillatory (WENO) method for discretizing the velocity-gradient term, which uses $\vec{u}$, and ϕ from the previous time step. This method is robust and is described in detail in [105].

In this thesis, to reduce simulation time, only the grid nodes in a small band surrounding the zero level set are actually updated. This does not reduce accuracy because the whole level set must be reset to a distance function periodically (see Section 3.5.1).

3.4.2 Solve For Pressure

The domain, Ω, of the droplet is defined to be the regions in the x-y plane where the level set function, ϕ, is positive (see Figure 3.3). The computational domain of a hypothetical droplet is depicted in Figure 3.2. Each of the grid nodes is located on an electrode with a known applied voltage. The local curvature of the boundary, Γ, is then given by [105]:

$$\kappa = -\frac{\phi_x^2\phi_{yy} - 2\phi_x\phi_y\phi_{xy} + \phi_y^2\phi_{xx}}{(\phi_x^2 + \phi_y^2)^{3/2}}\Bigg|_{\phi=0} \tag{3.2}$$

where the level set derivative terms are approximated using central differences. Because of fundamental problems with differentiating numerical data, the level set function, ϕ, must be filtered prior to computing the derivative terms [112]. In addition, κ must be post-filtered to ensure smooth curvature data. This is mainly due to the explicit nature of the curvature calculation used here.

Then, we get the bottom contact angle, θ_b, at each boundary node using the known voltage there and the two plate saturation curve in Figure 2.6. Voltage transitions near the edge between two electrodes are smoothed out using linear interpolation in a narrow region to prevent large localized velocities caused by discontinuous boundary conditions. Finally, the boundary pressure values are computed using equations (2.16), (2.17), (2.18), and (3.2) evaluated on the boundary nodes.

The pressure values at the interior nodes are computed by solving (2.15), which implicitly contains the conservation of mass equation (2.2). The numerical solution is obtained by using a simple red-black Gauss-Seidel iterative solver with a relative error tolerance of 10^{-8} [46]. Note that Gauss-Seidel is easily implemented on a regular cartesian grid. Other more advanced methods for solving a matrix system of equations exist but would require the matrix structure to be recreated at every time step because the domain of the droplet is always changing. In addition, the Gauss-Seidel solver is implemented in C, for speed, and called from MATLAB. Therefore, we saw no significant advantage with using a different method.

Once the pressure values are known, the pressure gradient, ∇p, at every interior node is computed using a central difference formula [46]. These values are then used in the velocity update routine.

3.4.3 Update Velocity Field

The fluid velocity, $\vec{u}$, obeys a first order time differential equation given by the vector equation (2.19). The pressure gradient provides a forcing term in the equations, which causes a velocity field to develop. We compute the velocities on our computational domain by discretizing (2.19) in space while keeping time continuous. This approach is commonly known as a semi-discrete method [130] (or method-of-lines) and allows for the use of an analytic solution to (2.19) for updating the velocity field.

For a time-invariant pressure gradient, the steady-state solution for equation (2.19) is given by

$$\vec{u}_{ss} = -\frac{\nabla p}{\beta}$$

Let $\vec{u}_i$ be the initial velocity at time t_i. Then, by assuming the pressure gradient remains constant during the time step, $\triangle t$, the velocity field at $t_i + \triangle t$ is given by:

$$\vec{u}|_{t_i+\triangle t} = e^{-\triangle t \frac{\beta}{\alpha}} \vec{u}_i + [1 - e^{-\triangle t \frac{\beta}{\alpha}}] \vec{u}_{ss} \tag{3.3}$$

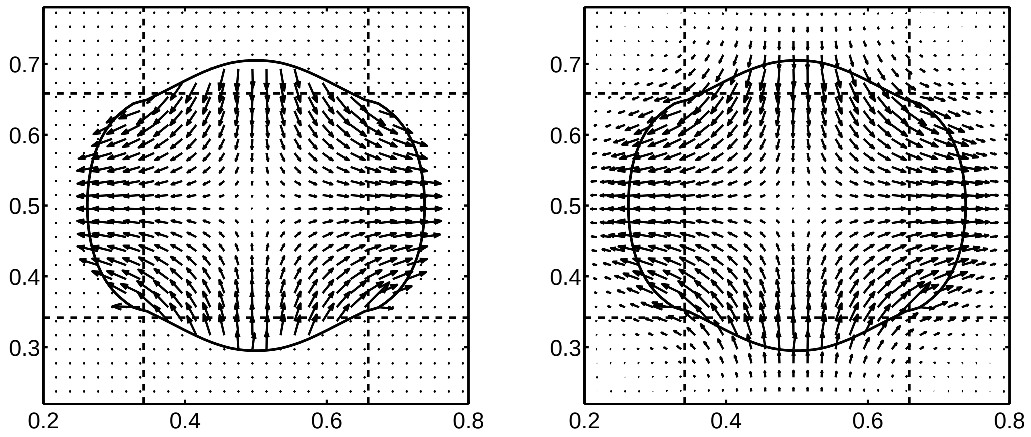
where the above equation is the analytic solution for the vector velocity in (2.19) evaluated at $t_i + \triangle t$. We apply this update to all interior grid nodes to obtain the velocity field inside the droplet for the current time step.

The last piece needed for updating the velocity is to extend it from inside the droplet to outside. In Section 3.4.1, the velocity field is needed to update the level set function. But in order to do this properly, it must be extended into the boundary and exterior nodes of the computational domain. This is accomplished by letting the velocity components diffuse into the exterior region (see Figure 3.4), which ensures a continuous velocity field for updating the level set. We now give the details of this extension algorithm.

First, fix the values of $\vec{u}_{ss}$ at the interior grid nodes and set the edge node values of the computational domain to be zero. At each boundary and exterior grid node (excluding the edge nodes), compute a value for $\vec{u}_{ss}$ using this formula:

$$\vec{u}_{ss}|_{i,j} = \frac{\vec{u}_{ss}|_{i+1,j} + \vec{u}_{ss}|_{i-1,j} + \vec{u}_{ss}|_{i,j+1} + \vec{u}_{ss}|_{i,j-1}}{4}$$

where (i, j) are the row and column coordinates for each grid node. Iterate this process a fixed number of times for all boundary and exterior grid nodes. With each iteration, the velocity values diffuse further away from the interior region. For our simulations, we used 50 iterations to extend $\vec{u}_{ss}$, which is then used in (3.3). This guarantees that the velocity field will always be extended. Our method is nothing more than Jacobi iterations executing on all boundary and exterior nodes and acting



(a) Without Extension

(b) With Extension

Figure 3.4: Velocity Field Extension. Illustration showing before and after results of extending the velocity field by diffusion using Jacobi iterations.

on velocity values. In [105] and [24], the method they use for extending the velocity field is based on a convection-type equation that propagates velocity data from the interior region into the boundary and exterior nodes. However, we do not use their method because it is computationally more complex. Another technique for generating velocity fields that also preserves the distance function property of the level set function is given in [123], but this is a very expensive computation. We prefer our method because it is simpler and gives excellent performance.

3.5 Remaining Tasks

After updating the level set function, pressure, and velocity fields, there remain two final tasks. Reconditioning the level set function and choosing the next time step. Once completed, the program loops back to Section 3.4 to continue the simulation.

3.5.1 Correct Level Set

Despite the ingenuity of the level set method, it does have problems. Since we are using the fluid velocity to update ϕ, it is highly likely that the level set will become distorted and introduce numerical inaccuracies [123]. This requires periodically resetting ϕ so that it is always close to being a distance function. This is done by explicitly finding the zero level set of ϕ, which represents the droplet boundary, and recomputing the signed distances at each grid point in the computational domain. We speed up this calculation by using a coarse sampling of the boundary for computing signed distances of grid nodes far from the boundary. For closer grid nodes, we use a finer sampling. The advantage of keeping it a distance function is that it increases the accuracy of computing spatial derivatives of ϕ. In addition, it ensures $|\nabla\phi| \approx 1$, which increases the accuracy of computing curvature with equation (3.2) because the denominator is close to unity. Other methods exist for maintaining the distance function character of the level set (see [105], [123], [24]), but we decided to use a more straightforward approach.

The other main problem with the level set method is that, even if it is updated with a divergence free velocity field, it does not preserve mass [105]. In general, it tends to lose mass as the simulation progresses. This is mainly due to inherent numerical diffusion in the discretization of equation (3.1). We alleviate this problem by adding an appropriate constant offset to ϕ at each time step. This ensures global mass conservation because the constant offset affects the size of the zero level set (see Figure 3.3). The mass is measured by computing the enclosed area of the zero level set, which is directly proportional to the mass (by incompressibility). If there is more than one droplet, say after a split, then different constants are added to the regions of the level set corresponding to those droplets. Hence, mass is conserved individually for each droplet.

Unfortunately, the constant offset does not ensure local mass conservation, such as in the pinching region of a splitting droplet, which can cause the droplet in our simulation to 'hesitate' while splitting and sometimes get stuck. The two left and right ends would bulge and pull apart as usual, but as the neck joining them became thin it stopped moving. This was completely erroneous because the velocity field inside the droplet dictated that it should split apart. One reason for this is that the level set method does not lose mass at equal rates in different regions of the domain. Therefore, a constant offset cannot properly correct for this. In addition, the dynamics of droplet pinching are not resolved very well because the grid resolution is fixed, uniform, and quite coarse in the pinching region (i.e. only two to five grid points).

Recently, [44] introduced a particle level set method that ensures global and local mass conservation. And in [96], a method for adaptive refinement is described that can resolve fine-scale dynamics. However, the first method is computationally intensive because of the number of seed particles needed to adequately reconstruct the level set as well as the particle re-seeding routines necessary to make the algorithm work. And the second method leads to more involved data structures and coding. Therefore, we opted for the following simpler, faster method for correcting the splitting problem.

First, we check for potential splitting of the droplet by looking for thin necking regions in the flow. This is done by using information contained in the level set function, ϕ, itself. If it is not close to splitting, then we do nothing. Otherwise, we modify ϕ by slightly decreasing its height in a small region around the pinch point at each time step. This prevents the level set from getting stuck and allows it to complete pinch-off without drastic modification to the level set function. In Table 3.1, we present simulation results for the grid resolution versus time to pinch-off for the splitting case discussed in Section 5.1. The splitting time of the simulated droplet only varies by a few percent from the 108x108 grid resolution case used in Section 5.1.3.

3.5.2 Choose Time Step

The next time step is chosen adaptively by the following formula [105]

$$\triangle t = \frac{h}{\max(|u| + |v|)}$$

where h is the grid spacing of the computational domain, u and v are the velocity components at the current time step, and the maximum is taken over all grid nodes in a thin band around the zero level set. It is not necessary to consider the whole domain because we only update level set values inside the thin band. This formula is based on the Courant-Friedrichs-Lewy (CFL) condition, which specifies the largest time step that can be taken and still allow the simulation to remain stable [102]. It is common to have some safety margin by choosing a smaller time step, but this is unnecessary because the HJ-WENO method in Section 3.4.1 has built-in artificial dissipation which enhances stability. For more details, see [105] and [102].

Grid Resolution	*Pinch Time (ms)*	*% Dev.*
84x84	127.9	−0.9
90x90	126.4	−2.0
96x96	125.5	−2.7
102x102	126.0	−2.3
108x108	129.0	+0.0
114x114	132.7	+2.9
120x120	133.9	+3.8
130x130	140.0	+8.5
140x140	131.2	+1.7
150x150	139.8	+8.4
161x161	125.0	−3.1
174x174	134.2	+4.0
187x187	125.4	−2.8
201x201	129.8	+0.6

Table 3.1: Pinch Time Versus Grid Resolution. Simulation results for the grid resolution versus time to pinch-off for the splitting case discussed in Section 5.1. Percent deviations are taken with respect to the 108x108 grid resolution case. The splitting time of the simulated droplet only varies by a few percent.

Chapter 4

Variational Method for Explicit Front Tracking

In the level set method, the interface was captured implicitly by the zero set of a scalar function (see Figure 3.3). For explicit front tracking, an explicit mesh is used to represent the liquid-gas interface (see Figure 1.5b). The following sections describe a variational formulation of the front tracking method that gives a stable way to compute curvature accurately using the Finite Element Method (FEM). This method was pursued in lieu of the difficulties we experienced with the level set method (i.e. noisy curvature calculation and poor mass conservation). It also gives us more tools for analyzing the well-posedness of our EWOD model (i.e. does the partial differential equation (PDE) model have a unique solution?).

The mathematical analysis of free boundary problems is wide ranging. At the PDE modeling level, the well-posedness and long-time behavior of solutions is not trivial due to the nonlinear nature of geometrically driven flows that can cause the solution domain to deform considerably (e.g. motion of droplets by surface tension) [12], [125], [61], [117]. Recently, [26] proved the well-posedness of a nonlinear elastic shell interacting with a Navier-Stokes fluid using higher order Sobolev spaces.

In this chapter, we start by reviewing the variational method and FEM. We then derive the variational formulation of our EWOD model, both with and without the contact line pinning model. We proceed to give an analysis of the well-posedness of the time-discrete and fully discrete variational formulations of the EWOD equations. Knowledge of the well-posedness of our numerical scheme is desirable because it ensures that our method is stable and accurate.

4.1 Introduction to the Variational Method and Finite Elements

A variational formulation [47], [58], [114], is an integral representation of the usual 'strong' form of a partial differential equation, such as that given in equations (2.15) and (2.19). A variational formulation is derived by multiplying a strong form equation with a smooth test function then integrating over the domain where the PDE is defined. We demonstrate this process through Laplace's equation.

4.1.1 Variational Form of Laplace's Equation

The strong form of Laplace's (or Poisson's) equation with Neumann data is given by

$$\begin{aligned} -\Delta p &= f, \text{ in } \Omega, \\ \frac{\partial p}{\partial \vec{n}} &= g, \text{ on } \Gamma, \end{aligned} \tag{4.1}$$

where g and f are given data and $\Delta := \nabla \cdot \nabla = \nabla^2$ is the Laplace operator. Multiplying the first equation by an *arbitrary* smooth test function q (see Figure 4.1) and integrating gives

$$\int_\Omega -\nabla \cdot \nabla p \, q = \int_\Omega fq. \tag{4.2}$$

Next, we perform an integration by parts to get

$$\int_\Omega \nabla p \cdot \nabla q - \int_\Gamma \frac{\partial p}{\partial \vec{n}} \, q = \int_\Omega fq, \tag{4.3}$$

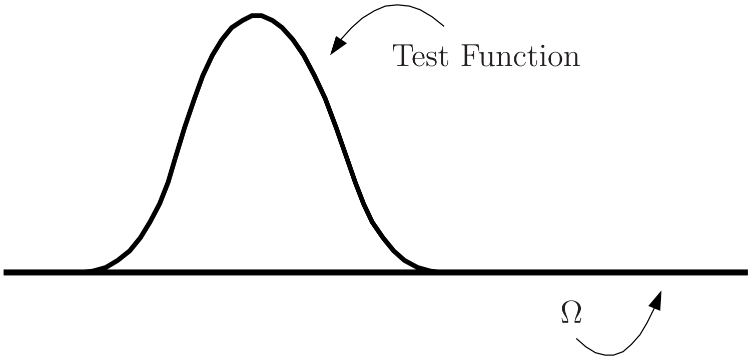


Figure 4.1: Example test function over the domain Ω. Intuitively, test functions are used to extract information about the solution of the PDE through the integral representation. The variational formulation provides a convenient way of encoding the conditions that a function must satisfy in order to be a solution of the PDE. A particular test function provides a very small amount of information, in the *support* of the test function (i.e. where it is non-zero), about the PDE solution. Hence, one must test with *all* possible test functions in order to 'see' the whole solution.

---

which introduces a boundary term. Inserting the boundary condition from (4.1) gives

$$\int_{\Omega} \nabla p \cdot \nabla q = \int_{\Gamma} gq + \int_{\Omega} fq. \tag{4.4}$$

The next step to obtain the variational formulation of (4.1) requires the use of certain function spaces. The first space needed is denoted by $L^2(\Omega)$, which is the space of all functions whose squares are integrable (over the domain Ω) in the sense of Lebesgue. The second space is given by:

$$H^1(\Omega) := \{p \in L^2(\Omega) : \nabla p \in [L^2(\Omega)]^2\}, \tag{4.5}$$

where $[L^2(\Omega)]^2$ is the set of 2-D vector functions, whose individual components are in $L^2(\Omega)$. Functions in H^1 are said to be 'weakly' differentiable because they do not have to be point-wise differentiable (i.e. their derivatives only exist in the Lebesgue sense). This is useful from a theoretical and practical viewpoint, because it gives freedom in approximating the true solution of the strong form PDE.

The variational formulation of (4.1) is now given by the following statement. Find a solution $p \in H^1(\Omega)$ such that

$$\int_\Omega \nabla p \cdot \nabla q = \int_\Gamma g\, q + \int_\Omega f q, \text{ for all } q \in H^1(\Omega), \tag{4.6}$$

where f is in the dual space of $H^1(\Omega)$ and g is in the dual space of $H^{1/2}(\Gamma)$ [59], [21]. The space H^1 was chosen because it is the least 'regular' function space such that the integrals in (4.6) make sense. Note that the solution p is not required to be twice differentiable, as it is in equation (4.1). For this reason, equation (4.6) is also called a *weak* formulation because there are fewer restrictions on the smoothness of p. The well-posedness of this formulation is well known [47] (i.e. it has a unique solution). In addition, if the solution to (4.6) is smooth (i.e. twice differentiable in the usual sense), then it is possible to derive the strong form (4.1) by only assuming the weak form (4.6) is true and (essentially) following the above steps in reverse. In this case, the strong and weak forms are said to be *equivalent*. For most formulations this is true, but one must take care that the introduction of the function spaces in the weak formulation does not introduce any 'strange' behavior in the weak solution.

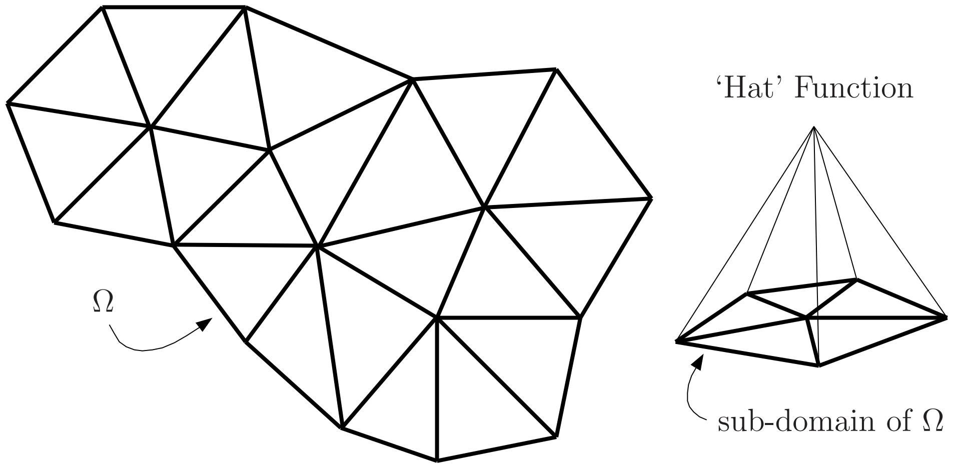


Figure 4.2: Triangulated domain and 'hat' function. On the left, a domain has been partitioned into a (coarse) triangulation, and is denoted by Ω. On the right, a local 'hat' function is shown over a sub-domain of Ω. There is one hat function for each vertex in the triangulation. The collection of hat functions defines a finite dimensional subspace of $H^1(\Omega)$ and is used to approximate solutions of the PDE listed in equation (4.1).

4.1.2 Finite Elements for Laplace's Equation

Finding a solution to (4.6) can be done by writing p and q as linear combinations of basis functions that span all of $H^1(\Omega)$ (i.e. the Galerkin method). This would lead to a set of equations for computing the coefficients of the basis functions. However, the basis functions are not easily obtainable when Ω is an arbitrary domain. Therefore, an alternative method consists of replacing the space $H^1(\Omega)$ by a finite dimensional subspace. One then only needs to construct a finite dimensional basis, which can be constructed in the following way. Let the domain Ω be partitioned into a set of disjoint elements. In 2-D, Ω can be partitioned into a set of triangles, for instance. Then, we define a set of functions which are subordinate to the partitioning. In the case of a triangulation, one can take a set of basis functions, $Q \subset H^1(\Omega)$, to be piecewise linear 'hat' functions that are linear over each triangle and continuous at the edges of the triangles (see Figure 4.2). This gives a new finite dimensional formulation: find a solution $p \in Q$ such that

$$\int_\Omega \nabla p \cdot \nabla q = \int_\Gamma gq + \int_\Omega fq, \text{ for all } q \in Q. \tag{4.7}$$

The well-posedness of this problem is directly inherited from the infinite dimensional case (4.6). Problem (4.7) leads to a sparse linear system of equations that can be solved by many standard techniques for solving matrix equations [18], [20], [76]. This process of choosing a finite dimensional subspace by partitioning the domain into a finite number of elements is called the *finite element method*.

In the following sections, we state the proper strong and weak form of the governing equations for EWOD driven flow. We show the well-posedness of the time-discrete (continuous in space) formulation, as well as for the fully discrete version.

4.2 Variational Formulation of Model Without Line Pinning

In this section, we derive the variational formulation of the governing equations (see Section 2.4.1) and show that the weak and strong forms of that model are equivalent. For clarity, we do not consider contact line pinning at this time. The line pinning formulation only has an additional term to the formulation derived here, and is postponed until Section 4.3. In deriving the variational form, we assume that the domain is C^2 (i.e. its boundary can be locally described by a twice differentiable function).

First, we rewrite the EWOD flow model in Section 2.4.1 in a slightly different form:

$$\begin{aligned} \alpha\partial_t\vec{u} + \beta\vec{u} + \nabla p = 0,\ \Omega, \\ \nabla\cdot\vec{u} = 0,\ \Omega, \\ p = \kappa + E,\ \Gamma, \end{aligned} \tag{4.8}$$

where the first equation is conservation of momentum, the second is conservation of mass, and the third is the pressure boundary condition. Note that Laplace's equation for pressure has been replaced by the second equation in (4.8). The equations in (4.8) are more convenient because they allow us to derive a **mixed** variational formulation of the PDE model that allows the pressure boundary condition to be included as a *natural boundary condition*. This is advantageous for computing curvature (see below). In addition to (4.8), we have the following equation of motion for the time-varying liquid-gas interface Γ

$$\begin{aligned} \vec{X}(\cdot,t) &= \Gamma(t), \\ \frac{d\vec{X}(s,t)}{dt} &= (\vec{u}(\vec{X}(s,t),t) \cdot \vec{n}(\vec{X}(s,t),t))\vec{n}(\vec{X}(s,t),t) + \phi(s,t)\vec{t}(\vec{X}(s,t),t), \end{aligned} \tag{4.9}$$

where $\vec{X}(\cdot,t) : I \to \Gamma(t) \subset \mathbb{R}^2$ is a parameterization (see Appendix A) of the set of points $\Gamma(t)$, s is the parameterization variable, I is the parameterization interval, $\vec{n}$ is the outward pointing normal vector of Γ, $\vec{t}$ is the tangent vector, and $\phi(s,t)$ is any bounded smooth function.

According to (4.9), the time-varying set of points $\Gamma(t)$ (i.e. the interface motion) only depends on the normal component of the velocity ($\vec{u}\cdot\vec{n}$) because the extra tangential component ϕ only serves to re-parameterize $\vec{X}$. In other words, the *shape* of the interface only depends on the normal velocity. And since only the shape matters in regards the boundary condition in (4.8), the extra ϕ term is completely irrelevant for the time-continuous problem. In fact, one can alter the equation of motion to be

$$\frac{d\vec{X}(s,t)}{dt} = \vec{u}(\vec{X}(s,t),t), \tag{4.10}$$

which is what we use in deriving the weak form of the time-discrete problem given in the following sections. We make a point about this ‘irrelevant’ tangential component now to address a small issue with the weak formulation of the time-discrete problem, which is discussed further in Section 4.2.3.

For reference, we call (4.8) and (4.10) the fully continuous strong form of the EWOD flow problem.

4.2.1 Time-Discrete Strong-Form

Next, we define the time-discrete version of (4.8) and (4.10) by first partitioning the time axis into time-steps $\triangle t_i$, for i in some finite index set. Let Ω^i and Γ^i be the domain and interface at time index t_i, and let $\vec{X}^i(\cdot)$ be a parameterization of Γ^i. Let $\vec{X}^{i+1}(\cdot)$ be a parameterization that approximates the shape of the interface at time index t_{i+1} and is given by the following time-discrete version of (4.10)

$$\vec{X}^{i+1}(s) := \vec{X}^i(s) + \triangle t_{i+1} \vec{u}^{i+1}(\vec{X}^i(s)), \text{ for all } s \in I, \tag{4.11}$$

where $\vec{u}^{i+1}$ is the velocity (defined on $\overline{\Omega^i}$) at the next time index.

Using a finite time-step to update the position of the interface introduces a time-discretization error (i.e. a difference between $\vec{X}^{i+1}$ and where the interface 'should' be for the time-continuous case) that is directly related to the size of $\triangle t_{i+1}$ and the velocity $\vec{u}^{i+1}$. For the fully continuous case (4.8), (4.10), the presence of an 'arbitrary' tangential component of the velocity was irrelevant (i.e. only the normal component determines the evolution). Ergo, it is desirable that the discretization error only depend on the normal component of velocity. This is the case if the tangential component is bounded (in some appropriate sense) by the normal component $(\vec{u}^{i+1} \cdot \vec{n}^i)\vec{n}^i$. Therefore, we consider any velocity with the same normal component as the update velocity in (4.11), and whose tangential component is bounded by the normal component, to be *equivalent* to the update velocity appearing in (4.11) with respect to the time-discrete strong form.

We obtain the update velocity by solving a linearized, time-discrete version of equation (4.8), which is listed as

$$\begin{aligned} \alpha \frac{\vec{u}^{i+1} - \vec{u}^{i}}{\triangle t_{i+1}} + \beta \vec{u}^{i+1} + \nabla p^{i+1} &= 0, \ \Omega^i, \\ \nabla \cdot \vec{u}^{i+1} &= 0, \ \Omega^i, \\ p^{i+1} &= \kappa^{i+1} + E^i, \ \Gamma^i, \end{aligned} \tag{4.12}$$

where $\vec{u}^i$ is the (known) velocity at time index t_i. Here, we have used a finite difference approximation of the time derivative term $\partial_t \vec{u}$. Given $\vec{X}^{i+1}$, we define the interface at time t_{i+1} by

$$\Gamma^{i+1} = \{\vec{X}^{i+1}(s) : s \in I\}, \tag{4.13}$$

and κ^{i+1} is an approximation of the curvature of Γ^{i+1}.

The solution domain is kept explicit in (4.12) when solving for the new velocity $\vec{u}^{i+1}$ (i.e. $\vec{u}^{i+1}$ is defined on the current domain Ω^i), which is effectively a linearization step. But the curvature term κ^{i+1} is not explicit, meaning that κ^{i+1} is not the curvature of Γ^i. In fact, we have some freedom in choosing how the curvature is computed. Before stating how the curvature is approximated, we re-write the update (4.11) in a more convenient form:

$$\vec{X}^{i+1} \circ (\vec{X}^i)^{-1}(\vec{x}) = \vec{X}^i \circ (\vec{X}^i)^{-1}(\vec{x}) + \triangle t_{i+1} \vec{u}^{i+1}(\vec{x}), \text{ for all } \vec{x} \in \Gamma^i, \tag{4.14}$$

where we use '$\circ$' to emphasize the composition of maps or functions (i.e. if f and g are functions, then $f \circ g(x) = f(g(x))$). Equation (4.14) can be written more concisely as

$$\vec{X}^{i+1} \circ (\vec{X}^i)^{-1} = \vec{X}^i \circ (\vec{X}^i)^{-1} + \triangle t_{i+1} \vec{u}^{i+1}, \tag{4.15}$$

where $(\vec{X}^i)^{-1} : \Gamma^i \to I$ is the inverse map of $\vec{X}^i$. Obviously, $\vec{X}^i \circ (\vec{X}^i)^{-1}$ is the identity map $\mathrm{id}_{\Gamma^i} : \Gamma^i \to \Gamma^i$, but we prefer to keep it as written for clarity.

From differential geometry [37], [58], [7], the vector curvature of Γ^i and Γ^{i+1} is defined (see Appendix A, equation (A.26)) using the Laplace-Beltrami operator or 'surface Laplacian' by

$$\kappa^{i+1}\vec{n}^{i+1} = -\Delta_{\Gamma^{i+1}}(\vec{X}^{i+1} \circ (\vec{X}^{i+1})^{-1}), \tag{4.16}$$

$$\kappa^{i}\vec{n}^{i} = -\Delta_{\Gamma^{i}}(\vec{X}^{i} \circ (\vec{X}^{i})^{-1}). \tag{4.17}$$

In other words, the vector curvature is given by the surface Laplacian of the identity map on the surface. Formulas (4.16) and (4.17) are true for 1-D and 2-D surfaces. For a 1-D surface, the surface Laplacian of the identity map is just the second derivative, with respect to arc-length, of the curve parameterization. For the purpose of our time-discrete problem, using (4.16) corresponds to an implicit computation of the curvature, while (4.17) is explicit because it is the curvature of the current domain boundary Γ^i. In our case, we use a compromise given by

$$\kappa^{i+1}\vec{n}^{i} := -\Delta_{\Gamma^{i}}(\vec{X}^{i+1} \circ (\vec{X}^{i})^{-1}), \tag{4.18}$$

$$\Rightarrow \kappa^{i+1} = -\Delta_{\Gamma^{i}}(\vec{X}^{i+1} \circ (\vec{X}^{i})^{-1}) \cdot \vec{n}^{i},$$

which is semi-implicit because we use the updated surface parameterization defined by (4.15) but compute the surface Laplacian on the original surface Γ^i.

The above time-discretization was chosen because it leads to a linear set of equations that describe the velocity at each time-step. A fully implicit approach would not be linear because the domain Ω^{i+1} at the next time-step is <u>not</u> known a priori. In addition, the semi-implicit discretization in (4.18) ensures a stable computation of the curvature [8], [9], [43]. This decouples geometry from the physical variables and converts the original nonlinear PDE system into a sequence of *linear* elliptic PDE.

4.2.2 Variational Formulation of the Time-Discrete Problem

From the time-discrete equations given in Section 4.2.1, we proceed to derive the variational formulation by the standard means (see Section 4.1). Let $\vec{v}$ be an arbitrary smooth vector test function and multiply the first equation in (4.12) by $\vec{v}$ and integrate:

$$\frac{\alpha}{\triangle t_{i+1}} \int_{\Omega^i} (\vec{u}^{i+1} - \vec{u}^i) \cdot \vec{v} + \beta \int_{\Omega^i} \vec{u}^{i+1} \cdot \vec{v} + \int_{\Omega^i} \nabla p^{i+1} \cdot \vec{v} = 0. \tag{4.19}$$

Next, integrate the pressure gradient term by parts, plug in the boundary condition in (4.12), and rearrange to get

$$(\frac{\alpha}{\triangle t_{i+1}} + \beta) \int_{\Omega^i} \vec{u}^{i+1} \cdot \vec{v} - \int_{\Omega^i} p^{i+1} \nabla \cdot \vec{v} + \int_{\Gamma^i} \kappa^{i+1} \vec{n}^i \cdot \vec{v} = \frac{\alpha}{\triangle t_{i+1}} \int_{\Omega^i} \vec{u}^i \cdot \vec{v} - \int_{\Gamma^i} E^i \vec{v} \cdot \vec{n}^i. \tag{4.20}$$

We now concentrate on simplifying the boundary term involving the curvature. Using our semi-implicit definition (4.18), we have

$$\int_{\Gamma^i} \kappa^{i+1} \vec{n}^i \cdot \vec{v} = - \int_{\Gamma^i} \Delta_{\Gamma^i} (\vec{X}^{i+1} \circ (\vec{X}^i)^{-1}) \cdot \vec{v}.$$

Using the fact that $\Delta_{\Gamma^i} = \nabla_{\Gamma^i} \cdot \nabla_{\Gamma^i}$ (i.e. the surface Laplacian is the surface divergence of the surface gradient; see Appendix A and equation (A.18)) and integrating by parts on the boundary (A.32), we get

$$\int_{\Gamma^i} \kappa^{i+1} \vec{n}^i \cdot \vec{v} = \int_{\Gamma^i} \nabla_{\Gamma^i} (\vec{X}^{i+1} \circ (\vec{X}^i)^{-1}) \cdot \nabla_{\Gamma^i} \vec{v},$$

where the dot $\cdot$ (on the right side) is the 'double dot product' of two matrices (A.33). Finally, plugging in the update formula (4.15) gives

$$\int_{\Gamma^i} \kappa^{i+1} \vec{n}^i \cdot \vec{v} = \int_{\Gamma^i} \nabla_{\Gamma^i} (\vec{X}^i \circ (\vec{X}^i)^{-1} + \triangle t_{i+1} \vec{u}^{i+1}) \cdot \nabla_{\Gamma^i} \vec{v}, \tag{4.21}$$

where we update with the full velocity $\vec{u}^{i+1}$.

We use the full velocity because it simplifies the implementation. If we only used the normal velocity component, then our computational domain would need to have a C^1 boundary to ensure that the normal vector $\vec{n}^i$ is continuous. This is necessary because of the surface gradient term in (4.21). However, computing with a *moving* C^1 boundary complicates the numerical method, so is undesirable. Alternatively, using a polygonal domain (or iso-parametric elements) to represent the liquid-gas interface (as is typically done with FEM), would mean $\vec{n}^i$ is discontinuous, so $\nabla_{\Gamma^i} \vec{n}^i$ would not make sense in our formulation. But using the full $\vec{u}$ avoids this issue, and is equivalent to just using the normal component anyway (see Section 4.2.1). One way to keep the normal velocity update would be to replace the normal vector by a continuous approximation, but this would introduce extra discretization error <u>and</u> further complicate the method. Ergo, we chose to use $\vec{u}^{i+1}$ to update the interface position.

Combining equation (4.21) with (4.20) and rearranging gives the weak formulation of the first (conservation of momentum) equation:

$$\begin{aligned} (\frac{\alpha}{\triangle t_{i+1}} + \beta) \int_{\Omega^i} \vec{u}^{i+1} \cdot \vec{v} - \int_{\Omega^i} p^{i+1} \nabla \cdot \vec{v} + \triangle t_{i+1} \int_{\Gamma^i} \nabla_{\Gamma^i} \vec{u}^{i+1} \cdot \nabla_{\Gamma^i} \vec{v} = \\ \frac{\alpha}{\triangle t_{i+1}} \int_{\Omega^i} \vec{u}^i \cdot \vec{v} - \int_{\Gamma^i} E^i \vec{v} \cdot \vec{n}^i - \int_{\Gamma^i} \nabla_{\Gamma^i} (\vec{X}^i \circ (\vec{X}^i)^{-1}) \cdot \nabla_{\Gamma^i} \vec{v}. \end{aligned} \tag{4.22}$$

The variational form of the second (conservation of mass) equation in (4.12) is obtained simply by multiplying by an arbitrary smooth test function q and integrating:

$$\int_{\Omega^i} q\nabla \cdot \vec{u}^{i+1} = 0. \tag{4.23}$$

Next, we need the correct function space to pose the variational form in. We start by defining a scalar product $s_i(\cdot,\cdot) : C^\infty(\overline{\Omega^i}) \times C^\infty(\overline{\Omega^i}) \to \mathbb{R}$ by

$$s_i(\vec{u},\vec{v}) := \int_{\Omega^i} \vec{u}\cdot\vec{v} + \int_{\Omega^i} (\nabla\cdot\vec{u})(\nabla\cdot\vec{v}) + \int_{\Gamma^i} \vec{u}\cdot\vec{v} + \int_{\Gamma^i} \nabla_{\Gamma^i}\vec{u}\cdot\nabla_{\Gamma^i}\vec{v}, \tag{4.24}$$

and define a norm induced by this scalar product $|||\vec{u}||| := \sqrt{s_i(\vec{u},\vec{u})}$. We then define the function space $\mathbb{V}^i$ as the closure of $C^\infty(\overline{\Omega^i})$ with respect to the norm $|||\cdot|||$, i.e.

$$\mathbb{V}^i := \overline{C^\infty(\overline{\Omega^i})}^{|||\cdot|||}. \tag{4.25}$$

In other words, the space $\mathbb{V}^i$ is the set of functions that are limits of $C^\infty(\overline{\Omega^i})$ Cauchy sequences that converge with respect to the norm $|||\cdot|||$ [84], [87], [47]. We denote this norm by:

$$\|\vec{u}\|_{\mathbb{V}^i} := \sqrt{s_i(\vec{u},\vec{u})}. \tag{4.26}$$

Therefore, the space $\mathbb{V}^i$ and norm $\|\cdot\|_{\mathbb{V}^i}$ define a Hilbert space [47], [84], which is required for the well-posedness of the variational formulation. We now make note of some functional relations that will be useful later in showing the discrete INF-SUP condition. Using Definition 3.2, Corollary 3.4, and Theorem 3.17 in [1], one can show the following inclusions:

$$H^1(\Omega^i) \cap H^1(\Gamma^i) \subset \mathbb{V}^i \subset H(\mathrm{div},\Omega^i) \cap H^1(\Gamma^i), \tag{4.27}$$

where $H(\mathrm{div},\Omega^i) := \{\vec{v} \in [L^2(\Omega^i)]^2 : \nabla\cdot\vec{v} \in L^2(\Omega^i)\}$ [59].

The variational formulation of the problem now reads as follows: find a velocity $\vec{u} \in \mathbb{V}^i$ and pressure $p \in \mathbb{P}^i$, such that (4.22) and (4.23) are true for all $\vec{v} \in \mathbb{V}^i$ and all $q \in \mathbb{P}^i$, where $\mathbb{P}^i := L^2(\Omega^i)$. The electrowetting force E^i is required to be in the dual space $\mathbb{M}$ (discussed later; see equation (4.53)), although in practice it is actually more regular. The boundary Γ^i must be, at least, Lipschitz to make sense of the surface gradient operator ∇_{Γ^i}. This is satisfied in our computations because we use a continuous approximation of the boundary that is piecewise C^1 (i.e. a polygon with curved iso-parametric sides) (see [1] for more details). The spaces $\mathbb{V}^i$ and $\mathbb{P}^i$ are the ones with the minimal regularity needed to make sense of the integrals appearing in (4.22) and (4.23). Note that the superscript i denotes dependence on the current domain Ω^i.

4.2.3 Equivalence of Weak and Strong Forms

Next, we show that the weak formulation implies the strong form assuming the solution is smooth and the domain is smooth. Normally this is obvious, but in this case there is an 'artifact' of the weak formulation that arises because of the definition of $\mathbb{V}^i$. The space $\mathbb{V}^i$ contains vector velocity functions whose tangential component on Γ^i is unrelated to the vector function values in the interior of Ω^i. This could affect the equivalence with the time-discrete strong form in Section 4.2.1 if the tangential component of the solution is completely arbitrary. We now clarify this issue.

Notation for Interior and Decoupled Tangential Velocity

First, we define some notation for a function in $\mathbb{V}^i$. Let $\vec{v} \in \mathbb{V}^i$, which is a function defined on the closure $\overline{\Omega^i}$, and let $\vec{v}_{int}$ denote $\vec{v}$ on the open set Ω^i. Because of a standard result for the normal trace of $H(\text{div}, \Omega^i)$ functions, [59], we know that $\vec{v} \cdot \vec{n} = \vec{v}_{int} \cdot \vec{n}$ on Γ^i. Which means that the normal component of the velocity on the boundary is dependent on the velocity in the interior. No such result exists for the tangential component. In other words, there could be a 'jump' in the tangential velocity (i.e. $\vec{v}_{int} \cdot \vec{t} \neq \vec{v}|_{\Gamma^i} \cdot \vec{t}$). So we let $\vec{v}_{\vec{t}}$ denote the tangential component of $\vec{v}$ on Γ^i, which has no connection with $\vec{v}_{int}$. Therefore, when we write $\vec{v} \in \mathbb{V}^i$, we are really referring to the pair of functions $(\vec{v}_{int}, \vec{v}_{\vec{t}})$ that are decoupled.

Derive Strong Form Equations in the Bulk

Next we derive the strong form of the governing PDE from the weak formulation by assuming that the domain is smooth and that we have a smooth (i.e. $C^\infty(\overline{\Omega^i})$) solution $(\vec{u}, p)$ of the variational form. Let $\vec{v}$ be a smooth test function and apply it to (4.22). After integrating the pressure term by parts, we get

$$(\frac{\alpha}{\triangle t_{i+1}} + \beta) \int_{\Omega^i} \vec{u}^{i+1} \cdot \vec{v} + \int_{\Omega^i} \nabla p^{i+1} \cdot \vec{v} - \int_{\Gamma^i} p^{i+1} \vec{n}^i \cdot \vec{v} + \triangle t_{i+1} \int_{\Gamma^i} \nabla_{\Gamma^i} \vec{u}^{i+1} \cdot \nabla_{\Gamma^i} \vec{v} = \frac{\alpha}{\triangle t_{i+1}} \int_{\Omega^i} \vec{u}^i \cdot \vec{v} - \int_{\Gamma^i} E^i \vec{v} \cdot \vec{n}^i - \int_{\Gamma^i} \nabla_{\Gamma^i} (\vec{X}^i \circ (\vec{X}^i)^{-1}) \cdot \nabla_{\Gamma^i} \vec{v}. \tag{4.28}$$

Now suppose $\vec{v}$ has compact support in Ω^i, meaning $\vec{v}$ is chosen so that it has zero value on the boundary Γ^i. This eliminates the boundary terms and, after rearranging, gives

$$\int_{\Omega^i} (\alpha \frac{\vec{u}_{int}^{i+1} - \vec{u}_{int}^{i}}{\triangle t_{i+1}} + \beta \vec{u}_{int}^{i+1} + \nabla p^{i+1}) \cdot \vec{v}_{int} = 0, \tag{4.29}$$

where we have replaced $\vec{u}$ and $\vec{v}$ with $\vec{u}_{int}$ and $\vec{v}_{int}$ because the integral is on the open set Ω^i. Since (4.29) is true for all smooth functions $\vec{v}_{int}$, we get the strong form of the momentum equation:

$$\alpha \frac{\vec{u}_{int}^{i+1} - \vec{u}_{int}^{i}}{\triangle t_{i+1}} + \beta \vec{u}_{int}^{i+1} + \nabla p^{i+1} = 0, \text{ in } \Omega^i.$$

The conservation of mass equation follows similarly by letting q be smooth and arbitrary in (4.23),

$$\nabla \cdot \vec{u}_{int}^{i+1} = 0, \text{ in } \Omega^i.$$

Derive Strong Form of the Pressure Boundary Condition

Now that we have the strong mass and momentum equations, (4.28) simplifies to

$$\begin{aligned} &-\int_{\Gamma^i} p^{i+1} \vec{n}^i \cdot \vec{v} + \triangle t_{i+1} \int_{\Gamma^i} \nabla_{\Gamma^i} \vec{u}^{i+1} \cdot \nabla_{\Gamma^i} \vec{v} = \\ &-\int_{\Gamma^i} E^i \vec{v} \cdot \vec{n}^i - \int_{\Gamma^i} \nabla_{\Gamma^i} (\vec{X}^i \circ (\vec{X}^i)^{-1}) \cdot \nabla_{\Gamma^i} \vec{v}. \end{aligned} \tag{4.30}$$

Because the domain and $\vec{v}$ are smooth, we can integrate the last term by parts to get

$$\int_{\Gamma^i} \nabla_{\Gamma^i} (\vec{X}^i \circ (\vec{X}^i)^{-1}) \cdot \nabla_{\Gamma^i} \vec{v} = -\int_{\Gamma^i} \Delta_{\Gamma^i} (\vec{X}^i \circ (\vec{X}^i)^{-1}) \cdot \vec{v} = \int_{\Gamma^i} \kappa^i \vec{n}^i \cdot \vec{v}, \tag{4.31}$$

by definition of the curvature of Γ^i (4.17). But this implies that the left-hand side of (4.31) is zero if the test function $\vec{v}$ is purely tangential. Therefore, the last integral in (4.30) only sees the normal component of $\vec{v}$, which allows us to replace $\vec{v}$ by $\vec{v}_{int}$ in three of the four integrals of (4.30) because they only depend on the normal component. This gives

$$\begin{aligned} -\int_{\Gamma^i} p^{i+1}\vec{n}^i \cdot \vec{v}_{int} + \triangle t_{i+1}\int_{\Gamma^i} \nabla_{\Gamma^i}\vec{u}^{i+1} \cdot \nabla_{\Gamma^i}((\vec{v}_{int} \cdot \vec{n}^i)\vec{n}^i + \vec{v}_{\vec{t}}) = \\ -\int_{\Gamma^i} E^i\vec{v}_{int} \cdot \vec{n}^i - \int_{\Gamma^i} \nabla_{\Gamma^i}(\vec{X}^i \circ (\vec{X}^i)^{-1}) \cdot \nabla_{\Gamma^i}\vec{v}_{int}, \end{aligned} \tag{4.32}$$

where we have written $\vec{v}$ (in the other remaining integral) in terms of the decoupled normal and tangential components, both of which are assumed to be smooth and arbitrary.

Next, we are free to take $\vec{v}_{int} = 0$ but keep $\vec{v}_{\vec{t}}$ arbitrary (because it is a test function). This gives

$$\int_{\Gamma^i} \nabla_{\Gamma^i}((\vec{u}^{i+1}_{int} \cdot \vec{n}^i)\vec{n}^i + \vec{u}^{i+1}_{\vec{t}}) \cdot \nabla_{\Gamma^i}\vec{v}_{\vec{t}} = 0, \tag{4.33}$$

where we have expanded $\vec{u}^{i+1}$ in terms of its decoupled components. We then fix $\vec{v}_{\vec{t}} = \vec{u}^{i+1}_{\vec{t}}$ and obtain the relation:

$$\int_{\Gamma^i} \nabla_{\Gamma^i}\vec{u}^{i+1}_{\vec{t}} \cdot \nabla_{\Gamma^i}\vec{u}^{i+1}_{\vec{t}} = -\int_{\Gamma^i} \nabla_{\Gamma^i}((\vec{u}^{i+1}_{int} \cdot \vec{n}^i)\vec{n}^i) \cdot \nabla_{\Gamma^i}\vec{u}^{i+1}_{\vec{t}}. \tag{4.34}$$

After taking the absolute value and applying the Cauchy-Schwarz inequality, we get

$$\begin{aligned} \|\nabla_{\Gamma^i}\vec{u}^{i+1}_{\vec{t}}\|^2_{0,\Gamma^i} \le \|\nabla_{\Gamma^i}((\vec{u}^{i+1}_{int} \cdot \vec{n}^i)\vec{n}^i)\|_{0,\Gamma^i}\|\nabla_{\Gamma^i}\vec{u}^{i+1}_{\vec{t}}\|_{0,\Gamma^i}, \\ \Rightarrow \|\nabla_{\Gamma^i}\vec{u}^{i+1}_{\vec{t}}\|_{0,\Gamma^i} \le \|\nabla_{\Gamma^i}((\vec{u}^{i+1}_{int} \cdot \vec{n}^i)\vec{n}^i)\|_{0,\Gamma^i}. \end{aligned} \tag{4.35}$$

In fact, because $\vec{u}^{i+1}_{\vec{t}}$ is tangential (see Appendix A, Lemma A.6.2), we get that the full $H^1(\Gamma)$ norm of the tangential component is controlled by the $H^1(\Gamma)$ semi-norm of the normal component:

$$\|\vec{u}^{i+1}_{\vec{t}}\|_{1,\Gamma^i} \le C\|\nabla_{\Gamma^i}((\vec{u}^{i+1}_{int} \cdot \vec{n}^i)\vec{n}^i)\|_{0,\Gamma^i}, \tag{4.36}$$

where the constant C only depends on the diameter of Γ^i. This means that the decoupled tangential component is not completely arbitrary, and is bounded by the normal component of the interior velocity.

Because the decoupled tangential component is bounded by the normal component, we have that the full velocity $\vec{u}^{i+1}$ on Γ^i is *equivalent* to $\vec{u}^{i+1}_{int}$ with respect to the interface update equation (4.11) (or the more concisely written equation (4.15)). Hence, we can apply (4.15) to get the strong form of the boundary condition for the time-discrete problem. We proceed by first rearranging (4.32) and setting $\vec{v}_{\vec{t}} = 0$,

$$\begin{aligned} &-\int_{\Gamma^i} p^{i+1}\vec{n}^i \cdot \vec{v}_{int} + \int_{\Gamma^i} E^i \vec{v}_{int} \cdot \vec{n}^i + \\ &+\int_{\Gamma^i} \nabla_{\Gamma^i}((\vec{X}^i \circ (\vec{X}^i)^{-1}) + \triangle t_{i+1}\vec{u}^{i+1}) \cdot \nabla_{\Gamma^i}((\vec{v}_{int} \cdot \vec{n}^i)\vec{n}^i) = 0, \end{aligned} \tag{4.37}$$

where we have the full velocity as the update. Plugging in the equivalent update (4.15), while choosing a $\vec{v}_{int}$ such that $\vec{v}_{int}\big|_{\Gamma^i} = \varphi\vec{n}^i$ on the boundary, gives

$$-\int_{\Gamma^i} p^{i+1}\varphi + \int_{\Gamma^i} \nabla_{\Gamma^i}(\vec{X}^{i+1} \circ (\vec{X}^i)^{-1}) \cdot \nabla_{\Gamma^i}(\varphi\vec{n}^i) + \int_{\Gamma^i} E^i\varphi = 0. \tag{4.38}$$

Integrating the surface gradient term by parts and using the approximation of the updated curvature (4.18), we get

$$\begin{aligned} -\int_{\Gamma^i} p^{i+1}\varphi + \int_{\Gamma^i} (\kappa^{i+1}\vec{n}^i) \cdot (\varphi\vec{n}^i) + \int_{\Gamma^i} E^i\varphi = 0, \\ \Rightarrow \int_{\Gamma^i} (-p^{i+1} + \kappa^{i+1} + E^i)\varphi = 0 \end{aligned} \tag{4.39}$$

for all smooth φ. This gives the strong form of the pressure boundary condition for the time-discrete problem

$$p^{i+1} = \kappa^{i+1} + E^i,\ \Gamma^i.$$

Therefore, we have proved that the weak and strong forms of the time-discrete problem are equivalent. The decoupled tangential component in the weak formulation is not completely arbitrary and is bounded by the normal component. Hence, it only perturbs the evolution of the liquid-gas interface within the same time-discretization error (at each time-step) as if we had completely neglected the tangential component.

The well-posedness of this variational formulation is a special case of the well-posedness of the formulation that includes the contact line pinning model. So we defer answering this question until the end of Section 4.3.

4.3 Variational Formulation of Model Including Line Pinning

The variational formulation provides a natural way to include the contact line pinning model. This is another instance where the variational technique is able to improve on our level set method. The following sections start by rewriting the strong form of the time-discrete EWOD model and simplifying the notation. We skip writing the strong form of the fully continuous problem, because it is only a minor modification of (4.8) and (4.9) by the model given in Section 2.4.2. Next, we give the variational formulation, which is a minor adjustment to the formulation given in Section 4.2.2. Then, we discuss the well-posedness of the time-discrete and fully discrete equations.

4.3.1 Time-Discrete Strong Form

For simplicity, we only consider one time-step of the time-discrete problem. This allows us to drop the time index notation, which is convenient. The following equations are similar to those given in Section 4.2.1.

The interface update equation is given by

$$\vec{Y} \circ \vec{X}^{-1} = \vec{X} \circ \vec{X}^{-1} + \triangle t \, \vec{u}, \tag{4.40}$$

where $\vec{X}$ is a parameterization of the interface at the initial time t_0, $\vec{Y}$ is the new updated interface parameterization, and $\triangle t$ is the time-step. The velocity $\vec{u}$ is at the next time index t_1, and is obtained by solving

$$\begin{aligned} \alpha \frac{\vec{u} - \vec{u}^0}{\triangle t} + \beta \vec{u} + \nabla p = 0, \text{ in } \Omega, \\ \nabla \cdot \vec{u} = 0, \text{ in } \Omega, \\ p = \kappa + E + \lambda, \text{ on } \Gamma, \end{aligned} \tag{4.41}$$

where $\vec{u}^0$ is the velocity at time index t_0 and λ is given by

$$\lambda = P_{pin} \, \text{sgn}(\vec{u} \cdot \vec{n}), \; \Gamma, \tag{4.42}$$

where Ω and Γ denote the domain and interface at t_0, $\vec{n}$ is the outward normal vector of Γ, E is the electrowetting forcing at t_0, and λ is the contact line pinning 'pressure.' The curvature κ is defined using the semi-implicit scheme in (4.18), namely

$$\begin{aligned} \kappa \vec{n} &:= -\Delta_\Gamma (\vec{Y} \circ \vec{X}^{-1}), \\ \Rightarrow \kappa &= -\Delta_\Gamma (\vec{Y} \circ \vec{X}^{-1}) \cdot \vec{n}. \end{aligned} \tag{4.43}$$

4.3.2 Variational Formulation of the Time-Discrete Problem

The variational form of the EWOD pinning model is the same as in Section 4.2.2, except an extra unknown variable λ appears in the pressure boundary condition. This means we need another equation to close the system. The pinning variable λ acts as an inequality constraint on the velocity $\vec{u}$. Therefore, we append a variational inequality to our previous variational form [80], [60], [42]. This is outlined as follows.

Assuming we have a smooth solution to (4.41) on a smooth domain, we multiply it by smooth test functions and integrate to get

$$\begin{split}(\frac{\alpha}{\triangle t} + \beta) \int_{\Omega} \vec{u} \cdot \vec{v} - \int_{\Omega} p \nabla \cdot \vec{v} + \triangle t \int_{\Gamma} \nabla_{\Gamma} \vec{u} \cdot \nabla_{\Gamma} \vec{v} + \int_{\Gamma} \lambda \vec{v} \cdot \vec{n} = \\ \frac{\alpha}{\triangle t} \int_{\Omega} \vec{u}^{0} \cdot \vec{v} - \int_{\Gamma} E \vec{v} \cdot \vec{n} - \int_{\Gamma} \nabla_{\Gamma} (\vec{X} \circ \vec{X}^{-1}) \cdot \nabla_{\Gamma} \vec{v},\end{split} \tag{4.44}$$

$$\int_{\Omega} q \nabla \cdot \vec{u} = 0, \tag{4.45}$$

which is similar to the non-pinning case except an additional term, $\int_{\Gamma} \lambda \vec{v} \cdot \vec{n}$, is in (4.44) and λ satisfies (4.42). We now proceed to derive an inequality relation that is a direct consequence of the definition (4.42). This derivation is extremely non-obvious, but its purpose will be made clear later.

Derive Variational Inequality

First, let K be a set of smooth functions defined on Γ in the following way (see Figure 4.3)

$$K = \{\mu \in C^{\infty}(\Gamma) : |\mu| \leq P_{pin}\}, \tag{4.46}$$

(i.e. the set K is a convex set in the space of smooth functions defined on the interface Γ). Now let $\mu \in K$ and consider the following inequality:

$$(\vec{u} \cdot \vec{n})\mu \leq (\vec{u} \cdot \vec{n})\lambda. \tag{4.47}$$

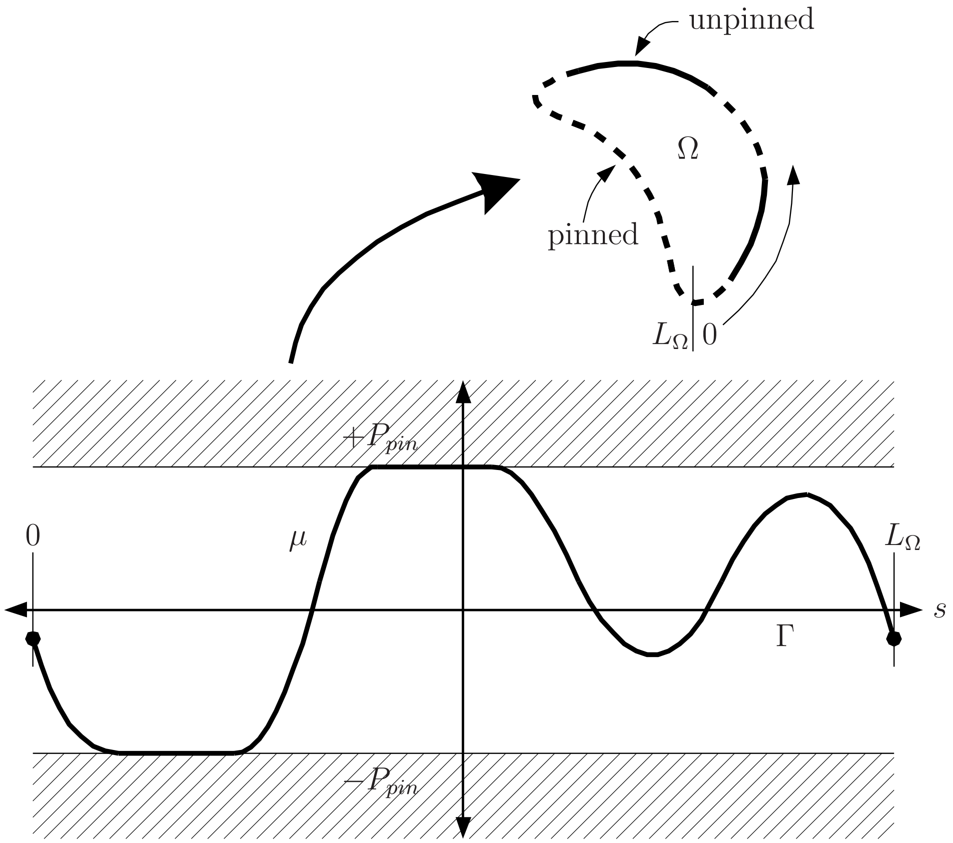


Figure 4.3: Convex set of functions. A plot of an example function μ in the convex set K is shown. The vertical axis is the amplitude and the s axis is along the interface Γ (μ is a function defined on Γ). The function μ is periodic because Γ is a closed curve. Above the plot, an example droplet domain Ω is shown with dashed lines indicating pinned regions of the boundary where $|\mu| < P_{pin}$. Functions in K are limited in amplitude to $\pm P_{pin}$ point-wise. The set K is convex because of the following: let μ_1 and μ_2 be in K, and let $\mu = (1-t)\mu_1 + t\mu_2$, where $0 \leq t \leq 1$. Then clearly $-P_{pin} \leq \mu \leq +P_{pin}$ at each point on Γ, which implies μ is in K. Hence, by the definition of convexity, K is convex.

That inequality is true for the following reason. If $\vec{u} \cdot \vec{n} = 0$, then (4.47) is clearly satisfied. If $\vec{u} \cdot \vec{n} > 0$, then $\lambda = P_{pin}$ and dividing by $(\vec{u} \cdot \vec{n})$ gives

$$\mu \leq P_{pin}, \tag{4.48}$$

which is satisfied because $\mu \in K$. If $\vec{u} \cdot \vec{n} < 0$, then $\lambda = -P_{pin}$ and dividing by $(\vec{u} \cdot \vec{n})$ flips the inequality to give

$$\mu \geq -P_{pin}, \tag{4.49}$$

which is also satisfied. Upon rearranging (4.47) and integrating, we get

$$\int_{\Gamma} (\vec{u} \cdot \vec{n})(\mu - \lambda) \leq 0, \tag{4.50}$$

which is the *variational inequality* we were looking for.

The relation (4.50) (and the set K) actually implies the formula (4.42) by the following argument. First, suppose $\mu \in K$ and can be arbitrarily chosen, and let $\lambda \in K$ with $|\lambda| < P_{pin}$ in an open set of the free surface Γ. Given a smooth function φ with support in the same open set, we can always find $\delta > 0$ sufficiently small such that $\mu := \lambda + \delta\varphi$ is still contained in K. Inserting this function μ into (4.50) we realize that

$$\int_{\Gamma} (\vec{u} \cdot \vec{n})\varphi = 0 \qquad \Rightarrow \qquad \vec{u} \cdot \vec{n} = 0,$$

because φ can have either sign and is arbitrary. Alternatively, if $\lambda = +P_{pin}$ (in an open set of Γ), then we can only choose $\mu := \lambda - \delta\varphi$ with $\delta > 0$ sufficiently small and the function $\varphi \geq 0$ to guarantee $\mu \in K$. This yields

$$\int_{\Gamma} (\vec{u} \cdot \vec{n})\varphi \geq 0 \qquad \Rightarrow \qquad \vec{u} \cdot \vec{n} \geq 0. \tag{4.51}$$

Likewise, we get a relation $\vec{u} \cdot \vec{n} \leq 0$ when $\lambda = -P_{pin}$. This implies that $\lambda = P_{pin}\ \mathrm{sgn}(\vec{u} \cdot \vec{n})$, where λ is free to vary between $-P_{pin}$ and $+P_{pin}$ when $\vec{u} \cdot \vec{n} = 0$. Altogether, this shows the equivalence of (4.42) and (4.50) (with the set K).

The variational inequality allows us to treat λ as an additional unknown, and embeds the relation (4.42) into the act of solving the weak formulation. This is advantageous because it avoids introducing a discontinuous function into our method and it captures the inequality constraint exactly.

Function Space For Pinning Variable

Before precisely stating the full weak formulation of the pinning model, we must specify the proper space in which the pinning variable λ lives. In other words, given $\vec{v}$ in $\mathbb{V}$, what is the correct space for λ such that $\int_\Gamma \lambda \vec{v} \cdot \vec{n}$ 'makes sense' *and* allows for the weak formulation to be well-posed? It turns out that λ should be viewed as a *functional* [87] that acts on objects of the form $\vec{v} \cdot \vec{n}$. In this case, the integral does not make sense, so we rewrite the weak form using the *duality pairing* (i.e. $< \lambda, \vec{v} \cdot \vec{n} > =: \int_\Gamma \lambda \vec{v} \cdot \vec{n}$).

We now precisely define the duality pairing. Let $G = \{\eta : \eta = \vec{v} \cdot \vec{n},\ \text{where } \vec{v} \in [H^1(\Gamma)]^2\}$. The space G is a Banach space endowed with the norm

$$\|\eta\|_G := \inf_{\substack{\vec{v} \in [H^1(\Gamma)]^2 \\ \vec{v} \cdot \vec{n} = \eta}} \|\vec{v}\|_{1,\Gamma}. \tag{4.52}$$

Next, let G^* be the dual space of G (i.e. the set of functionals defined on G) [47], [87], and let $< \cdot, \cdot >$ denote the duality pairing between G and G^*. The space for the pinning variable is defined to be

$$\mathbb{M} := \{\mu \in G^* : < \mu, 1 > = 0\}, \tag{4.53}$$

i.e. the space of functionals with *mean value zero*. This is done to enforce a unique solution for λ in the variational formulation. Physically, this is due to the fact that since λ depends on the velocity, and the velocity is unaffected by constant offsets to the pressure boundary condition, λ is arbitrary up to a constant. The norm on $\mathbb{M}$ is then given by

$$\|\mu\|_{\mathbb{M}} = \sup_{\eta \in G} \frac{< \mu, \eta >}{\|\eta\|_G}. \tag{4.54}$$

Next, we define a closed convex subset of $\mathbb{M}$ (analogous to K used before) that captures the inequality constraint dictated by equation (4.42) within the framework of this particular dual space.

Convex Set of Functionals

In order to include the condition of equation (4.42) into our variational framework, we introduce the following closed convex set of functionals:

$$\begin{gathered} \Lambda := \{\mu \in \mathbb{M} : -\, P_{pin} < 1, \vec{v} \cdot \vec{n} > \leq < \mu, \vec{v} \cdot \vec{n} > \leq \ P_{pin} < 1, \vec{v} \cdot \vec{n} >, \\ \text{for all } \vec{v} \in [H^1(\Gamma)]^2 \text{ such that } \vec{v} \cdot \vec{n} \geq 0 \text{ a.e.}\}. \end{gathered} \tag{4.55}$$

The convexity is clear. We now show why it is closed. Let $\{\mu_k\} \subset \Lambda$ be a sequence such that $\mu_k \longrightarrow \mu \in \mathbb{M}$. This means that

$$\lim_{k \to \infty} < \mu_k, \vec{v} \cdot \vec{n} > = < \mu, \vec{v} \cdot \vec{n} >, \text{ for all } \vec{v} \in [H^1(\Gamma)]^2.$$

So, since $\mu_k \in \Lambda$ for all k, we have that μ satisfies

$$-P_{pin} < 1, \vec{v} \cdot \vec{n} > \leq < \mu, \vec{v} \cdot \vec{n} > \leq \ P_{pin} < 1, \vec{v} \cdot \vec{n} >, \tag{4.56}$$

for all $\vec{v}$ in $[H^1(\Gamma)]^2$ such that $\vec{v} \cdot \vec{n} \geq 0$ almost everywhere (a.e.), which means $\mu \in \Lambda$. Therefore, Λ is closed.

The definition of Λ in (4.55) is somewhat abstract. However, if $\mu \in \Lambda$ can be identified with a function in $L^2(\Gamma)$, then μ satisfies $|\mu| \leq P_{pin}$ a.e. by the following argument. Suppose μ is in $L^2(\Gamma) \cap \Lambda$. Then the duality pairing can be replaced by an integral, and the condition in Λ reads

$$-\int_\Gamma P_{pin} \vec{v} \cdot \vec{n} \leq \int_\Gamma \mu \vec{v} \cdot \vec{n} \leq \int_\Gamma P_{pin} \vec{v} \cdot \vec{n}, \text{ for all } \vec{v} \in [H^1(\Gamma)]^2 \text{ such that } \vec{v} \cdot \vec{n} \geq 0 \text{ a.e.}$$

Then, since $\vec{v} \cdot \vec{n} \geq 0$ a.e., we have that

$$\int_\Gamma (P_{pin} - \mu) \vec{v} \cdot \vec{n} \geq 0 \Rightarrow (P_{pin} - \mu) \geq 0 \Rightarrow \mu \leq +P_{pin}, \text{ a.e.,}$$

$$\int_\Gamma (P_{pin} + \mu) \vec{v} \cdot \vec{n} \geq 0 \Rightarrow (P_{pin} + \mu) \geq 0 \Rightarrow \mu \geq -P_{pin}, \text{ a.e.,}$$

which means

$$|\mu| \leq P_{pin}, \text{ a.e.,}$$

and is the condition defined earlier by the convex set K. So, clearly the definition (4.55) captures the condition set by equation (4.42).

Variational Equations for the Pinning Case

Finally, we state the full variational formulation of the EWOD pinning model. The function space for the velocity is denoted by $\mathbb{V}$ and is a complete space under a certain norm (recalled below). The pressure space is given by $\mathbb{P} := L^2(\Omega)$, and the pinning multiplier space $\mathbb{M}$ is defined to be a set of mean value zero functionals (see (4.53)).

The variational formulation now reads as follows: find a solution $\vec{u}$ in $\mathbb{V}$, p in $\mathbb{P}$, and λ in Λ such that

$$\begin{aligned}\left(\frac{\alpha}{\triangle t}+\beta\right)\left(\int_\Omega \vec{u}\cdot\vec{v}+\gamma\int_\Omega(\nabla\cdot\vec{u})(\nabla\cdot\vec{v})\right)+\triangle t\int_\Gamma \nabla_\Gamma\vec{u}\cdot\nabla_\Gamma\vec{v}+&\\ -\int_\Omega p\nabla\cdot\vec{v}+<\lambda,\vec{v}\cdot\vec{n}>=&\\ \frac{\alpha}{\triangle t}\int_\Omega \vec{u}^0\cdot\vec{v}-\int_\Gamma E\vec{v}\cdot\vec{n}\;-\int_\Gamma \nabla_\Gamma(\vec{X}\circ\vec{X}^{-1})\cdot\nabla_\Gamma\vec{v},\ \text{for all } \vec{v}\in\mathbb{V},&\end{aligned} \tag{4.57}$$

$$-\int_\Omega q\nabla\cdot\vec{u}=0,\ \text{for all } q\in\mathbb{P}, \tag{4.58}$$

$$<(\mu-\lambda),(\vec{u}\cdot\vec{n})>\leq 0,\ \text{for all } \mu\in\Lambda. \tag{4.59}$$

We have added an augmented lagrangian term $\int_\Omega(\nabla\cdot\vec{u})(\nabla\cdot\vec{v})$, with parameter $\gamma>0$, to the first equation. This is consistent with the governing PDE because the velocity is divergence free by (4.58). This is added to improve the convergence of our algorithm for solving the variational inequality (see Section 4.3.6) and will be discussed later.

Next, we recall the appropriate norms for these spaces:

$$\|\vec{v}\|^2_{\mathbb{V}}:=\|\vec{v}\|^2_{H(\mathrm{div},\Omega)}+\|\vec{v}\|^2_{H^1(\Gamma)}, \tag{4.60}$$

$$\|q\|_{\mathbb{P}}:=\|q\|_{L^2(\Omega)}, \tag{4.61}$$

and the $\mathbb{M}$ norm $\|\mu\|_{\mathbb{M}}$ was defined in (4.54).

Now, let us define the following bilinear and linear forms:

$$a(\vec{u},\vec{v})=\left(\frac{\alpha}{\triangle t}+\beta\right)\left(\int_\Omega \vec{u}\cdot\vec{v}+\gamma\int_\Omega(\nabla\cdot\vec{u})(\nabla\cdot\vec{v})\right)+\triangle t\int_\Gamma \nabla_\Gamma\vec{u}\cdot\nabla_\Gamma\vec{v}, \tag{4.62}$$

$$b(\vec{v},(q,\mu))=-\int_\Omega q\nabla\cdot\vec{v}+<\mu,\vec{v}\cdot\vec{n}>, \tag{4.63}$$

$$\chi(\vec{v}) = \frac{\alpha}{\triangle t}\int_\Omega \vec{u}^0 \cdot \vec{v} \; - \int_\Gamma E\vec{v}\cdot\vec{n} \; - \int_\Gamma \nabla_\Gamma(\vec{X}\circ\vec{X}^{-1})\cdot\nabla_\Gamma\vec{v}, \tag{4.64}$$

With (4.62), (4.63), and (4.64), we can rewrite the variational form more concisely: find $(\vec{u}, p, \lambda) \in \mathbb{V}\times\mathbb{P}\times\Lambda$ such that

$$\begin{aligned} a(\vec{u},\vec{v}) + b(\vec{v},(p,\lambda)) &= \chi(\vec{v}), && \text{for all } \vec{v}\in\mathbb{V}, \\ b(\vec{u},(q,\mu-\lambda)) &\le 0, && \text{for all } (q,\mu)\in\mathbb{P}\times\Lambda. \end{aligned} \tag{4.65}$$

This is called a mixed variational formulation because there is more than one function space being used and they are different [21], [59]. The well-posedness of this formulation is discussed in Section 4.3.3. It should be noted that the domain Ω does not need to be smooth for our weak formulation (4.65) to make sense. It only needs to be piecewise smooth and continuous. This is true for polygonal domains and for domains with piecewise quadratic boundary. We make use of this when we discretize with finite elements.

4.3.3 Well-Posedness of the Time-Discrete Problem

We can guarantee the well-posedness of (4.65) if all of the following properties are true (see [22], [129], [80]):

- the bilinear form $a(\vec{u},\vec{v})$ is coercive over all of $\mathbb{V}$, i.e.

  $$a(\vec{v},\vec{v}) \ge c\|\vec{v}\|_{\mathbb{V}}^2,$$

  for some fixed constant $c > 0$.

- the bilinear form $b(\vec{v},(q,\mu))$ satisfies the INF-SUP condition.

- the convex set Λ is closed.

The condition on $a(\cdot,\cdot)$ can be relaxed to just being coercive over the kernel of b. However, it is convenient to take a to be coercive over the whole space. This makes the proof of the well-posedness of the fully discrete problem easier (Section 4.3.4) and guarantees the convergence of our iterative algorithm for solving (4.65) with the inequality constraint (Section 4.3.6). In the following sections, we prove the first two items in the list. We showed previously that the convex set was closed. Note that our convex set is not a convex cone, which is the usual assumption made in some of the literature.

An Alternate Norm on $\mathbb{V}$

Let $\|\vec{u}\|^2_{\text{alt}\mathbb{V}} := \int_\Omega \vec{u}\cdot\vec{v} + \int_\Omega(\nabla\cdot\vec{u})(\nabla\cdot\vec{v}) + \int_\Gamma \nabla_\Gamma\vec{u}\cdot\nabla_\Gamma\vec{v}$. We show that $\|\cdot\|_{\text{alt}\mathbb{V}}$ is a norm on $\mathbb{V}$. The only norm property that needs to be checked is that if $\|\vec{u}\|_{\text{alt}\mathbb{V}} = 0$, then $\mathbb{V} \ni \vec{u} = 0$. So, let $\vec{u} \in \mathbb{V}$, and let $\|\vec{u}\|_{\text{alt}\mathbb{V}} = 0$. Then $\|\nabla_\Gamma\vec{u}\|_{0,\Gamma} = 0$, which implies $\vec{u}|_\Gamma = \vec{u}_0$ (constant vector) in $L^2(\Gamma)$. By standard Sobolev embedding [93], [1], this gives

$$\|(\vec{u}-\vec{u}_0)\cdot\vec{n}\|_{-1/2,\Gamma} \leq c\|\vec{u}-\vec{u}_0\|_{0,\Gamma} = 0. \tag{4.66}$$

Next, because $\vec{u}$ is in $H^1(\Gamma)$, the $H^{-1/2}(\Gamma)$ norm of $\vec{u}$ can be written as

$$\|\vec{u}\cdot\vec{n}\|_{-1/2,\Gamma} = \|\vec{u}_0\cdot\vec{n}\|_{-1/2,\Gamma} = \sup_{w\in H^1(\Omega)} \frac{<\vec{u}_0\cdot\vec{n}, w>_{1/2}}{\|w\|_{1,\Omega}} = \sup_{w\in H^1(\Omega)} \frac{\int_\Gamma(\vec{u}_0\cdot\vec{n})w}{\|w\|_{1,\Omega}},$$

where we used (4.66), $<,>_{1/2}$ is the duality pairing between $H^{-1/2}(\Gamma)$ and $H^{1/2}(\Gamma)$, and the pairing has been replaced by an integral because of the regularity of $\vec{u}_0$ and w. Now note that there exists a non-empty subset Γ_0 of Γ such that $\vec{u}_0\cdot\vec{n} \geq c_1|\vec{u}_0|$

on Γ_0. And also note that we can find a w_0 in $H^1(\Omega)$ such that w_0 is smooth, $w_0 > 0$ on Γ_0, and $w_0 = 0$ on $\Gamma \setminus \Gamma_0$. Therefore, because $0 = \|\vec{u}\|_{H(div,\Omega)} \geq \|\vec{u} \cdot \vec{n}\|_{-1/2,\Gamma}$ (by hypothesis), we get

$$0 \geq \sup_{w \in H^1(\Omega)} \frac{\int_\Gamma (\vec{u}_0 \cdot \vec{n}) w}{\|w\|_{1,\Omega}} \geq \frac{c_1 |\vec{u}_0| \int_{\Gamma_0} w_0}{\|w_0\|_{1,\Omega}} \geq c_2 |\vec{u}_0|,$$

where $c_2 > 0$. Hence, $\vec{u}_0 = 0$, which means that $\vec{u} = 0$ in $\mathbb{V}$. So, $\|\cdot\|_{\mathrm{alt}\mathbb{V}}$ is a norm on $\mathbb{V}$. Specifically, $\|\cdot\|_{\mathrm{alt}\mathbb{V}}$ is equivalent to the $\mathbb{V}$ norm defined previously (4.60). In other words,

$$\|\vec{u}\|_{\mathbb{V}} \geq \|\vec{u}\|_{\mathrm{alt}\mathbb{V}} \geq c_{\mathbb{V}} \|\vec{u}\|_{\mathbb{V}}, \tag{4.67}$$

where $c_{\mathbb{V}} > 0$ is a constant that only depends on Ω.

Coercivity of $a(\cdot, \cdot)$

The coercivity follows from the previous result. Since $a(\vec{u}, \vec{u}) \geq \triangle t \|\vec{u}\|^2_{\mathrm{alt}\mathbb{V}}$ (the constants α and β are typically larger than 1), we get that

$$a(\vec{u}, \vec{u}) \geq \triangle t \|\vec{u}\|^2_{\mathrm{alt}\mathbb{V}} \geq \triangle t \, c_{\mathbb{V}}^2 \|\vec{u}\|^2_{\mathbb{V}}. \tag{4.68}$$

So the coercivity constant depends on the time-step $\triangle t$.

INF-SUP Proof

The ‘INF-SUP’ condition has many forms, all of which are equivalent [21], [18]. We use the following form of it:

$$\sup_{\vec{v} \in \mathbb{V}} \frac{b(\vec{v}, (q, \mu))}{\|\vec{v}\|_{\mathbb{V}}} \geq \beta_0 (\|q\|^2_{\mathbb{P}} + \|\mu\|^2_{\mathbb{M}})^{1/2}, \quad \text{for all } (q, \mu) \in \mathbb{P} \times \mathbb{M}, \tag{4.69}$$

for some constant β_0 strictly greater than zero. The idea of the proof is that we construct a $\vec{v}$ in $\mathbb{V}$ that gives us the inequality, then take the supremum (sup) over all of $\mathbb{V}$. The proof is broken up into the following steps:

STEP 0:

Fix q in $\mathbb{P}$ and μ in $\mathbb{M}$. We will construct a velocity $\vec{v}$ that will give us (4.69).

STEP 1:

First, we construct a vector function $\vec{v}_0 \in [H^1(\Gamma)]^2$ such that

$$\|\mu\|_{\mathbb{M}} = <\mu, \vec{v}_0 \cdot \vec{n}>, \quad \|\vec{v}_0\|_{1,\Gamma} = 1. \tag{4.70}$$

By the definition of the $\mathbb{M}$ norm (4.54), there exists an η_0 in G such that $\|\eta_0\|_G = 1$ and $\|\mu\|_{\mathbb{M}} = <\mu, \eta_0>$. Note that adding a constant to η_0 would not affect the numerator of (4.54) (because μ has mean value zero, i.e. $<\mu, 1> = 0$), whereas the denominator would change. Hence, to attain the supremum in (4.54), η_0 must have mean value zero to minimize the denominator. Likewise, by the definition of the G norm (4.52), there exists a $\vec{v}_0 \in [H^1(\Gamma)]^2$ such that $\|\vec{v}_0\|_{1,\Gamma} = \|\eta_0\|_G = 1$ and $\eta_0 = \vec{v}_0 \cdot \vec{n}$. Ergo, $\vec{v}_0$ satisfies (4.70) and $\vec{v}_0 \cdot \vec{n}$ has mean value zero.

STEP 2:

Next, we use Proposition B.0.1 in Appendix B. Let $\vec{v} \in [H^1(\Omega)]^2$ satisfy the following problem:

$$\begin{aligned} \nabla \cdot \vec{v} &= -q, \quad \Omega, \\ \vec{v} &= \|\mu\|_{\mathbb{M}} \vec{v}_0 + \alpha \vec{n}_s, \quad \Gamma, \end{aligned} \tag{4.71}$$

where α is a constant to guarantee compatibility and is given by

$$\alpha = \frac{1}{|\Gamma|} \int_{\Omega} -q,$$

and the vector $\vec{n}_s$ is described in Appendix C.

Now, note the following inequalities:

$$|\alpha| \leq \frac{1}{|\Gamma|}\int_\Omega |q| \leq \frac{|\Omega|^{1/2}}{|\Gamma|}\|q\|_{\mathbb{P}} \leq \frac{1}{2\sqrt{\pi}}\|q\|_{\mathbb{P}},$$

$$\|\vec{v}\|_{1,\Gamma} \leq \|\mu\|_{\mathbb{M}}\|\vec{v}_0\|_{1,\Gamma} + |\alpha|\|\vec{n}_s\|_{1,\Gamma}.$$

Using these inequalities, $\vec{v}_0$ from step 1, the construction of $\vec{n}_s$ (given in Appendix C), and Proposition B.0.1, we get that the solution to (4.71) satisfies these bounds

$$\|\vec{v}\|_{H(\mathrm{div},\Omega)} \leq c_2\|\vec{v}\|_{1,\Omega} \leq c_3(\|q\|_{\mathbb{P}} + \|\vec{v}\|_{1/2,\Gamma}) \leq c_4(\|q\|_{\mathbb{P}} + \|\mu\|_{\mathbb{M}}). \tag{4.72}$$

$$\|\vec{v}\|_{1,\Gamma} \leq c_1(\|q\|_{\mathbb{P}} + \|\mu\|_{\mathbb{M}}), \tag{4.73}$$

Note that $\vec{v}$ is also contained in $\mathbb{V}$ by (4.27).

STEP 3:

Now, insert the function $\vec{v}$ from step 2 into the bilinear form b

$$\frac{b(\vec{v},(q,\mu))}{\|\vec{v}\|_{\mathbb{V}}} = \frac{-\int_\Omega q\nabla\cdot\vec{v} + < \mu, \vec{v}\cdot\vec{n} >}{\|\vec{v}\|_{\mathbb{V}}} = \frac{\int_\Omega q^2 + \|\mu\|_{\mathbb{M}} < \mu, \vec{v}_0\cdot\vec{n} > + \alpha < \mu, 1 >}{\|\vec{v}\|_{\mathbb{V}}}. \tag{4.74}$$

By step 1, we get

$$\frac{b(\vec{v},(q,\mu))}{\|\vec{v}\|_{\mathbb{V}}} = \frac{\|q\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2}{\|\vec{v}\|_{\mathbb{V}}}. \tag{4.75}$$

STEP 4:

Finally, using the bounds given in (4.72) and (4.73) and taking the supremum over all of $\mathbb{V}$, we get

$$\sup_{\vec{v}\in\mathbb{V}} \frac{b(\vec{v},(q,\mu))}{\|\vec{v}\|_{\mathbb{V}}} \geq \beta_0(\|q\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2)^{1/2} \tag{4.76}$$

for some fixed constant $\beta_0 > 0$. Since q and μ were arbitrary, this proves the continuous INF-SUP condition.

This proof holds for domains Ω that are not just smooth but rather piecewise smooth. This is true for polygonal domains and domains with curved sides. The only part that changes is the particular choice of $\vec{n}_s$ (see Appendix C).

4.3.4 Finite Element Discretization

One must be careful when choosing how to discretize the function spaces in a mixed variational formulation (i.e. which finite element spaces to use). Choosing an 'obvious' or straightforward discretization can adversely affect the stability of the method (i.e. its well-posedness) as well as the order of convergence. The key to ensuring a stable method is to ensure that the same criteria listed in Section 4.3.3 is true when using finite dimensional approximations of the usual function spaces.

First, we partition the domain Ω (and Γ) into a set of triangles (and sides) (see Figure 4.2). Denote this set of triangles by $\top_\Omega$ and the set of sides of Γ by S_Γ. This means the domain Ω has a polygonal boundary Γ or, if curved triangles are used on the boundary, the boundary sides are piecewise quadratic (see Figure 4.4). If piecewise linear functions are used to approximate the velocity, then a polygonal representation of the boundary is adequate. In our case, we use piecewise quadratic functions to approximate the velocity in order to have a stable method. Thus, we approximate the domain with curved triangles.

Next, we state the polynomial spaces that will be used in approximating $\mathbb{V}$, $\mathbb{P}$, and $\mathbb{M}$. Let $\mathcal{P}_k(D)$ be the space of polynomials of order k on the sub-domain D.

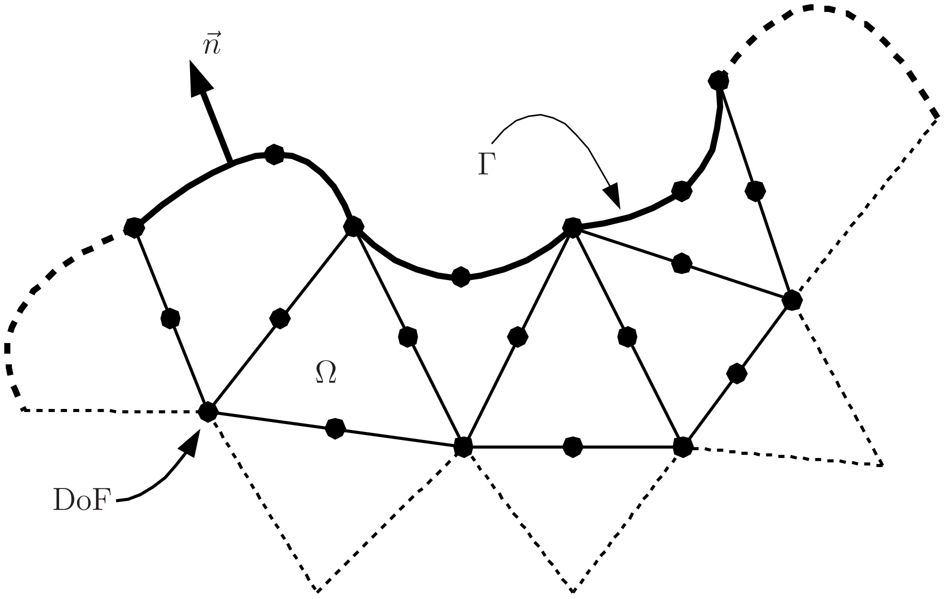


Figure 4.4: A triangulated domain with curved triangles on the boundary. The boundary Γ of the domain Ω is composed of curved sides that are piecewise quadratic and continuous. The 'dots' mark the degrees-of-freedom (DoF) of the quadratic polynomials that are defined on each triangle. Quadratic sides are required to compute curvature accurately when using piecewise quadratic polynomial functions to approximate the velocity field [128].

Define

$$V_k := \{\vec{u} = (u, v) \in [C(\bar{\Omega}) \cap H^1(\Omega)]^2 : u, v|_T \in \mathcal{P}_k(T),\ \text{for } T \in \top_\Omega\} \tag{4.77}$$

to be a space of piecewise vector polynomials of order k, and define

$$Q_k := \{q \in C(\Omega) \cap L^2(\Omega) : q|_T \in \mathcal{P}_k(T),\ \text{for } T \in \top_\Omega\} \tag{4.78}$$

to be a space of piecewise scalar polynomials of order k. We also need a space of piecewise discontinuous polynomial functions on the boundary Γ:

$$M_k := \{\mu \in L_0^2(\Gamma) : q|_S \in \mathcal{P}_k(S),\ \text{for } S \in \mathsf{S}_\Gamma\}, \tag{4.79}$$

where $L_0^2(\Gamma)$ is the space of L^2 functions on Γ that have *mean value zero.*

Let $\mathbb{V}_h \subset \mathbb{V}$, $\mathbb{P}_h \subset \mathbb{P}$, and $\mathbb{M}_h \subset \mathbb{M}$ be conforming finite dimensional subspaces defined by:

$$\mathbb{V}_h := V_2, \qquad \mathbb{P}_h := Q_1, \qquad \mathbb{M}_h := M_0, \tag{4.80}$$

where h is a discretization parameter that refers to the maximum size of triangles in $\top_\Omega$. We also need a discrete version of the convex set defined in (4.55). This is given by

$$\Lambda_h := \{\mu \in \mathbb{M}_h : -P_{pin} \le \mu|_S \le +P_{pin},\ \text{for each } S \in \mathsf{S}_\Gamma\}, \tag{4.81}$$

and is conforming: $\Lambda_h \subset \Lambda$.

The variational formulation of the discrete problem then reads: Find $(\vec{u}, p, \lambda) \in \mathbb{V}_h \times \mathbb{P}_h \times \Lambda_h$ such that

$$\begin{aligned} a(\vec{u}, \vec{v}) + b(\vec{v}, (p, \lambda)) &= \chi(\vec{v}), && \text{for all } \vec{v} \in \mathbb{V}_h, \\ b(\vec{u}, (q, \mu - \lambda)) &\le 0, && \text{for all } (q, \mu) \in \mathbb{P}_h \times \Lambda_h. \end{aligned} \tag{4.82}$$

Note the very similar structure of (4.82) to the continuous case (4.65). The only difference is that we have chosen finite-dimensional spaces $(\mathbb{V}_h, \mathbb{P}_h, \Lambda_h)$ to replace the continuous spaces $(\mathbb{V}, \mathbb{P}, \Lambda)$.

4.3.5 Well-Posedness of the Fully Discrete Problem

The well-posedness of (4.82) follows by the same criteria as for the continuous case. Because the finite element spaces are conforming to the infinite dimensional spaces, the only criteria to check is that the INF-SUP condition is still satisfied [21], [18].

We split the proof of the discrete INF-SUP condition into two stages. First, we prove it for the triple, (V_2, Q_0, M_0), (i.e. for piecewise continuous and quadratic velocity, piecewise constant pressure, and piecewise constant pinning variable). Then, we prove it for the triple, (V_2, Q_1, M_0), (i.e. the spaces we chose in the formulation (4.82)).

In both cases, we make use of Fortin's criteria [21], [18], which uses the continuous INF-SUP condition to prove the discrete version:

Proposition 4.3.1. (Fortin's Criteria) *Suppose that the bilinear form $b : \mathbb{V} \times (\mathbb{P} \times \mathbb{M}) \to \mathbb{R}$ satisfies the INF-SUP condition. In addition, suppose that for the conforming subspaces $\mathbb{V}_h, \mathbb{P}_h, \mathbb{M}_h$, there exists a bounded linear projector $\Pi : \mathbb{V} \to \mathbb{V}_h$ such that*

$$b(\vec{v} - \Pi\vec{v}, (q, \mu)) = 0, \text{ for all } (q, \mu) \in \mathbb{P}_h \times \mathbb{M}_h. \tag{4.83}$$

If $\|\Pi\vec{v}\|_{\mathbb{V}} \leq c\|\vec{v}\|_{\mathbb{V}}$ for some constant independent of h, then the finite element spaces

$\mathbb{V}_h$ *and* $(\mathbb{P}_h \times \mathbb{M}_h)$ *satisfy the INF-SUP condition.*

Proof. Let $\mu \in \mathbb{M}_h \subset \mathbb{M}$ and let $q \in \mathbb{P}_h \subset \mathbb{P}$. By the assumption that the continuous INF-SUP is satisfied, we have

$$\begin{aligned}\beta_0(\|q\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2)^{1/2} &\leq \sup_{\vec{v}\in\mathbb{V}} \frac{b(\vec{v},(q,\mu))}{\|\vec{v}\|_{\mathbb{V}}} = \sup_{\vec{v}\in\mathbb{V}} \frac{b(\Pi\vec{v},(q,\mu))}{\|\vec{v}\|_{\mathbb{V}}} \leq \\ &\leq c \sup_{\vec{v}\in\mathbb{V}} \frac{b(\Pi\vec{v},(q,\mu))}{\|\Pi\vec{v}\|_{\mathbb{V}}} = c \sup_{\vec{v}\in\mathbb{V}_h} \frac{b(\vec{v},(q,\mu))}{\|\vec{v}\|_{\mathbb{V}}},\end{aligned}$$

which is the discrete INF-SUP condition, with constant β_0/c. □

The projector Π is called a Fortin interpolation operator. The goal next is to construct a suitable Fortin interpolator.

Discrete INF-SUP with (V_2, Q_0, M_0)

Again we proceed in steps. Note that the duality pairing for the pinning variable can now be written as an integral because μ is piecewise constant in $L_0^2(\Gamma)$.

STEP 0:

Assume that $\vec{v} \in [H^1(\Omega)]^2 \cap [H^1(\Gamma)]^2 \subset \mathbb{V}$.

STEP 1:

Define the $V_2 - Q_0$ projection $\Pi_2\vec{v}|_T$ on each triangle T in the following way:

$$\begin{aligned} &\Pi_2\vec{v}|_T \in \mathcal{P}_2(T), \\ &\Pi_2\vec{v}|_T(\vec{x}) = 0, \text{ for all } \vec{x} = \text{vertex of } T, \\ &\int_S (\vec{v} - \Pi_2\vec{v}) \cdot \vec{n} = 0, \text{ for all } S \subset \partial T, \\ &\int_S (\vec{v} - \Pi_2\vec{v}) \cdot \vec{t} = 0, \text{ for all } S \subset \partial T, \end{aligned} \tag{4.84}$$

where $\vec{n}$ and $\vec{t}$ are the normal and tangent vectors of the triangle side S. In the case of a triangle that is not curved, the last two conditions in (4.84) would simplify to

$$\int_S (\vec{v} - \Pi_2 \vec{v}) = 0, \text{ for all } S \subset \partial T. \tag{4.85}$$

However, for iso-parametric elements (i.e. curved triangles on the boundary), we need the other two conditions.

STEP 2:

Now verify that the following is true

$$b(\vec{v} - \Pi_2\vec{v}, (q, \mu)) = 0, \text{ for all } \vec{v} \in [H^1(\Omega)]^2 \cap [H^1(\Gamma)]^2 \tag{4.86}$$

and for all $q \in Q_0$ and $\mu \in M_0$. We do this by breaking up the integrals over individual triangles and sides:

$$\begin{aligned} b(\vec{v} - \Pi_2\vec{v}, (q, \mu)) &= -\int_\Omega q\nabla \cdot (\vec{v} - \Pi_2\vec{v}) + \int_\Gamma \mu(\vec{v} - \Pi_2\vec{v}) \cdot \vec{n} = \\ &= -\sum_{T \in \mathsf{T}_\Omega} q|_T \int_T \nabla \cdot (\vec{v} - \Pi_2\vec{v}) + \sum_{S \in \mathsf{S}_\Gamma} \mu|_S \int_S (\vec{v} - \Pi_2\vec{v}) \cdot \vec{n} \\ &\overset{(4.84)}{=} -\sum_{T \in \mathsf{T}_\Omega} q|_T \sum_{S \subset \partial T} \int_S (\vec{v} - \Pi_2\vec{v}) \cdot \vec{n} \overset{(4.84)}{=} 0, \end{aligned}$$

where the divergence theorem was used in translating integrals over triangles to integrals over sides.

STEP 3:

We need the following estimates for the Π_2 interpolant (taken from [21]):

$$\begin{aligned} |\Pi_2\vec{v}|_{1,T} = |\widehat{\Pi_2\vec{v}}|_{1,\hat{T}} \le c\|\hat{\vec{v}}\|_{1,\hat{T}} \le c(h_T^{-1}|\vec{v}|_{0,T} + |\vec{v}|_{1,T}), \\ \Longrightarrow \|\Pi_2\vec{v}\|_{1,T} \le c(h_T^{-1}\|\vec{v}\|_{0,T} + |\vec{v}|_{1,T}), \end{aligned} \tag{4.87}$$

where h_T is the size of triangle T and $\hat{T}$ is the standard reference triangle [18], [20]. The next estimate follows by an inverse inequality (see equation (4.5.2) on page 111 of [20]),

$$\|\Pi_2\vec{v}\|_{1,S} \leq c h_S^{-1} \|\vec{v}\|_{0,S}, \tag{4.88}$$

where h_S is the length of a side $S \subset \Gamma$ of a triangle.

STEP 4:

Let Π_1 be the Clément [30] or Scott-Zhang interpolant [122] (onto vector piecewise linear polynomials, V_1), which has the following properties:

$$\sum_{T \in \mathsf{T}_\Omega} h_T^{-2} |\vec{v} - \Pi_1\vec{v}|_{0,T}^2 \leq c\|\vec{v}\|_{1,\Omega}^2, \tag{4.89}$$

$$\sum_{T \in \mathsf{T}_\Omega} |\vec{v} - \Pi_1\vec{v}|_{1,T}^2 \leq c\|\vec{v}\|_{1,\Omega}^2, \tag{4.90}$$

$$\sum_{S \in \mathsf{S}_\Gamma} h_S^{-2} |\vec{v} - \Pi_1\vec{v}|_{0,S}^2 \leq c|\vec{v}|_{1,\Gamma}^2, \tag{4.91}$$

$$\|\Pi_1\vec{v}\|_{1,\Gamma} \leq c\|\vec{v}\|_{1,\Gamma}, \tag{4.92}$$

where (4.91) and (4.92) follows from the properties of the Scott-Zhang interpolant, or by setting the Clément interpolant on the boundary to the nodal interpolant of the $H^1(\Gamma)$ function, which has point-wise values because Γ is a 1-D curve (i.e. $H^1(\Gamma) \subset C^0(\Gamma)$ when Γ is 1-D [47], [93], [1]).

STEP 5:

Now we derive two intermediate inequalities:

$$\begin{aligned} \|\Pi_2(I - \Pi_1)\vec{v}\|_{1,\Omega}^2 &= \sum_{T \in \mathsf{T}_\Omega} \|\Pi_2(I - \Pi_1)\vec{v}\|_{1,T}^2 \overset{(4.87)}{\leq} \\ \leq c_1 \sum_{T \in \mathsf{T}_\Omega} \{h_T^{-2}|(I - \Pi_1)\vec{v}|_{0,T}^2 &+ |(I - \Pi_1)\vec{v}|_{1,T}^2\} \overset{(4.89),\,(4.90)}{\leq} c_2\|\vec{v}\|_{1,\Omega}^2, \end{aligned} \tag{4.93}$$

$$\begin{aligned}\|\Pi_2(I-\Pi_1)\vec{v}\|_{1,\Gamma}^2 = \sum_{S\in\mathsf{S}_\Gamma} \|\Pi_2(I-\Pi_1)\vec{v}\|_{1,S}^2 &\overset{(4.88)}{\leq} \\ \leq c_1 \sum_{S\in\mathsf{S}_\Gamma} h_S^{-2}|(I-\Pi_1)\vec{v}|_{0,S}^2 &\overset{(4.91)}{\leq} c_2\|\vec{v}\|_{1,\Gamma}^2,\end{aligned} \tag{4.94}$$

STEP 6:

We define the main (Fortin) interpolant:

$$\Pi_h\vec{v} := \Pi_1\vec{v} + \Pi_2(\vec{v}-\Pi_1\vec{v}), \tag{4.95}$$

and we can derive the following estimates, starting with the $[H^1(\Omega)]^2$ norm,

$$\begin{aligned}\|\Pi_h\vec{v}\|_{1,\Omega} \leq \|\Pi_1\vec{v}\|_{1,\Omega} + \|\Pi_2(\vec{v}-\Pi_1\vec{v})\|_{1,\Omega} &\overset{(4.93)}{\leq} \\ \leq \|\Pi_1\vec{v}-\vec{v}\|_{1,\Omega} + \|\vec{v}\|_{1,\Omega} + c_1\|\vec{v}\|_{1,\Omega} &\overset{(4.89),\,(4.90)}{\leq} c_2\|\vec{v}\|_{1,\Omega}\end{aligned} \tag{4.96}$$

and also for the $[H^1(\Gamma)]^2$ norm,

$$\begin{aligned}\|\Pi_h\vec{v}\|_{1,\Gamma} \leq \|\Pi_1\vec{v}\|_{1,\Gamma} + \|\Pi_2(\vec{v}-\Pi_1\vec{v})\|_{1,\Gamma} &\overset{(4.94)}{\leq} \\ \leq \|\Pi_1\vec{v}\|_{1,\Gamma} + c_1\|\vec{v}\|_{1,\Gamma} &\overset{(4.92)}{\leq} c_2\|\vec{v}\|_{1,\Gamma}\end{aligned} \tag{4.97}$$

Hence, we get that the Fortin operator (4.95) is bounded in the $[H^1(\Omega)]^2 \cap [H^1(\Gamma)]^2$ norm,

$$\|\Pi_h\vec{v}\|_{1,\Omega} + \|\Pi_h\vec{v}\|_{1,\Gamma} \leq c(\|\vec{v}\|_{1,\Omega} + \|\vec{v}\|_{1,\Gamma}). \tag{4.98}$$

And we also have that Π_h satisfies (4.83),

$$\begin{aligned} b(\vec{v}-\Pi_h\vec{v},(q,\mu)) = 0, \text{ for all } \vec{v} \in [H^1(\Omega)]^2 \cap [H^1(\Gamma)]^2, \\ \text{and all } q \in Q_0, \mu \in M_0,\end{aligned} \tag{4.99}$$

because of (4.86) and the fact that $\vec{v}-\Pi_h\vec{v} = (\vec{v}-\Pi_1\vec{v}) - \Pi_2(\vec{v}-\Pi_1\vec{v}) = \vec{w}-\Pi_2\vec{w}$, where $[H^1(\Omega)]^2 \cap [H^1(\Gamma)]^2 \ni \vec{w} = \vec{v}-\Pi_1\vec{v}$.

STEP 7:

Next, we extend the Fortin operator to all of $\mathbb{V}$ (recall definition (4.25)) using Proposition B.0.1. To do this, first let $\vec{w} \in \mathbb{V}$ and note that $\nabla \cdot \vec{w} \in L^2(\Omega)$ and $\vec{w} \in H^1(\Gamma)$ by the inclusion relation (4.27). Moreover, we have the following compatibility condition

$$\int_\Omega \nabla \cdot \vec{w} = \int_\Omega \nabla \cdot \vec{w}_{int} = \int_\Gamma \vec{w}_{int} \cdot \vec{n} = \int_\Gamma \vec{w} \cdot \vec{n},$$

where we use the decoupled velocity notation from Section 4.2.3. Hence, we can let $\vec{v} \in [H^1(\Omega)]^2 \cap [H^1(\Gamma)]^2$ satisfy the following divergence problem:

$$\begin{aligned} \nabla \cdot \vec{v} &= \nabla \cdot \vec{w}, \quad \Omega, \\ \vec{v} &= \vec{w}, \qquad\quad \Gamma, \end{aligned}$$

where $\vec{v}$ satisfies the bound: $\|\vec{v}\|_{1,\Omega} \leq c(\|\vec{w}\|_{1,\Gamma} + \|\nabla \cdot \vec{w}\|_{0,\Omega})$. Then define a new operator $\Pi : \mathbb{V} \to \mathbb{V}_h$, where $\mathbb{V}_h$ is the space of (vector) piecewise continuous quadratic polynomials, in the following way: $\Pi\vec{w} = \Pi_h\vec{v}$ (note: $\vec{v}$ depends on $\vec{w}$).

STEP 8:

To prove that this new operator satisfies the Fortin criteria, we must first verify that it is bounded in the $\mathbb{V}$ norm. This is done using properties of the solution of the problem in the previous step,

$$\begin{aligned} \|\Pi\vec{w}\|_{\mathbb{V}} \leq c_1(\|\Pi_h\vec{v}\|_{1,\Omega} + \|\Pi_h\vec{v}\|_{1,\Gamma}) \overset{(4.98)}{\leq} c_2(\|\vec{v}\|_{1,\Omega} + \|\vec{v}\|_{1,\Gamma}) \leq \\ \leq c_3(\|\nabla \cdot \vec{w}\|_{0,\Omega} + \|\vec{w}\|_{1,\Gamma}) \leq c_3(\|\vec{w}\|_{H(\mathrm{div})} + \|\vec{w}\|_{1,\Gamma}) \leq c_4\|\vec{w}\|_{\mathbb{V}} \end{aligned} \tag{4.100}$$

STEP 9:

Finally, we verify that Π satisfies (4.83),

$$\begin{aligned} b(\vec{w}-\Pi\vec{w},(q,\mu)) &= -\int_\Omega q\nabla\cdot(\vec{w}-\Pi\vec{w}) + \int_\Gamma \mu(\vec{w}-\Pi\vec{w})\cdot\vec{n} = \\ &= -\int_\Omega q\nabla\cdot(\vec{v}-\Pi_h\vec{v}) + \int_\Gamma \mu(\vec{v}-\Pi_h\vec{v})\cdot\vec{n} = \\ &= b(\vec{v}-\Pi_h\vec{v},(q,\mu)) \overset{(4.99)}{=} 0, \end{aligned} \tag{4.101}$$

for all q in Q_0 and μ in M_0. The property (4.101) is true for any $\vec{w}\in\mathbb{V}$. Hence, by Fortin's criteria, the discrete INF-SUP condition is satisfied, using the Fortin interpolant Π.

Discrete INF-SUP with (V_2, Q_1, M_0)

Here we will use the previous section and the proof of the standard Taylor-Hood element to prove that the discrete INF-SUP condition holds for the triple (V_2, Q_1, M_0).

STEP 0:

Let $q \in Q_1$ (i.e. a piecewise linear continuous function over Ω) and let $\overline{q}$ be the L^2 projection of q onto piecewise constants. Also, let $\mu \in M_0$.

STEP 1:

Since the INF-SUP condition is true for (V_2, Q_0, M_0) (see previous section), there exists a $\vec{w}_0 \in \mathbb{V}_h$ such that

$$\frac{b(\vec{w}_0,(\overline{q},\mu))}{\|\vec{w}_0\|_{\mathbb{V}}} \geq \beta_0(\|\overline{q}\|^2_{\mathbb{P}} + \|\mu\|^2_{\mathbb{M}})^{1/2}.$$

By letting $\vec{w} := (\|\overline{q}\|^2_{\mathbb{P}} + \|\mu\|^2_{\mathbb{M}})^{1/2}\frac{\vec{w}_0}{\|\vec{w}_0\|_{\mathbb{V}}}$, we get the following useful property:

$$\begin{gathered} b(\vec{w},(\overline{q},\mu)) \geq \beta_0(\|\overline{q}\|^2_{\mathbb{P}} + \|\mu\|^2_{\mathbb{M}}), \\ \|\vec{w}\|_{\mathbb{V}} = (\|\overline{q}\|^2_{\mathbb{P}} + \|\mu\|^2_{\mathbb{M}})^{1/2}. \end{gathered} \tag{4.102}$$

Using (4.102), gives the following estimate

$$\begin{aligned} b(\vec{w},(q,\mu)) &= -\int_\Omega q\nabla\cdot\vec{w} + \int_\Gamma \mu(\vec{w}\cdot\vec{n}) = \\ &= -\int_\Omega \overline{q}\nabla\cdot\vec{w} + \int_\Gamma \mu(\vec{w}\cdot\vec{n}) - \int_\Omega (q-\overline{q})\nabla\cdot\vec{w} = \\ &= b(\vec{w},(\overline{q},\mu)) - \int_\Omega (q-\overline{q})\nabla\cdot\vec{w} \geq \\ &\geq \beta_0(\|\overline{q}\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2) - \|q-\overline{q}\|_{\mathbb{P}}(\|\overline{q}\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2)^{1/2}, \end{aligned} \tag{4.103}$$

where, in the last line, we used the Cauchy-Schwarz inequality and the fact that $\|\nabla\cdot\vec{w}\|_{0,\Omega} \leq \|\vec{w}\|_{\mathbb{V}}$

STEP 2:

We use some of the proof from the Taylor-Hood case. From [21], there exists a $\vec{z} \in V_2$ such that $\vec{z} = 0$ on Γ, $\|\vec{z}\|_{\mathbb{V}} \leq c_1\|\vec{z}\|_{1,\Omega} \leq c_2\|q-\overline{q}\|_{\mathbb{P}}$, and

$$-\int_\Omega q\nabla\cdot\vec{z} \geq c_3\|q-\overline{q}\|_{\mathbb{P}}^2.$$

STEP 3:

Let $\vec{v} = \vec{w} + \delta\vec{z}$, where $\delta > 0$ is to be chosen. Plugging into $b(\cdot,\cdot)$ gives

$$b(\vec{v},(q,\mu)) = b(\vec{w},(q,\mu)) - \delta\int_\Omega q\nabla\cdot\vec{z} \overset{(4.103),\,(\text{Step 2})}{\geq}$$

$$\geq \beta_0(\|\overline{q}\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2) - \|q-\overline{q}\|_{\mathbb{P}}(\|\overline{q}\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2)^{1/2} + \delta c_3\|q-\overline{q}\|_{\mathbb{P}}^2 \geq$$

$$\overset{(\text{Cauchy ineq.})}{\geq} \frac{\beta_0}{2}(\|\overline{q}\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2) + (\delta c_3 - \frac{1}{2\beta_0})\|q-\overline{q}\|_{\mathbb{P}}^2.$$

And then setting $\delta = \frac{1}{2\beta_0 c_3} + \frac{\beta_0}{2c_3}$ gives

$$b(\vec{v},(q,\mu)) \geq \frac{\beta_0}{2}(\|\overline{q}\|_{\mathbb{P}}^2 + \|q-\overline{q}\|_{\mathbb{P}}^2) + \frac{\beta_0}{2}\|\mu\|_{\mathbb{M}}^2 = \frac{\beta_0}{2}(\|q\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2), \tag{4.104}$$

where the equality follows by orthogonality in $L^2(\Omega)$.

STEP 4:

Finally, the bound for $\vec{v}$ is

$$\begin{aligned}\|\vec{v}\|_{\mathbb{V}}^2 \leq 2(\|\vec{w}\|_{\mathbb{V}}^2 + \delta^2\|\vec{z}\|_{\mathbb{V}}^2) \leq c_4^2(\|\overline{q}\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2 + \|q - \overline{q}\|_{\mathbb{P}}^2) = \\ = c_4^2(\|q\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2),\end{aligned} \tag{4.105}$$

and forming the quotient gives

$$\frac{b(\vec{v},(q,\mu))}{\|\vec{v}\|_{\mathbb{V}}} \geq \frac{\beta_0}{2c_4}(\|q\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2)^{1/2}.$$

Therefore, b satisfies the discrete INF-SUP condition for (V_2, Q_1, M_0):

$$\sup_{\vec{v}\in\mathbb{V}_h} \frac{b(\vec{v},(q,\mu))}{\|\vec{v}\|_{\mathbb{V}}} \geq \frac{\beta_0}{2c_4}(\|q\|_{\mathbb{P}}^2 + \|\mu\|_{\mathbb{M}}^2)^{1/2}. \tag{4.106}$$

4.3.6 Solving the Linear System with Inequality Constraint

Finally, given the inequality constraint (4.59), we still need a method of computing a solution in the discrete case. In this section, we present our method for solving the variational inequality for the case when $\mathbb{M}$ is approximated by discrete L^2 functions (i.e. $\mathbb{M}$ is replaced by $\mathbb{M}_h$, which is a space of piecewise constant functions). Note the duality pairing in (4.59) can now be written as an integral involving L^2 functions (we make use of this). The rest of the analysis is alla Glowinski [60], but applied to our problem.

For convenience, we rewrite the discrete variational equations with the duality pairing replaced by an integral. For the rest of this section, let $\vec{u} \in \mathbb{V}_h$, $p \in \mathbb{P}_h$, and

$\lambda \in \Lambda_h$ (the discrete convex set) be the true solution of the discrete problem

$$\begin{aligned}\left(\frac{\alpha}{\triangle t}+\beta\right)\left(\int_\Omega \vec{u}\cdot\vec{v}+\gamma\int_\Omega(\nabla\cdot\vec{u})(\nabla\cdot\vec{v})\right)+\triangle t\int_\Gamma \nabla_\Gamma\vec{u}\cdot\nabla_\Gamma\vec{v}+\\ -\int_\Omega p\nabla\cdot\vec{v}+\int_\Gamma \lambda\vec{v}\cdot\vec{n}=\frac{\alpha}{\triangle t}\int_\Omega \vec{u}^0\cdot\vec{v}-\int_\Gamma E\vec{v}\cdot\vec{n}+\\ -\int_\Gamma \nabla_\Gamma(\vec{X}\circ\vec{X}^{-1})\cdot\nabla_\Gamma\vec{v},\ \text{for all } \vec{v}\in\mathbb{V}_h,\end{aligned} \tag{4.107}$$

$$-\int_\Omega q\nabla\cdot\vec{u}=0,\ \text{for all } q\in\mathbb{P}_h, \tag{4.108}$$

$$\int_\Gamma(\vec{u}\cdot\vec{n})(\mu-\lambda)\leq 0,\ \text{for all } \mu\in\Lambda_h. \tag{4.109}$$

Iterative Solution Scheme (Uzawa Method)

We proceed to derive an Uzawa method, which basically takes advantage of an $L^2(\Gamma)$ projection property. Note that all functions in this derivation are discrete, hence all integrals are well-defined. The first step is to perform the following trick using the variational inequality (4.109):

$$0\geq\int_\Gamma \rho\vec{u}\cdot\vec{n}(\mu-\lambda)=\int_\Gamma[\lambda-(\lambda+\rho\vec{u}\cdot\vec{n})](\lambda-\mu)=\int_\Gamma(\lambda-\omega)(\lambda-\mu), \tag{4.110}$$

where $\rho > 0$ and $\omega := \lambda + \rho\vec{u}\cdot\vec{n}$. This implies that λ is the L^2 projection of ω onto Λ_h (i.e. $\lambda = P_{\Lambda_h}(\omega)$). This can be seen by Figure 4.5 or by minimizing the following functional over the convex set Λ_h:

$$J(\lambda)=\int_\Gamma|\lambda-\omega|^2=\min_{\mu\in\Lambda_h}\int_\Gamma|\mu-\omega|^2. \tag{4.111}$$

In other words, using the standard techniques of minimization leads to equation (4.110).

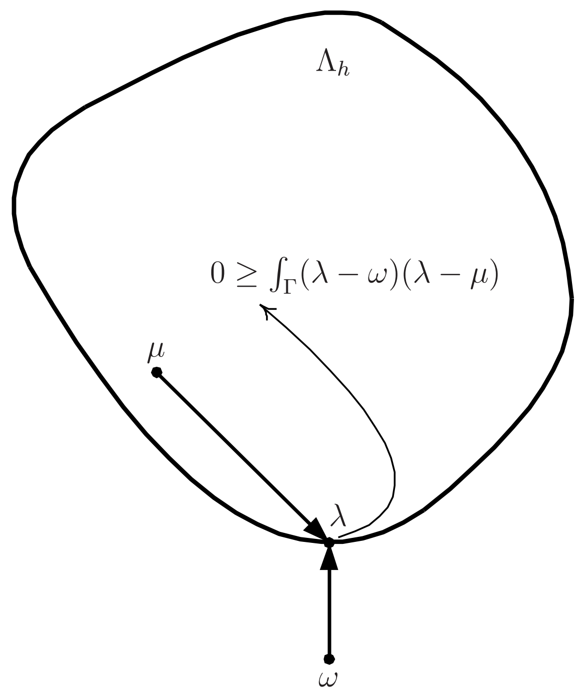


Figure 4.5: Projection onto a convex set. By equation (4.110), we know that the inner product of the functions $(\lambda - \omega)$ and $(\lambda - \mu)$ must be negative or zero for all μ in the convex set Λ_h (note: $\omega := \lambda + \rho\vec{u} \cdot \vec{n}$). This is only possible if λ is the $L^2(\Gamma)$ projection of ω onto Λ_h (i.e. $\lambda = P_{\Lambda_h}(\omega)$).

Therefore, the exact solution of the discrete problem must satisfy this L^2 projection property

$$\lambda = P_{\Lambda_h}(\lambda + \rho \vec{u} \cdot \vec{n}). \tag{4.112}$$

This suggests the following iterative method for solving the variational inequality:

$$\lambda^{j+1} = P_{\Lambda_h}(\lambda^j + \rho \vec{u}^j \cdot \vec{n}). \tag{4.113}$$

where λ^0 is an initial guess (say 0) and ρ acts as a relaxation parameter. The initial velocity $\vec{u}^0$ is obtained by plugging λ^0 into equation (4.107) and solving (4.107) and (4.108) using standard methods for saddle point problems. Then we apply the projection (4.113) to obtain λ^1 and plug it into (4.107) and solve for $\vec{u}^1$. We iterate this process several times until both sequences $\{\lambda^j\}$ and $\{\vec{u}^j\}$ converge.

In the actual implementation, we let $U_{\vec{n}}$ and $U_{\vec{n}}^j$ be the $L^2(\Gamma)$ projection of $\vec{u} \cdot \vec{n}$ and $\vec{u}^j \cdot \vec{n}$ onto $\mathbb{M}_h$, respectively. This allows $\vec{u} \cdot \vec{n}$ in (4.110) to be replaced by $U_{\vec{n}}$. So by the same argument as before, we get that the exact solution λ satisfies

$$\lambda = P_{\Lambda_h}(\lambda + \rho U_{\vec{n}}).$$

Therefore, we actually use the following update equation in our iterative algorithm

$$\lambda^{j+1} = P_{\Lambda_h}(\lambda^j + \rho U_{\vec{n}}^j). \tag{4.114}$$

The convergence proof using (4.114) is almost exactly the same as with (4.113), except one must also use the properties of the $L^2(\Gamma)$ projection with respect to $U_{\vec{n}}$ and $U_{\vec{n}}^j$. But the convergence of the solution variables $\vec{u}^j$, p^j, and λ^j are exactly the same, hence we only give the proof using (4.113).

Convergence Proof of the Uzawa Method

This proof is similar to Glowinski, except it is applied to a different problem. Let $\lambda^j_{err} = \lambda^j - \lambda$, $p^j_{err} = p^j - p$, and $\vec{u}^j_{err} = \vec{u}^j - \vec{u}$ denote the error between the iterate and the exact solution. Because the $L^2(\Gamma)$ projection, P_{Λ_h}, is an $L^2(\Gamma)$ norm decreasing map (follows by the minimization argument (4.111)), we have the following:

$$\|\lambda^{j+1}_{err}\|_{0,\Gamma} \leq \|\lambda^j_{err} + \rho \vec{u}^j_{err} \cdot \vec{n}\|_{0,\Gamma}. \tag{4.115}$$

Squaring both sides gives

$$\|\lambda^{j+1}_{err}\|^2_{0,\Gamma} \leq \|\lambda^j_{err}\|^2_{0,\Gamma} + 2\rho \int_\Gamma \lambda^j_{err}(\vec{u}^j_{err} \cdot \vec{n}) + \rho^2 \|\vec{u}^j_{err} \cdot \vec{n}\|^2_{0,\Gamma}, \tag{4.116}$$

which, by moving terms around, gives

$$\|\lambda^j_{err}\|^2_{0,\Gamma} - \|\lambda^{j+1}_{err}\|^2_{0,\Gamma} \geq \rho[-2\int_\Gamma \lambda^j_{err}(\vec{u}^j_{err} \cdot \vec{n}) - \rho\|\vec{u}^j_{err} \cdot \vec{n}\|^2_{0,\Gamma}]. \tag{4.117}$$

Now we need to take advantage of the PDE. Due to the linearity of the terms in (4.107) and (4.108), we have

$$\begin{aligned}\left(\frac{\alpha}{\triangle t} + \beta\right)\left(\int_\Omega \vec{u}^j_{err} \cdot \vec{v} + \gamma \int_\Omega (\nabla \cdot \vec{u}^j_{err})(\nabla \cdot \vec{v})\right) + \triangle t \int_\Gamma \nabla_\Gamma \vec{u}^j_{err} \cdot \nabla_\Gamma \vec{v} +& \\ - \int_\Omega p^j_{err} \nabla \cdot \vec{v} \ + \int_\Gamma \lambda^j_{err} \vec{v} \cdot \vec{n} = 0, \text{ for all } \vec{v} \in \mathbb{V}_h,& \quad (4.118)\\ - \int_\Omega q \nabla \cdot \vec{u}^j_{err} = 0, \text{ for all } q \in \mathbb{P}_h.&\end{aligned}$$

By setting the test functions $\vec{v} = \vec{u}^j_{err}$ and $q = p^j_{err}$ in (4.118), we get

$$\begin{aligned}-\int_\Gamma \lambda^j_{err}(\vec{u}^j_{err} \cdot \vec{n}) &= \left(\frac{\alpha}{\triangle t} + \beta\right)\left(\int_\Omega |\vec{u}^j_{err}|^2 + \gamma \int_\Omega |\nabla \cdot \vec{u}^j_{err}|^2\right) + \\ &\quad + \triangle t \int_\Gamma |\nabla_\Gamma \vec{u}^j_{err}|^2, \\ &= a(\vec{u}^j_{err}, \vec{u}^j_{err}),\end{aligned}$$

which implies that (because $a(\cdot,\cdot)$ is coercive)

$$-\int_{\Gamma} \lambda_{err}^{j} (\vec{u}_{err}^{j} \cdot \vec{n}) \geq c \triangle t \|\vec{u}_{err}^{j}\|_{\mathbb{V}}^{2}, \tag{4.119}$$

where $c > 0$ is a constant. Combining (4.119) with (4.117), and the fact that $\|\vec{u}_{err}^{j} \cdot \vec{n}\|_{0,\Gamma} \leq \|\vec{u}_{err}^{j}\|_{1,\Gamma} \leq \|\vec{u}_{err}^{j}\|_{\mathbb{V}}$, gives

$$\|\lambda_{err}^{j}\|_{0,\Gamma}^{2} - \|\lambda_{err}^{j+1}\|_{0,\Gamma}^{2} \geq \rho(2c\triangle t - \rho)\|\vec{u}_{err}^{j}\|_{\mathbb{V}}^{2}. \tag{4.120}$$

Hence, if $\rho < 2c\triangle t$, then $\{\|\lambda_{err}^{j}\|_{0,\Gamma}^{2}\}$ is a decreasing sequence of numbers, so it converges to some number (by the least upper bound axiom/Bolzano-Weierstrauss Theorem [85]). This means the right hand side of (4.120) goes to zero as j goes to ∞. Therefore,

$$\vec{u}^{j} \rightarrow \vec{u}, \qquad \text{as } j \rightarrow \infty. \tag{4.121}$$

In order to show that $\|\lambda_{err}^{j}\|_{\mathbb{M}}$ and $\|p_{err}^{j}\|_{\mathbb{P}}$ go to zero as $j \rightarrow \infty$, we must make use of the INF-SUP condition. Using (4.118) and the definition of the bilinear forms (4.62), (4.63) in (4.82), we have that

$$b(\vec{v},(p_{err}^{j},\lambda_{err}^{j})) = a(-\vec{u}_{err}^{j},\vec{v}), \qquad \text{for all } \vec{v} \in \mathbb{V}_{h}. \tag{4.122}$$

So, we just divide by the norm and take the supremum to get:

$$\sup_{\vec{v}\in\mathbb{V}_h} \frac{b(\vec{v},(p_{err}^{j},\lambda_{err}^{j}))}{\|\vec{v}\|_{\mathbb{V}}} = \sup_{\vec{v}\in\mathbb{V}_h} \frac{a(-\vec{u}_{err}^{j},\vec{v})}{\|\vec{v}\|_{\mathbb{V}}} \leq a_0\|\vec{u}_{err}^{j}\|_{\mathbb{V}}, \tag{4.123}$$

where $a_0 > 0$ is the continuity constant for $a(\cdot,\cdot)$. Using (4.106), we obtain

$$\|\vec{u}^{j} - \vec{u}\|_{\mathbb{V}} \geq \frac{\beta_0}{2a_0 c_4}(\|p^{j} - p\,\|_{\mathbb{P}}^{2} + \|\lambda^{j} - \lambda\|_{\mathbb{M}}^{2})^{1/2}. \tag{4.124}$$

So, by (4.121), we have that $p^{j} \rightarrow p$ and $\lambda^{j} \rightarrow \lambda$ in their respective norms. Note that the convergence of λ^{j} is in a negative norm (recall the dual space $\mathbb{M}$ (4.53)).

It is possible that there may be oscillations in the true discrete solution λ because we are using piecewise constants to approximate the continuous solution in $\mathbb{M}$. In practice, the behavior of λ is determined by the 'forcing' in the problem (i.e. the curvature κ and electrowetting force E). For the computations in this thesis, the behavior of λ seems appropriate.

4.4 Handling Large Deformations and Topological Changes

One of the drawbacks of using the explicit front tracking method is the presence of the mesh. Because the interface is moving, the underlying mesh must move with it. If the droplet is undergoing a large deformation, such as a splitting motion, this can cause mesh distortion (i.e. elongated triangles/elements in the droplet triangulation and/or inverted elements). It is known that distorted meshes can adversely affect the accuracy of the finite element solution [127]. Therefore, any explicit front tracking method must also be able to handle severe mesh distortion.

We handle large mesh deformations by using mesh smoothing and periodic re-meshing. Topological changes are handled using a hybrid variational-level set method and is described in [140]. We give some highlights of the algorithm in the following list. For more details, see [140].

- Mesh smoothing and re-meshing. We use standard techniques, such as optimization based mesh smoothing [45] and re-meshing with the program 'Triangle' [126].

- Updating mesh topology. We use a level set method to guide the evolution of the explicit finite element mesh through a topological change.

- Mesh reconstruction after the topological change. We use an active contour-based minimization approach to adjust the mesh in the local region of the topological change.

Chapter 5

Simulations Versus Experiments

We present comparisons between our simulation method and five different experiments. In each section, we describe the experimental setup and the corresponding simulation results. In the first experiment (splitting water droplet), we discuss the various physical phenomena affecting the motion of the droplet (i.e. saturation and hysteresis) and how the simulations were modified to capture these effects. Simulations for the first two experiments were run using the level set method (Chapter 3) and variational method (Chapter 4). The remaining experiments are compared to the variational method only.

5.1 A Splitting Water Droplet

In Figure 5.1, an overhead view of an EWOD device with three electrodes running left to right is depicted with a splitting droplet. The voltage actuation, from left to right, is 25 volts, 0 volts, 25 volts and is constant throughout the split. In the first frame, an initial near-circular droplet is shown just before voltage activation. After the voltage is turned on, the liquid-gas interface over the left and right electrodes deforms and induces a low pressure region there. The regions where no voltage is activated remain at high pressure. In the subsequent frames, the droplet is pulled from the left and right sides, while it is pushed in from the top and bottom. The droplet elongates along the horizontal dimension and is being pinched in the vertical direction. This causes two smaller droplets to form on the left and right sides, with a thin neck joining them. The neck eventually gets so thin that it snaps due to capillary instability. The two smaller droplets then continue moving to the left and right electrodes because of the pressure differential created from the voltage actuation. Finally, the two droplets come to rest on the two 25 volt electrodes. The total time of this experiment is approximately 167 milliseconds.

Next, we present three simulations showing the effects of the various physics at the boundary. The first simulation is for contact angle variations obeying the ideal Young-Lippmann curve (see the Y-L curve in Figure 2.6). Next, we simulate droplet splitting assuming only contact angle saturation (see the saturation curve for two plates in Figure 2.6). Finally, we show a simulation that includes saturation and hysteresis. In Table 5.1, we list the pertinent parameters of each simulation for the level set method; Table 5.2 lists the parameters for the variational method. In all cases, the velocity scale U_0 is chosen so that the non-dimensional velocities during the simulation are order one.

5.1.1 Ideal Young-Lippmann

In Figures 5.2 and 5.3, we have a simulation (level set) of droplet motion when no contact angle saturation or hysteresis is being modeled. As can be seen, the general shape of the splitting droplet is not the same as in the experiment. Just after the voltage is turned on, the droplet shape has much more of a bulge

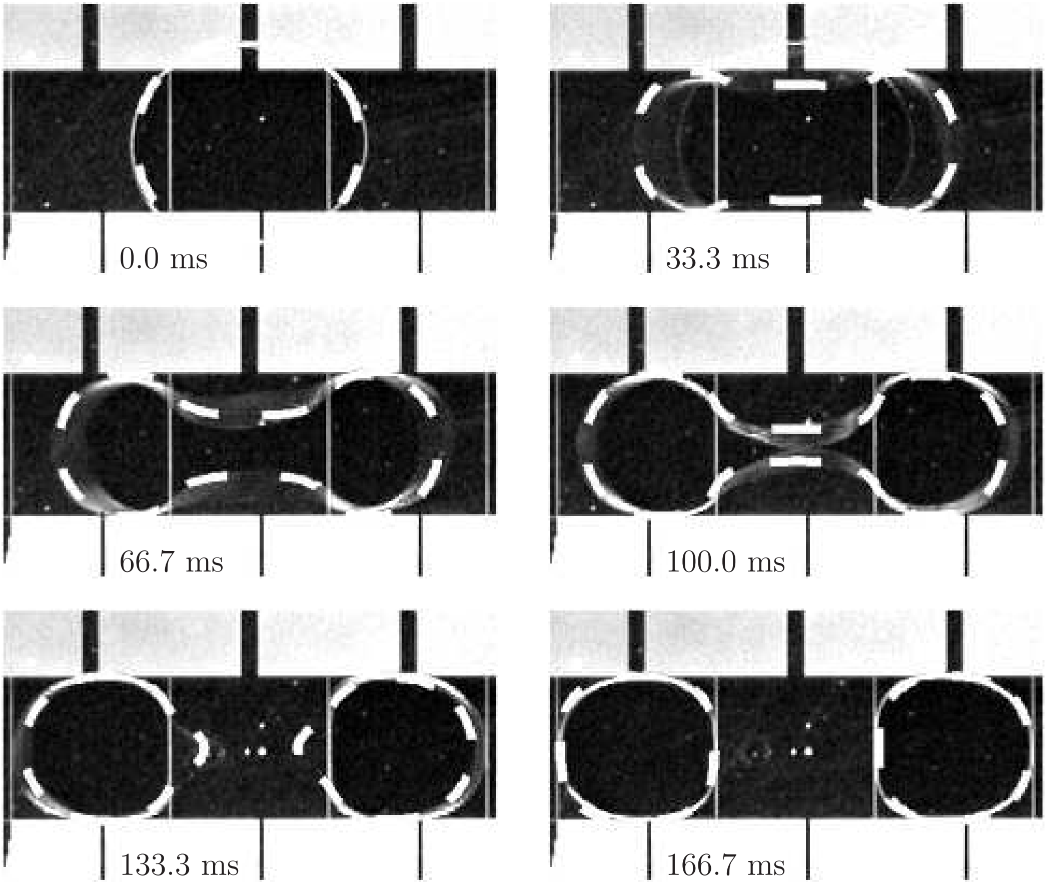


Figure 5.1: Droplet splitting experimental results with level set simulation overlay. Six frames showing the video snapshots of the experiment (courtesy of CJ Kim and Sung Kwon Cho at UCLA). The three electrodes shown in each frame have activation voltages (from left to right) of 25, 0, and 25 volts. Each electrode is approximately square with a side length of 1.4 millimeters. The dashed-line droplet outlines are from the simulation depicted in Figures 5.10 and 5.11 and show a direct comparison between experiment and the level set simulation including contact angle saturation and hysteresis.

	Y-L	*Sat*	*All*	*Units*
L	4.406	4.406	4.406	mm
U_0	1500	500	50	mm/sec
t_0	2.94	8.81	88.12	msec
P_0	16.3	16.3	16.3	N/m^2
Re	117.6	39.2	3.92	non-dim.
Ca	0.01854	0.006181	0.000618	non-dim.
α	137.3	15.25	1.695	non-dim.
β	881.4	293.85	326.5	non-dim.
K_{hys}	1.0	1.0	0.09	non-dim.

Table 5.1: Simulation Parameters for the Level Set Method: Numbers for three simulations are listed here: (Y-L) uses the Young-Lippmann theory, (Sat) adds in saturation, and (All) includes saturation and hysteresis. Each simulation uses a different value of U_0 so the maximum non-dimensional velocity is close to unity. This also causes Re, Ca, α, β, and t_0 to differ. Grid resolution is 108x108.

in the center than shown in the experiment. This becomes more pronounced in later frames, with two thin necks developing between the three bulging parts of the droplet. The droplet then splits into three pieces instead of two as in the experiment. Finally, the two larger droplets come to rest on the active electrodes, with the smaller satellite droplet resting in the center. The total time to complete the split and reach equilibrium is 8.8 milliseconds, 18.9 times faster than the experiment.

The variational method, shown in Figures 5.4 and 5.5, gives a similar result except the total split time is 7.6 milliseconds.

Because no saturation or hysteresis is being modeled, the simulated EWOD

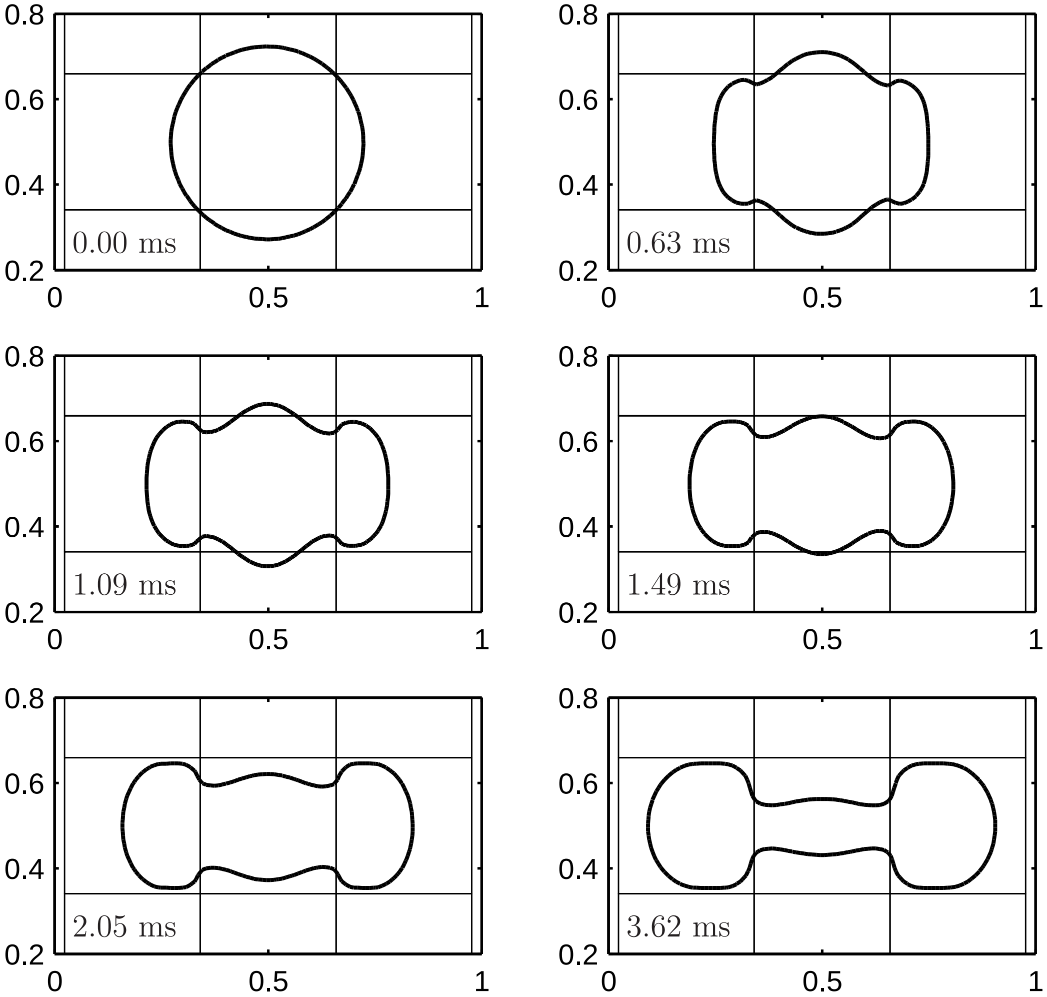


Figure 5.2: Young-Lippmann Model with the level set method (part A): simulation frames showing splitting behavior under the ideal Young-Lippmann theory. Simulation continued in Figure 5.3.

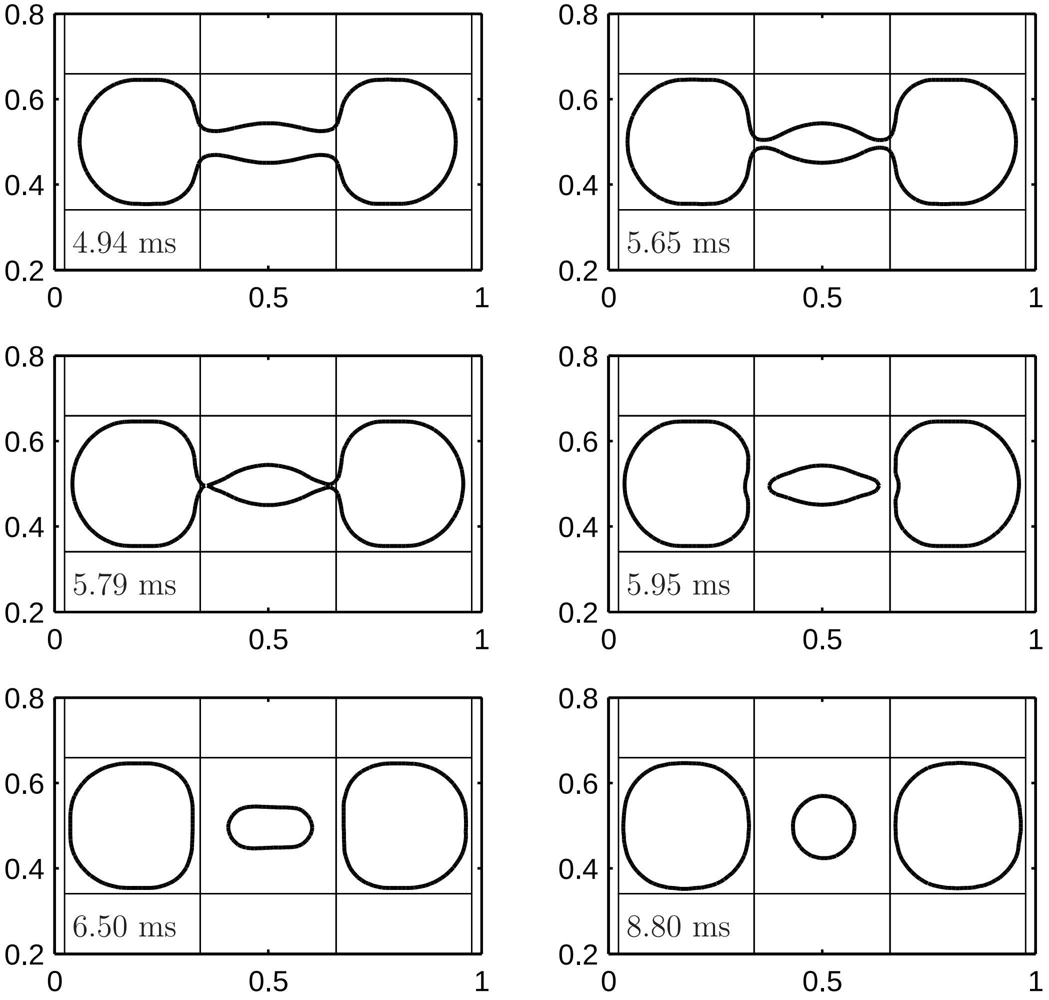


Figure 5.3: Young-Lippmann Model with the level set method (part B): simulation frames showing splitting behavior under the ideal Young-Lippmann theory. Split time is 18.9 times faster than the experiment.

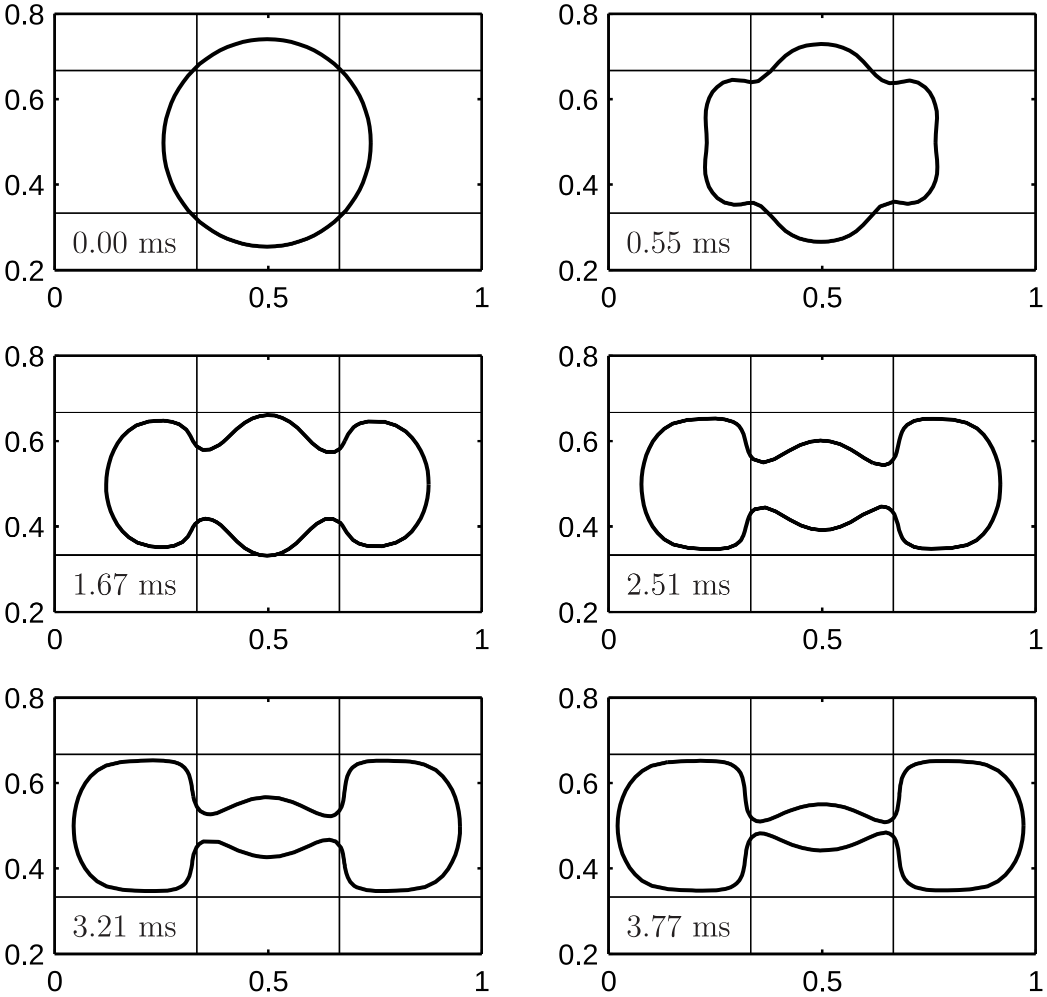


Figure 5.4: Young-Lippmann Model with the variational method (part A): simulation frames showing splitting behavior under the ideal Young-Lippmann theory. Simulation continued in Figure 5.5.

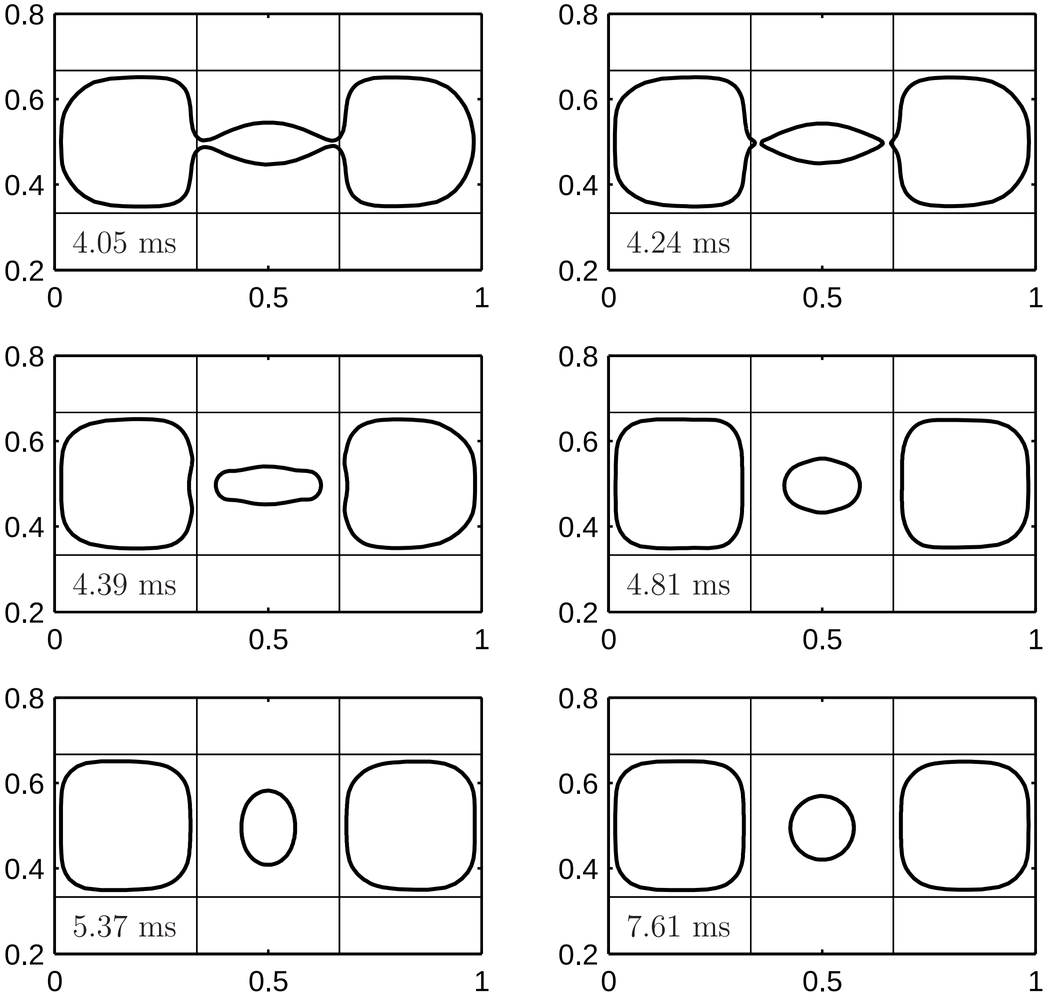


Figure 5.5: Young-Lippmann Model with the variational method (part B): simulation frames showing splitting behavior under the ideal Young-Lippmann theory. Split time is slightly shorter than with the level set method.

	Y-L	*Sat*	*All*	*Units*
L	4.2	4.2	4.2	mm
U_0	750	250	50	mm/sec
t_0	5.6	16.8	84	msec
P_0	17.14	17.14	17.14	N/m^2
c_{pin}	0.0	0.0	0.003	N/m
Re	58.8	19.6	3.9	non-dim.
Ca	0.009272	0.00309071	0.0006181	non-dim.
α	32.7	3.64	0.9662	non-dim.
β	400.6	133.5	177.4	non-dim.
K_{hys}	1.0	1.0	0.1505	non-dim.

Table 5.2: Simulation Parameters for the Variational Method: Numbers for three simulations are listed here: (Y-L) uses the Young-Lippmann theory, (Sat) adds in saturation, and (All) includes saturation and hysteresis. Each simulation uses a different value of U_0 so the maximum non-dimensional velocity is close to unity. This also causes Re, Ca, α, β, and t_0 to differ. The length scale is slightly different than for the level set method because there is no 'buffer region'.

force is much larger than in reality. This causes the droplet to be pulled apart so fast that the middle region is never able to become a thin neck. As a result, three satellite droplets are created instead of two. In fact, the z curvature of the liquid-gas interface (i.e. the EWOD force) is so large that the x-y curvature component is practically negligible. This is why the droplet does not resist being pinched in two places; the EWOD force here dominates the large curvature forces induced by the pinched regions.

5.1.2 Saturation

For the simulation (level set) shown in Figures 5.6 and 5.7, we have added the effect of contact angle saturation. The splitting motion of the droplet now looks much closer to the experiment. As the droplet is pulled apart, a single thin neck joins the two bulging ends. The neck then breaks, allowing the two droplets to come to rest on the left and right electrodes. However, the time scale is still not correct. The time to reach equilibrium here is 14.33 milliseconds, which is 11.6 times faster than the experiment.

The variational method (Figures 5.8 and 5.9) gives a similar result except a small satellite drop is released in the center and the split time is 12.17 milliseconds. The lack of a small satellite drop in the level set simulation is possibly due to the coarse grid used (108x108), whereas the triangular mesh in the variational simulation becomes more refined around the pinching area.

Including saturation does slow the droplet, but it is still not enough. In reality, the line pinning induced contact angle hysteresis is extremely significant in most wetting phenomena. Hysteresis slows motion by reducing the pressure gradient (see Section 2.2.4). Therefore, it is not surprising that this effect must be included to accurately simulate droplet speed. The following section presents our final simulation of splitting with hysteresis included (and also contact line pinning for the variational method).

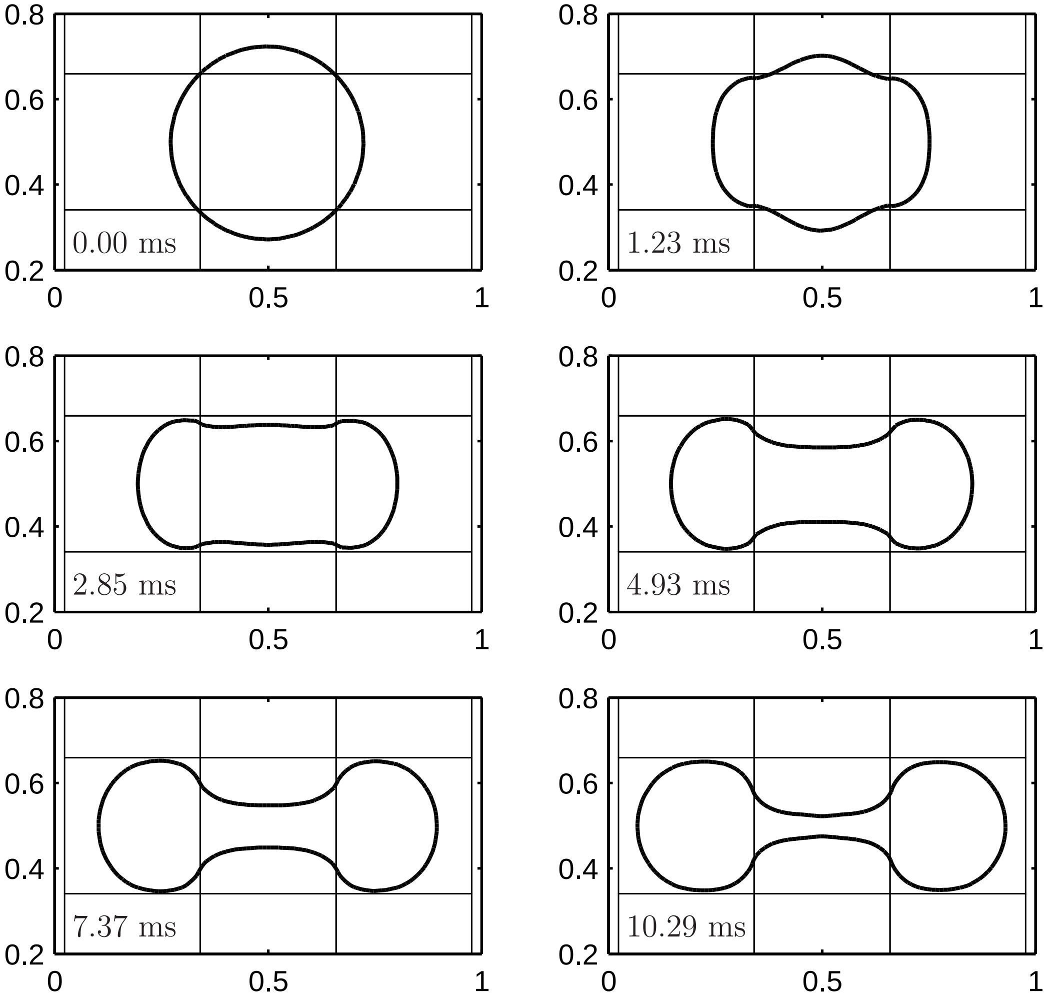


Figure 5.6: Saturation Model with the level set method (part A): simulation frames showing splitting behavior when contact angle saturation is included. Simulation continued in Figure 5.7.

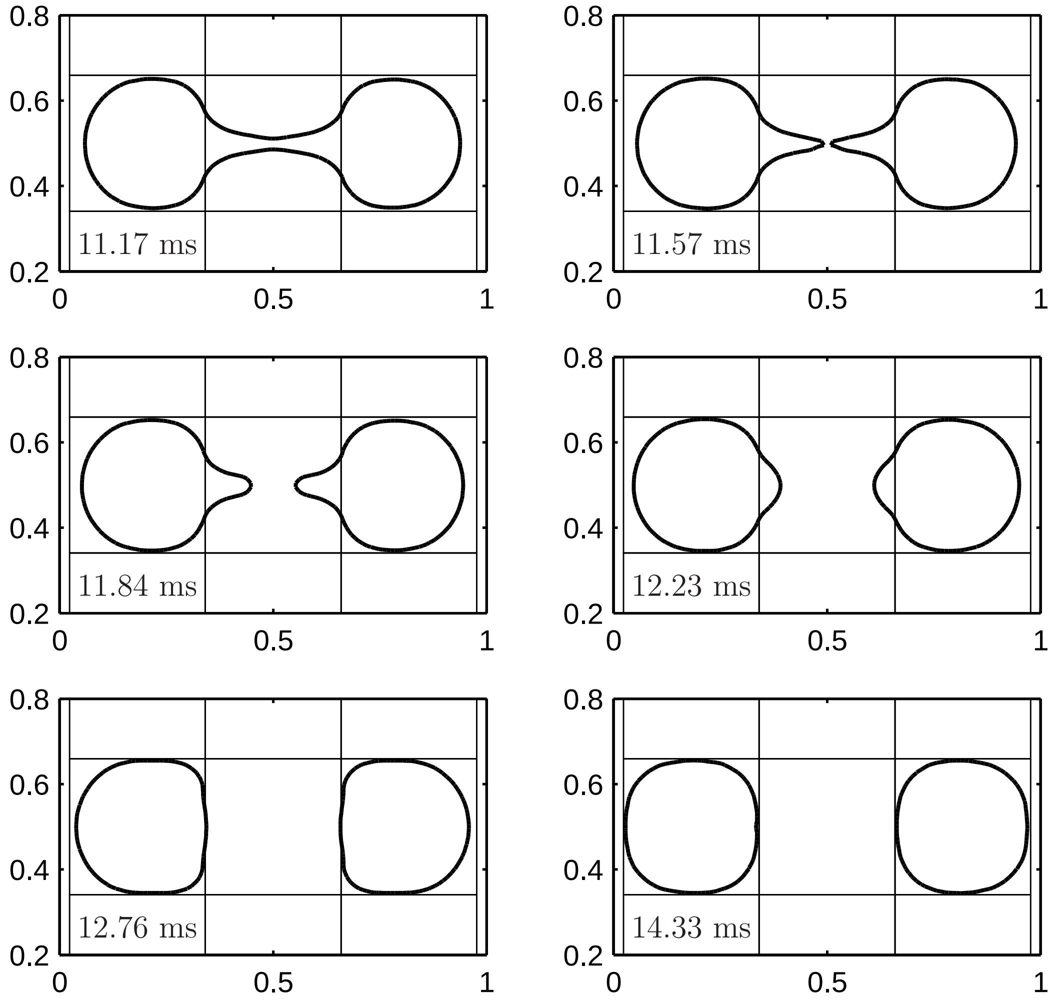


Figure 5.7: Saturation Model with the level set method (part B): simulation frames showing splitting behavior under the contact angle saturation model. Split time is 11.6 times faster than the experiment.

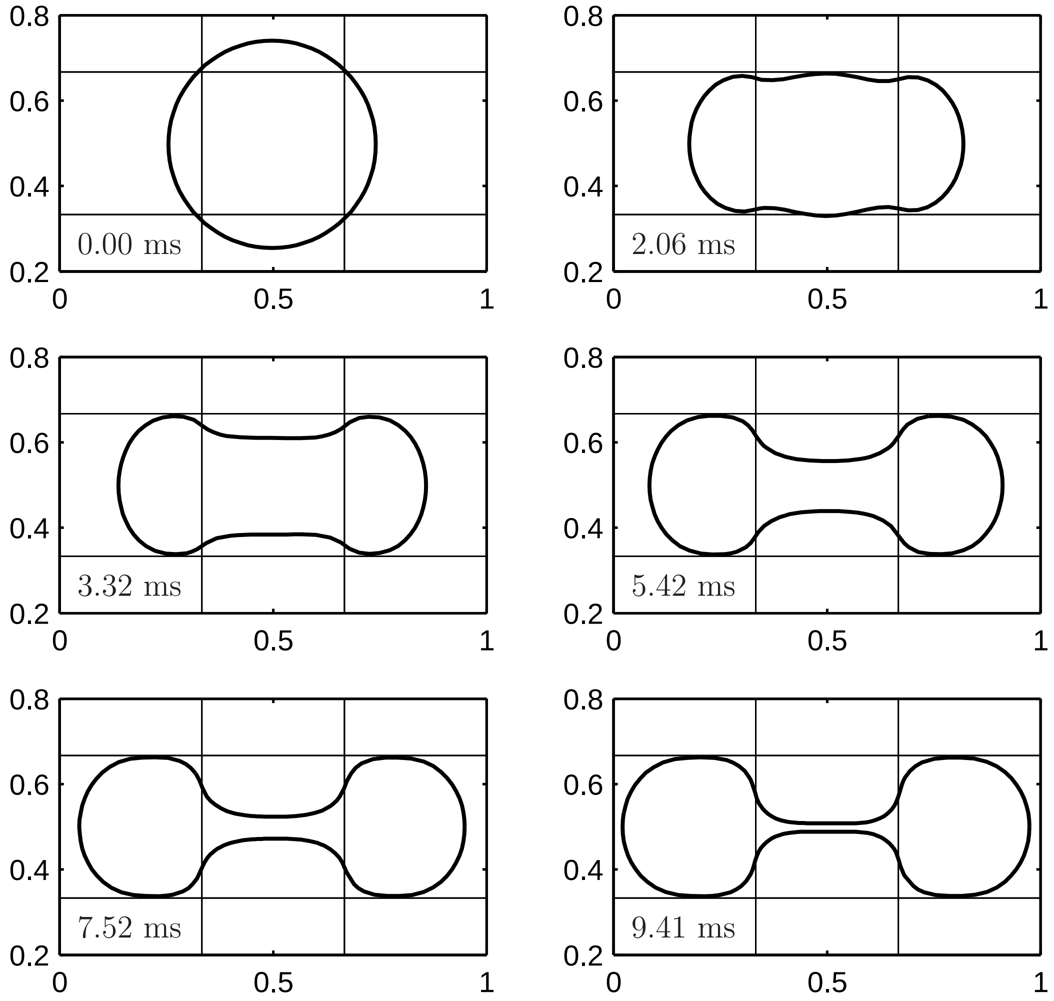


Figure 5.8: Saturation Model with the variational method (part A): simulation frames showing splitting behavior when contact angle saturation is included. Simulation continued in Figure 5.9.

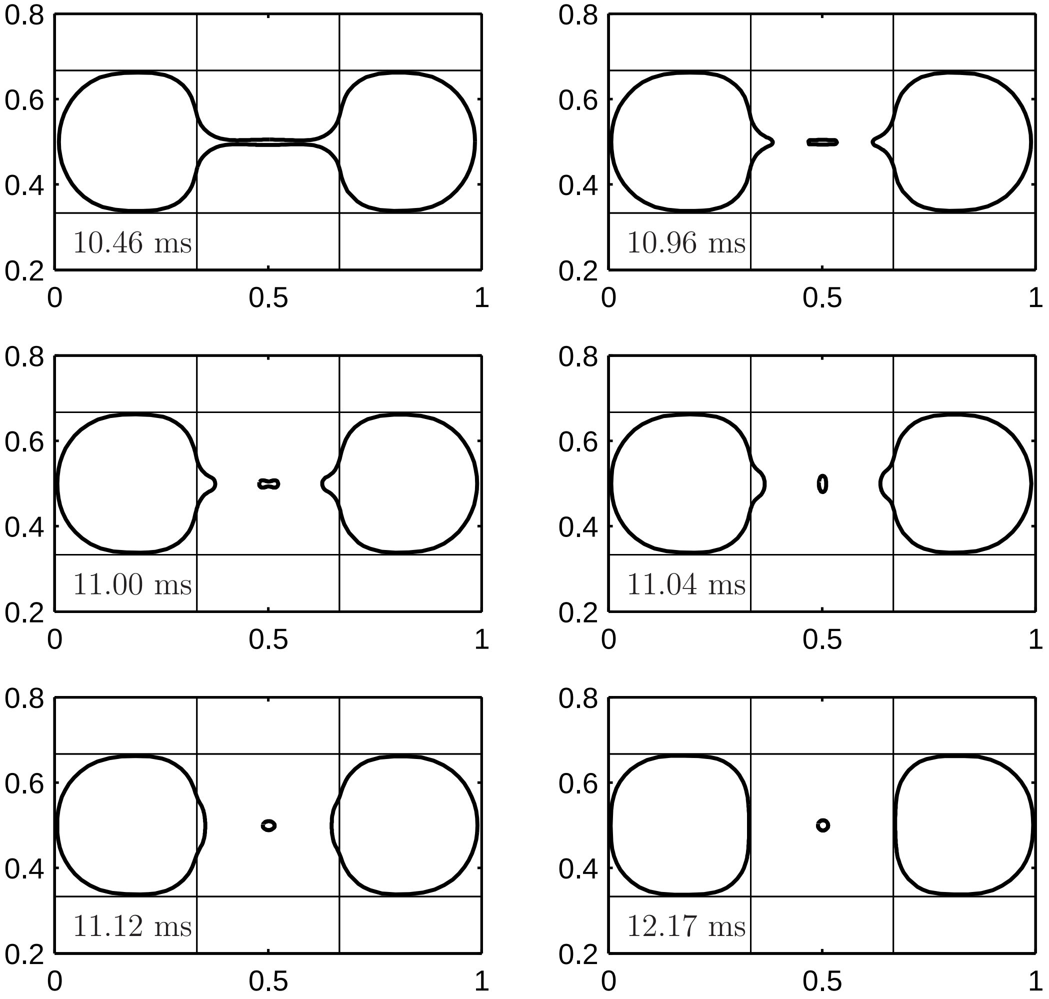


Figure 5.9: Saturation Model with the variational method (part B): simulation frames showing splitting behavior under the contact angle saturation model. A small satellite drop is released in the center (not present in the level set simulation). The split time is also slightly shorter than with the level set method.

5.1.3 Contact Angle Hysteresis and Line Pinning

In Figures 5.10 and 5.11, we show several frames of our simulated (level set) splitting droplet, which are also shown as dashed-line overlays in Figure 5.1. The hysteresis constant, K_{hys}, is 0.09. This simulation is similar to the one in Figures 5.6 and 5.7, except that the time scale is now correct. The simulated droplet now splits in the same amount of time as the experiment, as shown in Figure 5.1.

The value of the hysteresis constant, $K_{hys} = 0.09$, was chosen to make the simulation time scale match the experiment. By using (2.11) and experimental data from Figure 2.6, we estimate the contact angle deflection due to hysteresis to be $\Delta_{hys} = 6.4°$. In [106], the authors give a value of about 20° for sessile drops of water sliding on top of a Teflon surface. This discrepancy is reasonable because the droplet size and geometry in the EWOD device is different than in [106].

Our hysteresis constant is also analogous to the contact line friction coefficient in [143], where they treat contact line friction as an extra forcing term proportional to the velocity of the contact line. In their case, the forcing term has units of force per unit contact line length. By scaling their friction force by the ratio of contact line length to volume for a droplet in an EWOD device (to put it into units of force per unit volume), we can include this as a body force term in the Navier-Stokes equations. After going through the same derivation as in Section 2.1.2, we obtain an equation similar to (2.6), except the coefficient of the velocity term has an extra positive term added to $12(L/H)^2$. Hence, the coefficient is larger than before. If we ignore the velocity time derivative term in (2.6), the extra friction force is equivalent

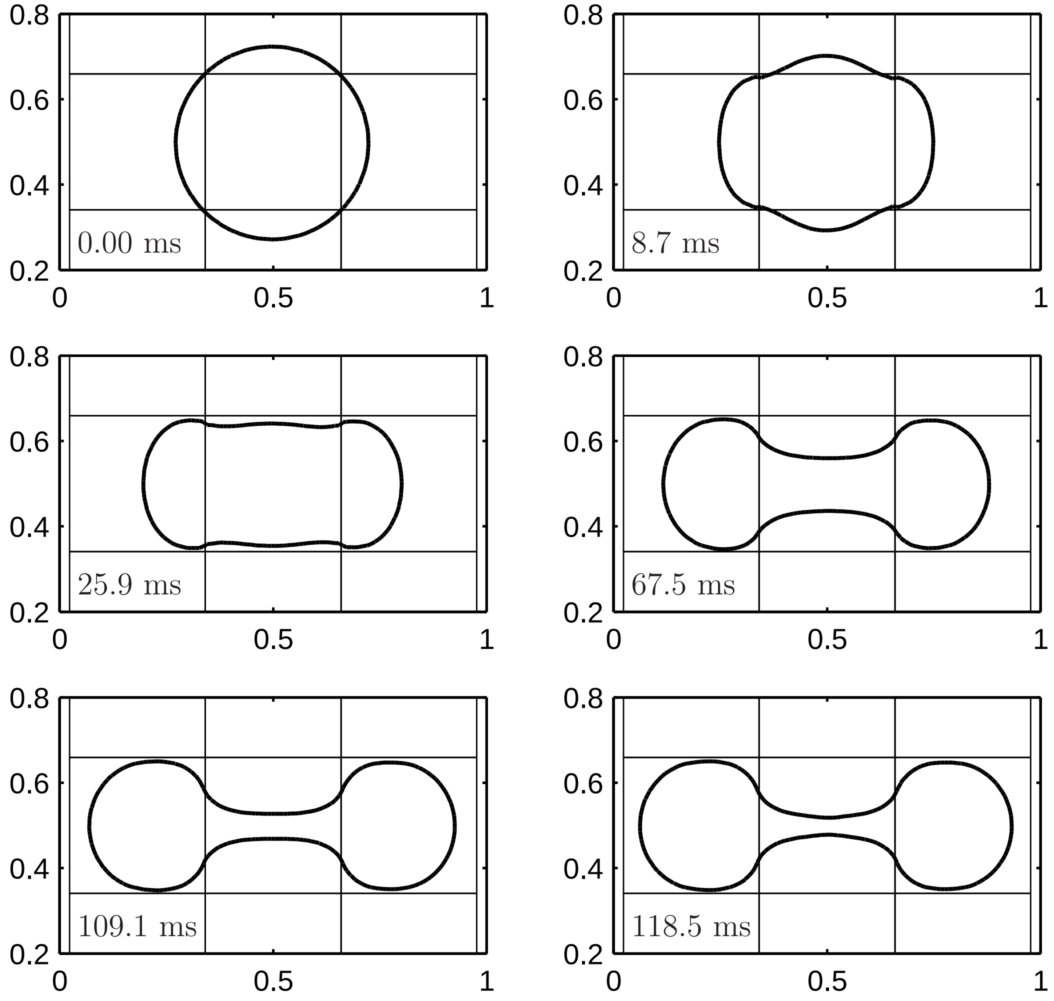


Figure 5.10: Saturation and Hysteresis Model with the level set method (part A): simulation frames showing splitting behavior when both contact angle saturation and hysteresis are added ($K_{hys} = 0.09$). Simulation continued in Figure 5.11.

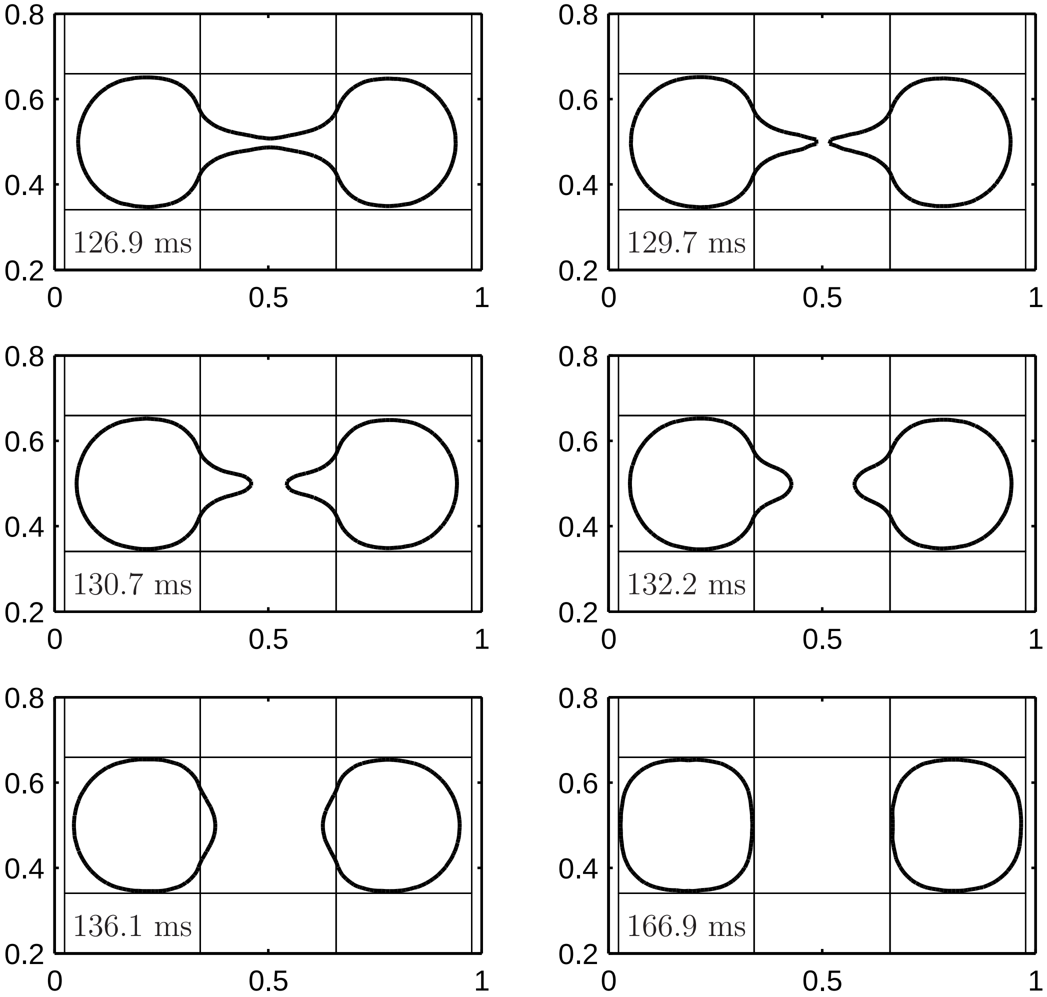


Figure 5.11: Saturation and Hysteresis Model with the level set method (part B): continuation from previous figure. The droplet splits in the same amount of time as the experiment. Overlays of this simulation are shown in Figure 5.1 on top of video frames from the actual experiment.

to multiplying the pressure gradient by a constant smaller than 1 (i.e. K_{hys}). In fact, our choice of hysteresis constant corresponds to a contact line friction coefficient of 0.5664 Newton-seconds per square meter, which is comparable to the values listed in [143] for a column of fluid comprised of deionized water between two parylene coated electrodes.

However, keep in mind that these estimates are rough approximations. The hysteresis constant is not an exact model and it does not capture line pinning. Our initial goal was not to model line pinning or contact angle hysteresis in great detail, but to show that a simple scaling constant is all that is needed to produce simulations that approximately capture the shape and speed of splitting droplet motion. In other cases of droplet motion, however, contact line pinning is more noticeable (see experiments in the following sections) and should be included in the model.

In Figures 5.12 and 5.13, we have our variational simulation that includes contact angle saturation, hysteresis, and contact line pinning. The contact line pinning coefficient is taken from experiments done in our lab, and is given by $c_{pin} = 3 \times 10^{-3}$ N / m. The hysteresis constant required to match the simulated split time to the experiment is given by $K_{hys} = 0.1505$, which is larger than for the level set method. This is because our contact line pinning model partially retards the EWOD force, so less hysteresis is required to slow the droplet down. See Figure 5.14 for an overlay of the variational simulation with the experiment.

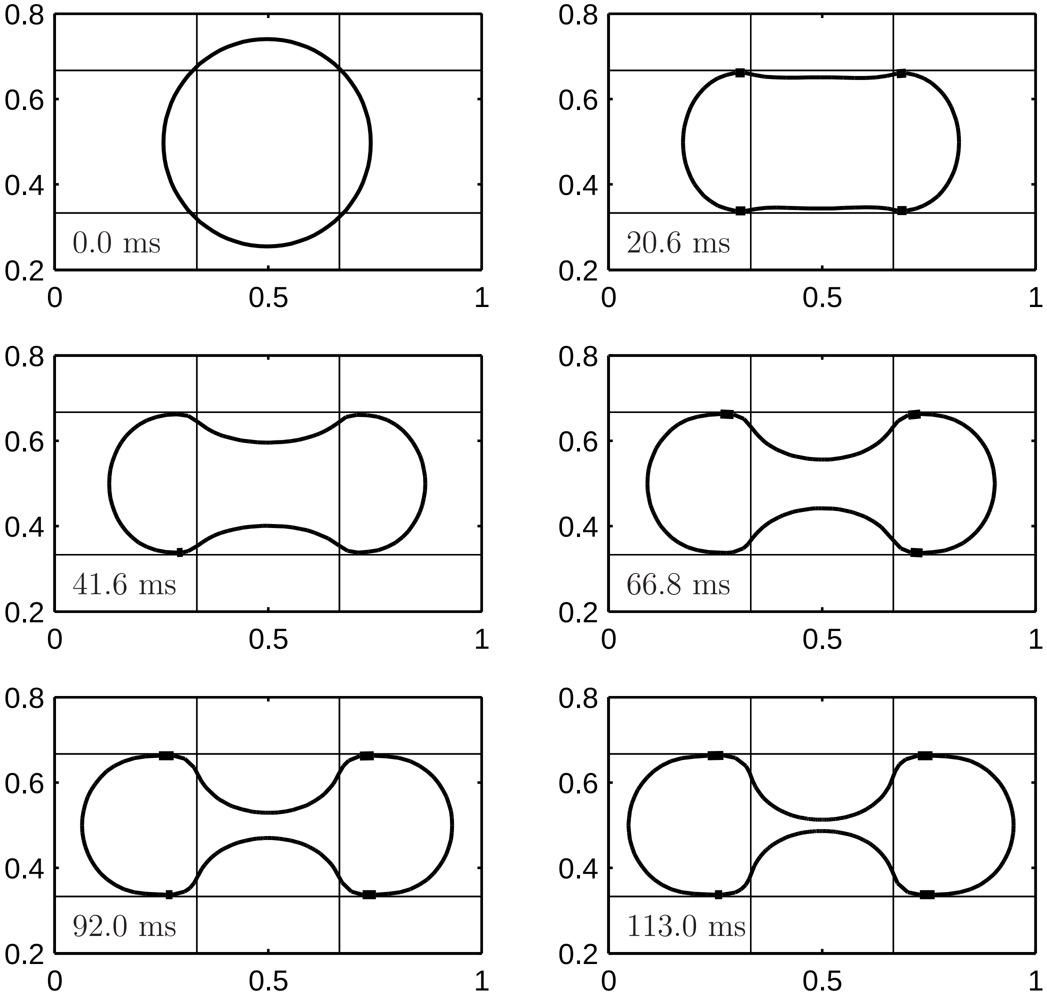


Figure 5.12: Saturation, Hysteresis, and Contact Line Pinning Model with the variational method (part A): simulation frames showing splitting behavior when the full model is used ($K_{hys} = 0.1505$, $c_{pin} = 3 \times 10^{-3}$ N / m). Bolded parts of the droplet interface represent regions that are 'pinned' because of contact line pinning. Note that pinning only restricts the normal velocity to zero; there can still be a tangential motion. Simulation continued in Figure 5.13.

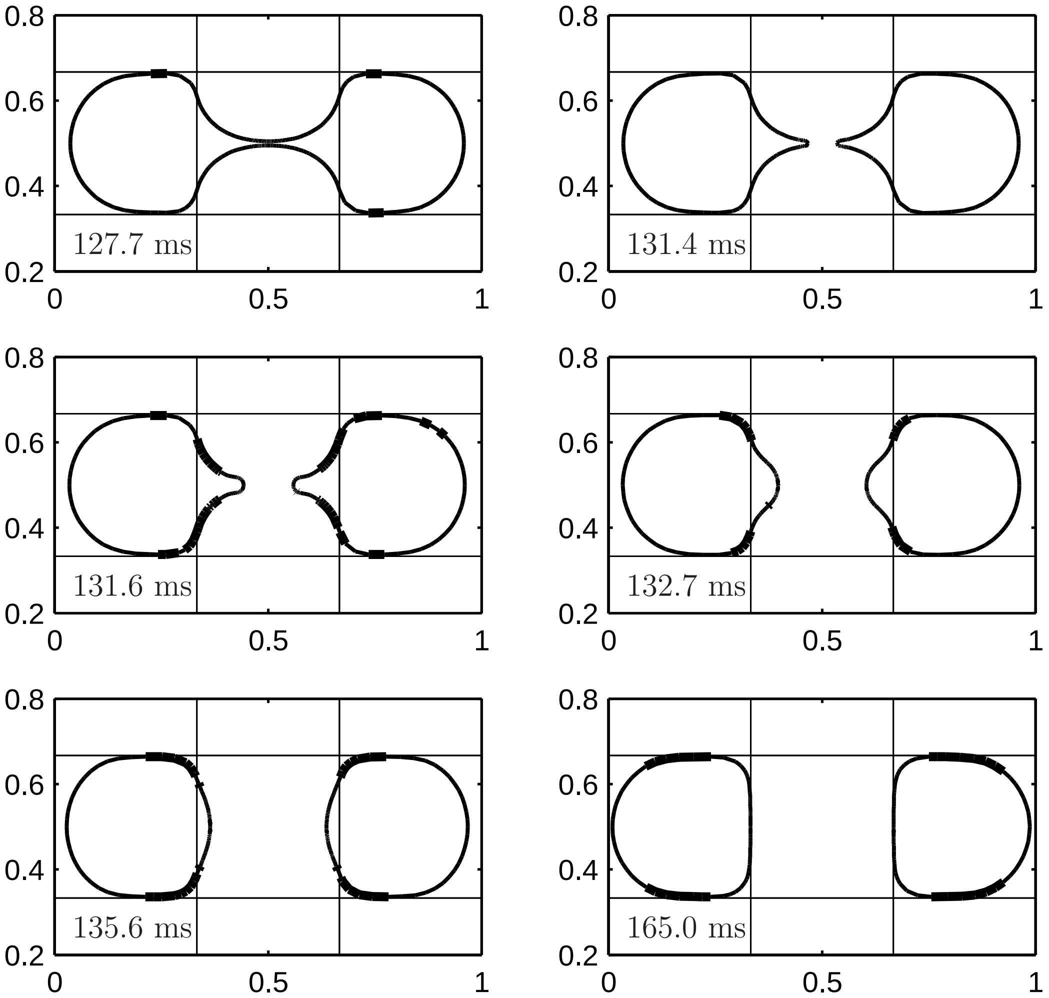


Figure 5.13: Saturation, Hysteresis, and Contact Line Pinning Model with the variational method (part B): continuation from previous figure (same format). The droplet splits in the same amount of time as the experiment. Overlays of this simulation are shown in Figure 5.14 on top of video frames from the actual experiment.

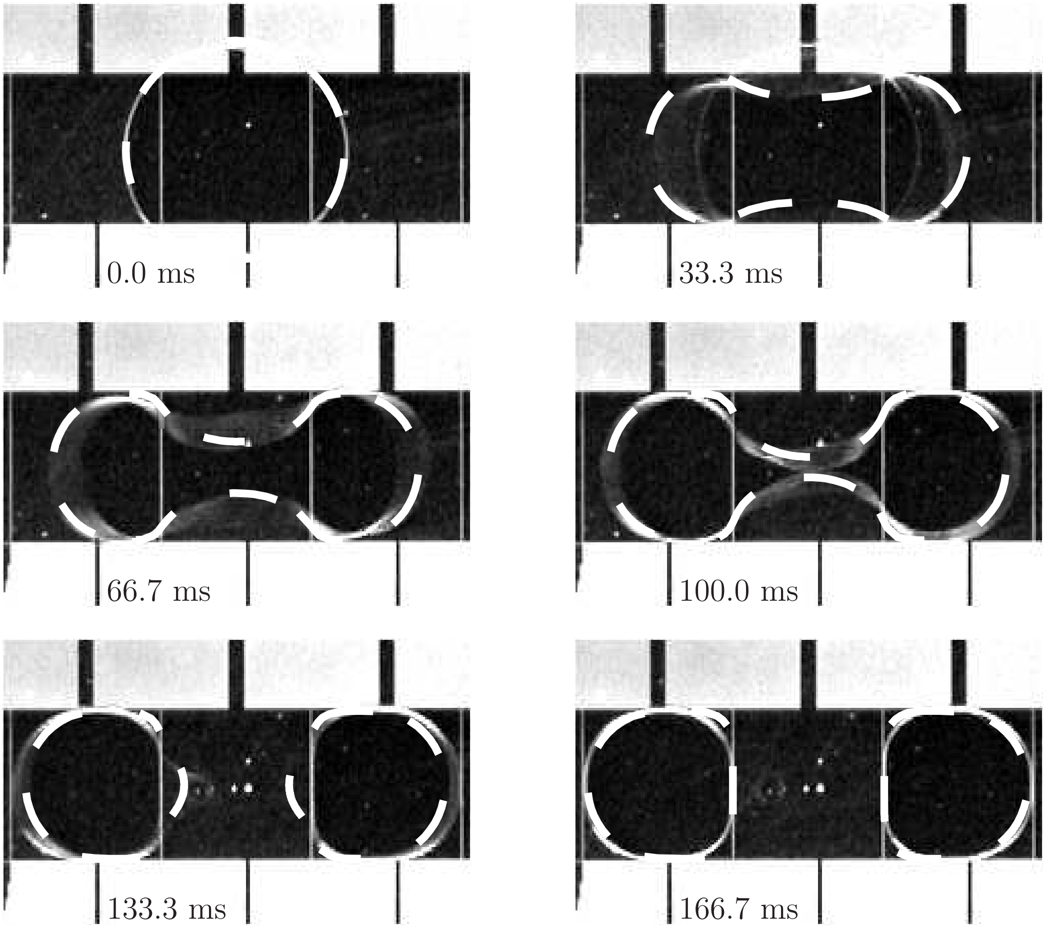


Figure 5.14: Droplet splitting experimental results with variational simulation overlay. Same format as Figure 5.1. The dashed-line droplet outlines are from the simulation depicted in Figures 5.12 and 5.13 and show a direct comparison between experiment and the variational simulation including contact angle saturation, hysteresis, and contact line pinning.

5.2 A Moving Water Droplet

This section compares our simulation to another experimental case to give more supporting evidence for our model. The EWOD device shown here has eight electrodes arranged in a square-like pattern. A predetermined voltage sequence was used to actuate the droplet so it moves to the right first, then up, and finally to the left. All constants in our level set simulation are the same as those used in Section 5.1.3 (i.e. $K_{hys} = 0.09$), and the same computational grid resolution (108x108) is used (see Table 5.1).

From Figure 5.15, it is evident that the simulation follows the experiment fairly well. The match is not exact, however, and this is mainly because line pinning is not taken into account, and hysteresis is not being modeled very accurately. But the overall motion and time scale are correct.

Figure 5.16 shows the same moving droplet experiment with our variational simulation that also includes contact line pinning. All constants are the same as in Section 5.1.3 (i.e. $K_{hys} = 0.1505$, $c_{pin} = 3 \times 10^{-3}$ N / m); see Table 5.2.

5.3 A Moving Glycerin Droplet

The EWOD device in this experiment has only two electrodes arranged in a horizontal fashion with a droplet of Glycerin (Glycerol) being actuated. A voltage of 50 volts is first applied to the left electrode with 0 volts on the right. This causes the droplet to flow to the left electrode. The voltage actuation is kept constant for two seconds when it switches to 0 volts on the left, 50 volts on the right, which

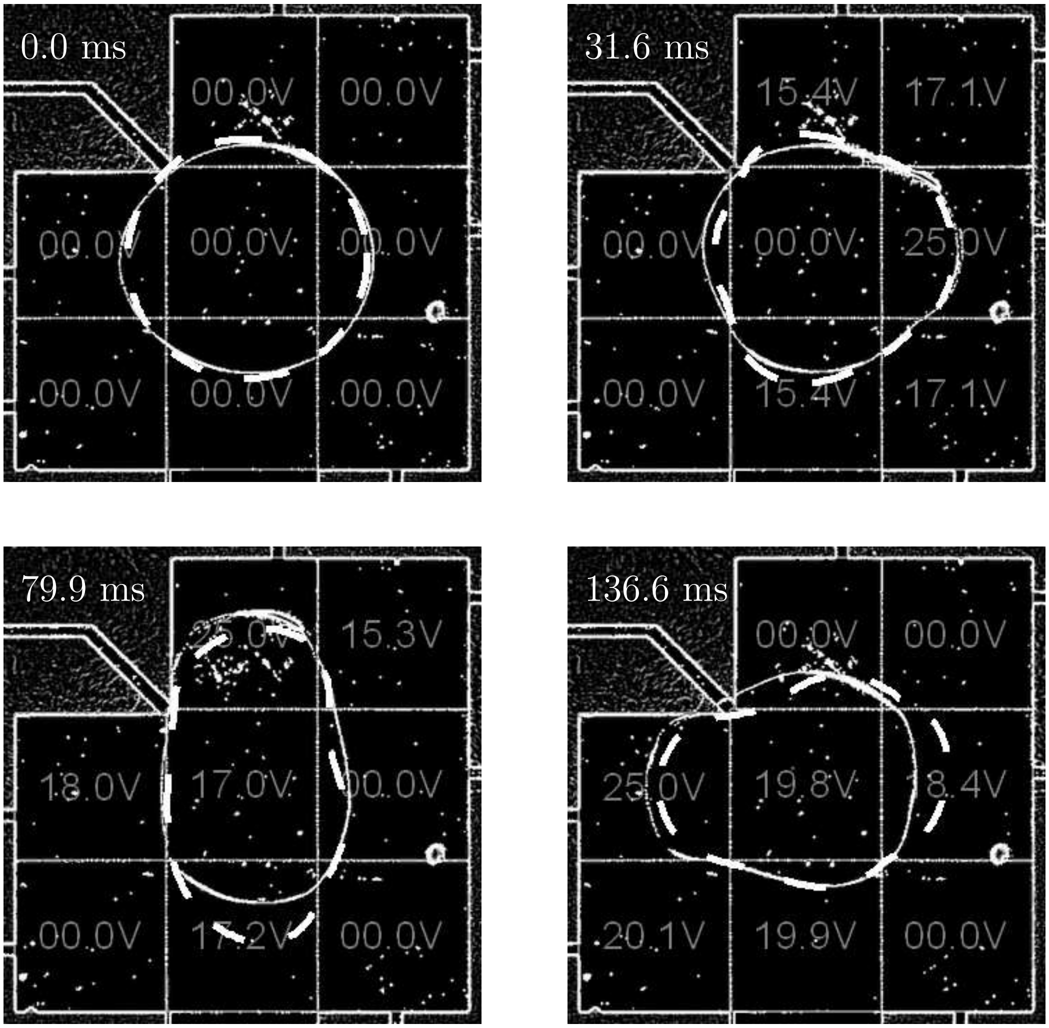


Figure 5.15: Moving water droplet motion experimental results with level set simulation overlay. Four frames show video snapshots of the experiment (courtesy of CJ Kim and Jian Gong at UCLA). A time-varying sequence of voltages is applied to the eight electrode pattern so as to make the droplet move right, up, and then left. Each electrode is square with a side length of 1.4 millimeters. All device parameters here are the same as for the splitting experiment shown in Figure 5.1 except the electrode pattern is different. The dashed-line droplet outlines (from simulation) show a direct comparison between the experiment and a level set simulation including contact angle saturation and hysteresis ($K_{hys} = 0.09$).

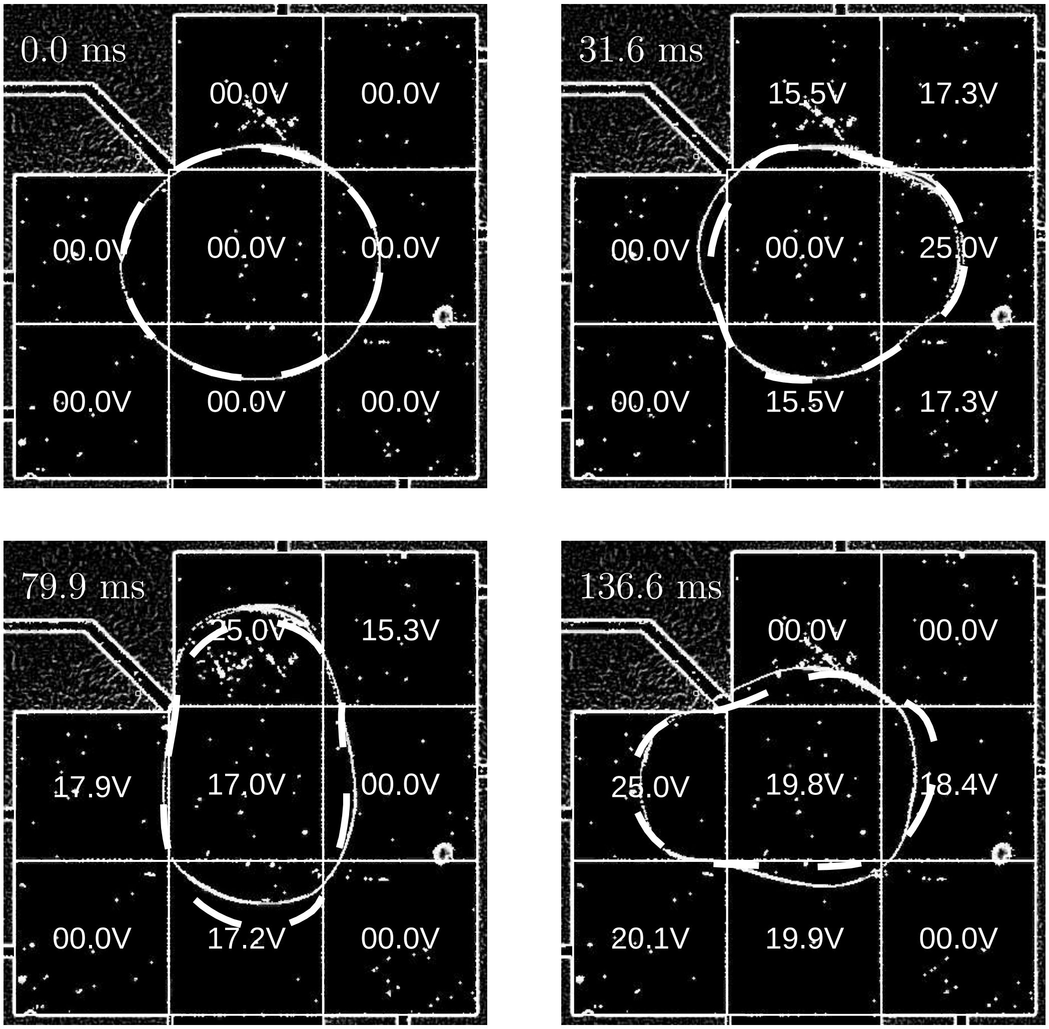


Figure 5.16: Moving water droplet motion experimental results with variational simulation overlay. Same format as Figure 5.15. The dashed-line droplet outlines (from simulation) show a direct comparison between the experiment and a variational simulation including contact angle saturation, hysteresis, and line pinning ($K_{hys} = 0.1505$, $c_{pin} = 3 \times 10^{-3}$ N / m). The simulated droplet appears to follow the real droplet a little more closely as compared to the level set simulation in Figure 5.15.

causes the droplet to switch direction of motion. Again, the voltage actuation is kept constant until after two seconds it switches back, causing the droplet to also switch its direction of travel. This process repeats.

Figure 5.17 shows a comparison between the variational simulation and the experiment; Figure 5.18 shows only the simulation. Simulation and device parameters are given in Table 5.3.

5.4 A Joining Water Droplet

The EWOD device in this experiment has three electrodes arranged in a horizontal fashion with two droplets of water being actuated to join together. A voltage of 65 volts is applied to the center electrode (with wire coming out the top), with 0 volts applied to the left and right electrodes, which causes the two droplets to flow together. The voltage actuation is constant throughout the experiment. Eventually, the two droplets connect and merge together.

Figure 5.19 shows a comparison between the variational simulation and the experiment; Figure 5.20 shows only the simulation. Simulation and device parameters are given in Table 5.3.

5.5 A Splitting Glycerin Droplet

This experiment uses the same EWOD device as in Section 5.4, except that a Glycerin droplet is being actuated to split apart. A voltage of 65 volts is applied to the left and right electrodes, with 0 volts applied on the center electrode which

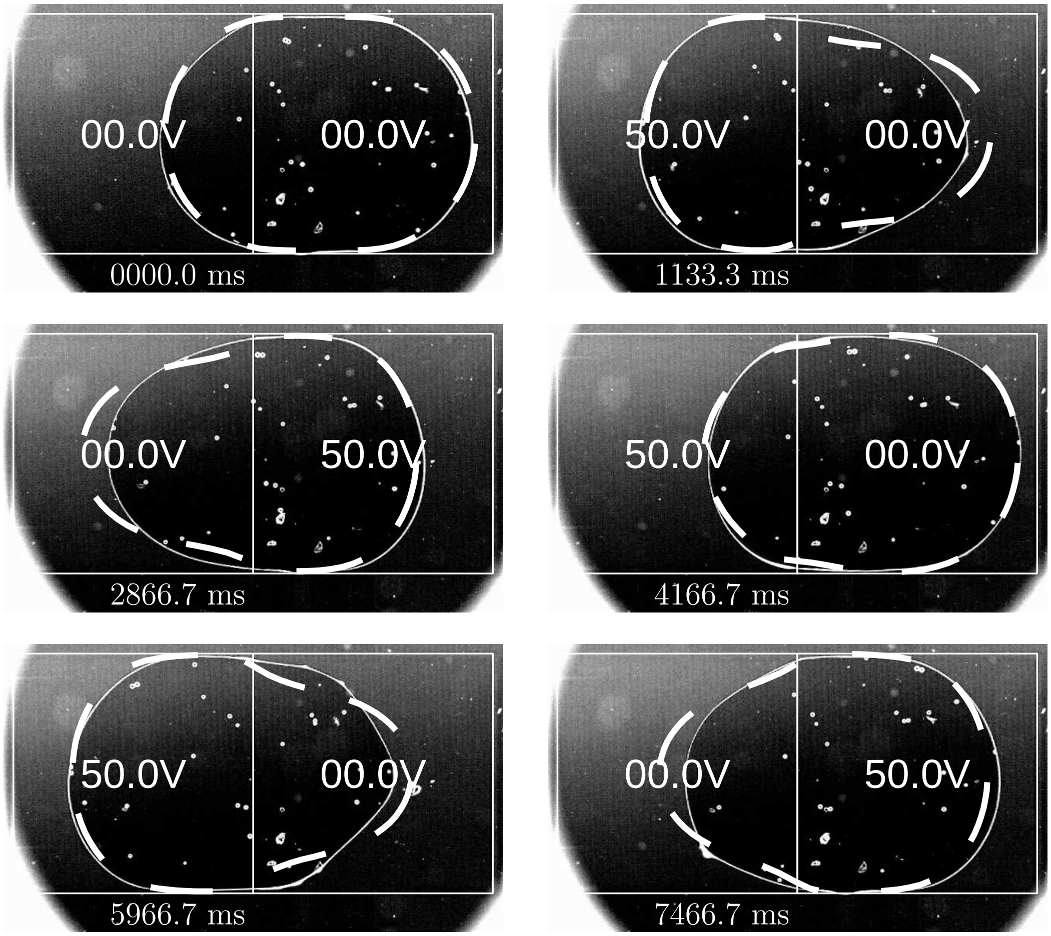


Figure 5.17: Moving Glycerin droplet experimental results with variational simulation overlay. Frames show video snapshots of the experiment (courtesy of CJ Kim and Jian Gong at UCLA). The applied voltage (50 volts) switches between the left and right electrodes every two seconds. Each electrode is square with a side length of 1.5 millimeters. Simulation and device parameters are given in Table 5.3. Note the large time-scale because glycerin is highly viscous. The simulation follows the experiment fairly well, except the 'tail' narrows more in the simulation than in the experiment.

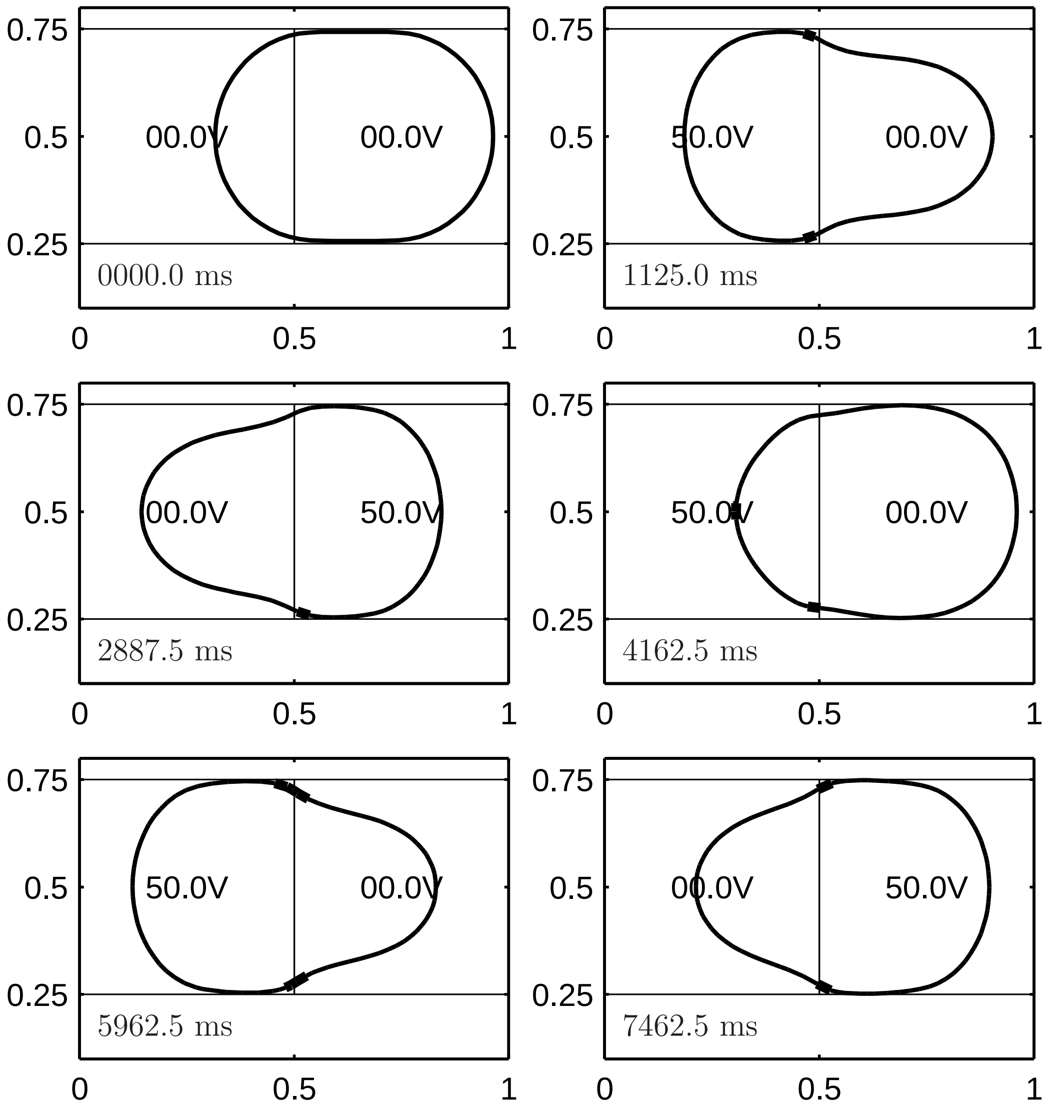


Figure 5.18: Moving Glycerin droplet variational simulation. The applied voltage (50 volts) switches between the left and right electrodes every two seconds. Each electrode is square with a side length of 1.5 millimeters. Simulation and device parameters are given in Table 5.3. Bolded parts of the droplet interface represent regions that are 'pinned' because of contact line pinning.

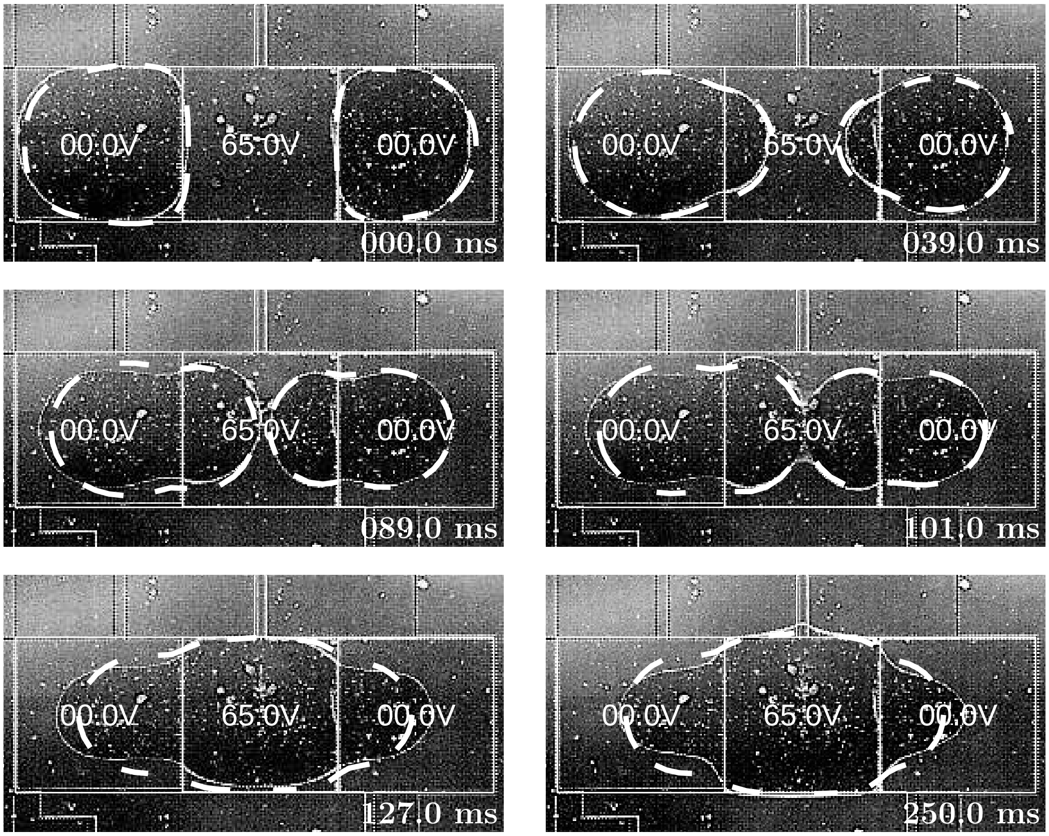


Figure 5.19: Joining water droplets experimental results with variational simulation overlay. Frames show video snapshots of the experiment (courtesy of CJ Kim and Jian Gong at UCLA). The applied voltage (65 volts on the center electrode only) causes the two side droplets to flow together and eventually merge. Each electrode is approximately square with a side length of 1.5 millimeters. Simulation and device parameters are given in Table 5.3. The simulation matches the experiment fairly well in the first four frames. However, the pinning behavior in the experiment is significantly different than the simulation in the last two frames.

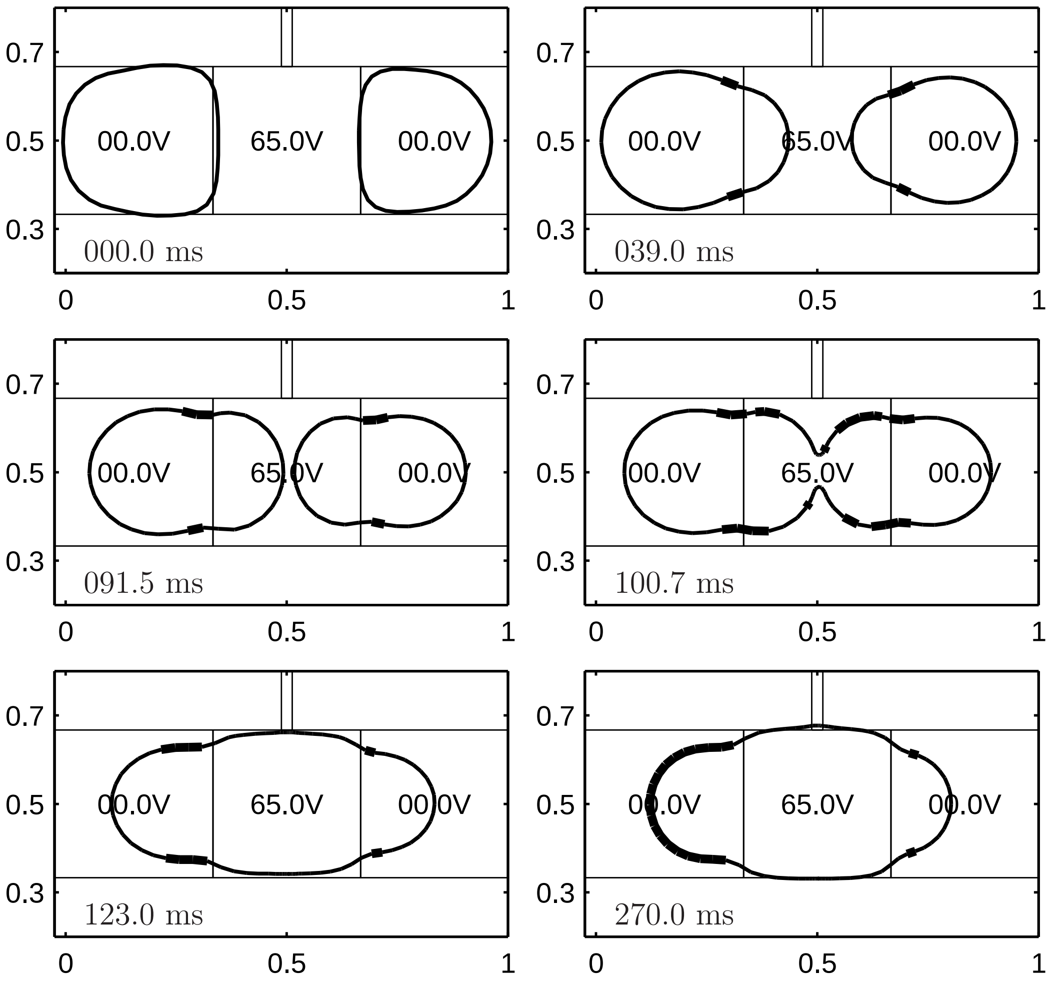


Figure 5.20: Joining water droplets variational simulation. The applied voltage (65 volts on the center electrode only) causes the two side droplets to flow together and eventually merge. Each electrode is approximately square with a side length of 1.5 millimeters. Simulation and device parameters are given in Table 5.3. Bolded parts of the droplet interface represent regions that are 'pinned' because of contact line pinning.

causes the droplet to pull apart. The voltage actuation is constant throughout the experiment. Eventually, a thin neck develops between two smaller droplets and the neck collapses.

Figure 5.21 shows a comparison between the variational simulation and the experiment; Figure 5.22 shows only the simulation. Simulation and device parameters are given in Table 5.3.

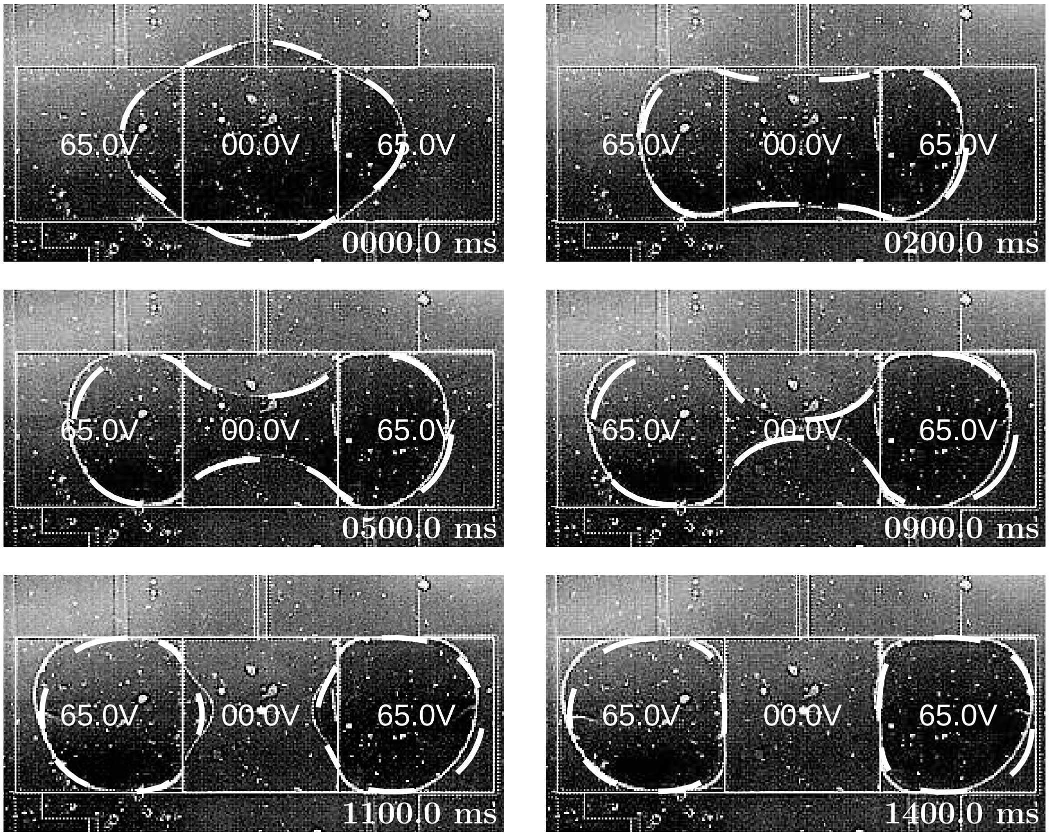


Figure 5.21: Splitting Glycerin droplet experimental results with variational simulation overlay. Frames show video snapshots of the experiment (courtesy of CJ Kim and Jian Gong at UCLA). The applied voltage (65 volts on the left and right electrodes) causes the droplet to be pulled apart and eventually split. Each electrode is approximately square with a side length of 1.5 millimeters. Simulation and device parameters are given in Table 5.3. The match between the simulation and experiment is very good. The only difference is that, in the experiment, slightly more fluid flows into the left satellite droplet than in the right droplet.

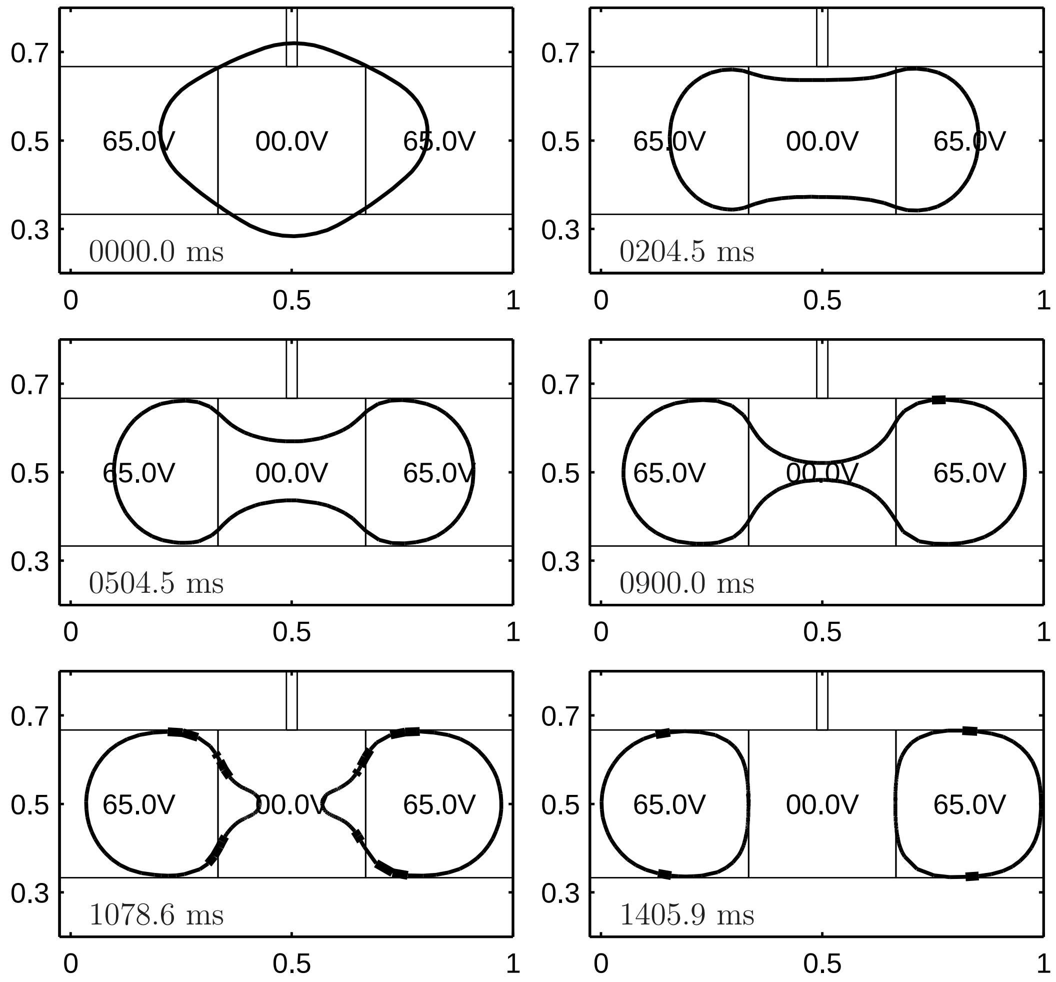


Figure 5.22: Splitting Glycerin droplet variational simulation. The applied voltage (65 volts on the left and right electrodes) causes the droplet to be pulled apart and eventually split. Each electrode is approximately square with a side length of 1.5 millimeters. Simulation and device parameters are given in Table 5.3. Bolded parts of the droplet interface represent regions that are 'pinned' because of contact line pinning.

	Moving Glycerin	Joining Water	Splitting Glycerin	*Units*
σ_{lg}	0.0645	0.072294	0.0645	J / m^2
μ	265.98	0.9348	265.98	g / m-sec
ρ	1235.15	997.44	1235.15	Kg / m^3
H	0.1	0.1	0.1	mm
L_{Elec}	1.5	1.5	1.5	mm
L	3.0	4.5	4.5	mm
U_0	0.4	15	1.1	mm / sec
t_0	7500	300	2727.3	msec
P_0	21.5	16.065	21.5	N / m^2
c_{pin}	0.0015	0.009	0.0015	J / m^2
Re	1.8575E-4	1.6005	5.1081E-4	non-dim.
Ca	1.6495E-3	1.9396E-4	4.5361E-3	non-dim.
α	1.8759E-5	0.19675	8.4772E-5	non-dim.
β	36.356	66.383	59.744	non-dim.
K_{hys}	0.49	0.071	0.82	non-dim.
$\theta_{b,0V}$	107.35	111.62	107.35	degrees
$\theta_{b,50V}$	68.46	-	-	degrees
$\theta_{b,65V}$	-	70.01	64.32	degrees

Table 5.3: Simulation Parameters for the experiments in Sections 5.3 (Moving Glycerin), 5.4 (Joining Water), and 5.5 (Splitting Glycerin) using the Variational Method. Note the large viscosity of glycerin. We assumed a 90%/10% glycerin/water mixture for the glycerin experiments. $\theta_{b,0V}$, $\theta_{b,50V}$, $\theta_{b,65V}$ are the contact angles on the bottom plate of the EWOD device at 0, 50, and 65 volts, respectively.

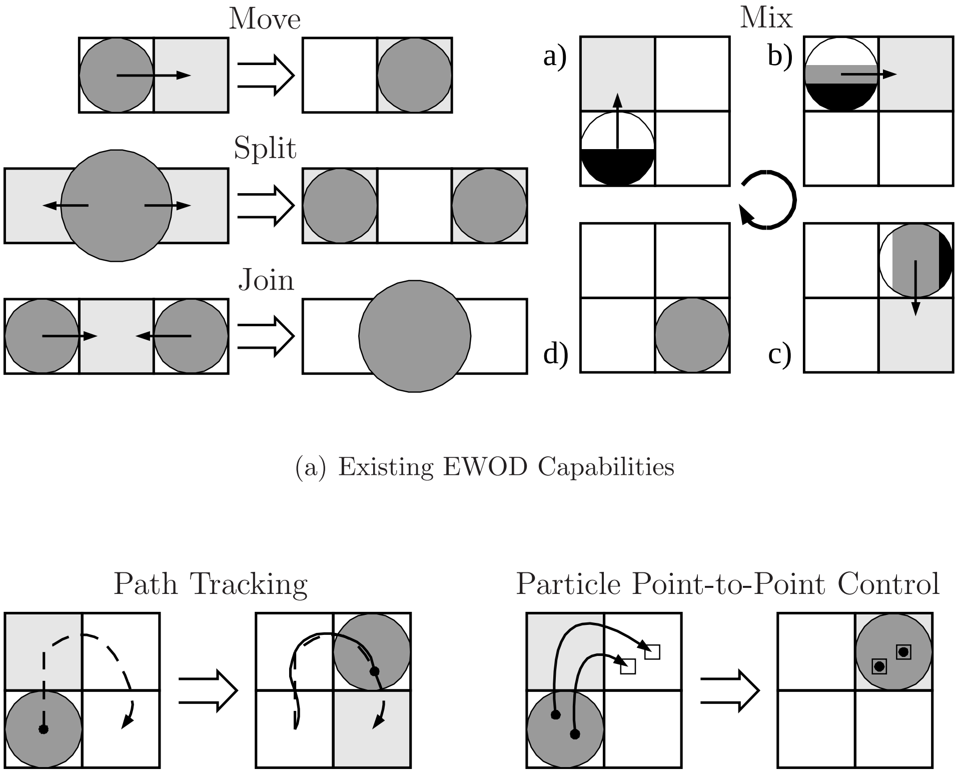
Chapter 6

Controlling EWOD

6.1 Introduction

This chapter shows that Electro-Wetting-on-Dielectric (EWOD) systems contain enough control authority to steer individual particles on trajectories inside the liquid drops. It is possible, for example, to actuate the available electrodes surrounding a single droplet in such a way that the resulting fluid flow inside the drop will carry a particle around a figure '8' path or will carry two particles along separate trajectories (see Figure 6.1).

Steering particles inside droplets introduces another level of functionality into electro-wetting systems. By being able to steer individual particles inside droplets to 10 micrometer resolution, it will be possible to precisely place cells under or on top of localized sensors, to stretch out DNA strands (by steering beads attached to the two ends of the strand in different directions), to move particles from one location to another at rates much faster than those created by diffusion, to steer and sort particles within a single drop, or to ensure that certain particles remain in one drop while other particles are steered to a drop that is being split away. In essence, our control methods allow electrowetting systems to achieve some of the same capabilities as laser tweezers [6], [72], [32], although only in two spatial dimensions and with less accuracy.



(a) Existing EWOD Capabilities

(b) New Particle Steering Capabilities

Figure 6.1: The EWOD system manipulates fluids by charging a dielectric layer underneath the liquid that effectively changes the local surface tension properties of the liquid/gas interface creating liquid motion. Existing (move, split, join, and mix) capabilities of electrowetting devices are shown schematically (see [89], [79], [131], [107], [111], [28], [55]) above the new particle steering capability developed in this thesis. The view is from overhead the EWOD device. Shaded circles represent droplets of liquid. Squares are electrodes, where the lighter shading indicates the electrode is on. Directed lines specify the direction of motion. The multi-shaded droplet shows the diffusion and mixing of two chemicals; here mixing is enhanced by the fluid dynamics created inside the droplet due to its imposed motion.

Steering results are demonstrated using a model [142] of fluid flow in the UCLA EWOD system [89], [29] and the simulation tools developed in the previous chapters. This model of EWOD fluid dynamics includes surface tension and electrowetting interface forces, Hele-Shaw type 2-phase fluid flow, and the essential loss mechanisms due to contact angle saturation, contact line pinning, and the related mechanism of contact angle hysteresis. The model has one free parameter associated with the frictional effect due to hysteresis, but all other parameters are either derived from first principles or taken from experimental data. And the model is validated against experimental results for the UCLA devices (see [142] and Chapter 5).

To experimentally demonstrate particle steering in the UCLA EWOD system would require real time implementation of the least squares control algorithm, a vision system to find the location of the particles and to track droplet shapes in real time, and integration and experimental validation of the feedback closed loop architecture shown in Figure 6.2. Some of these tasks have already been demonstrated in Shapiro, et al [4], [5], which developed a particle tracking vision system, a real time control algorithm implementation, and closed loop feedback integration for steering of particles in a micro fluidic system driven by electro-osmotic (as opposed to EWOD) actuation. It remains to create a real time control implementation for the EWOD system, to implement our vision algorithm on the UCLA devices, and to achieve the feedback control system integration. However, the size of the vision system, which is currently a microscope and a camera with an image algorithm implemented on a digital signal processing chip, can be reduced to an on-chip contact imager as demonstrated in [54]. This permits the feedback EWOD system, with vision feedback, to remain portable and hand-held.

Section 6.2 describes the particle steering control task and algorithm development. This is followed by some numerical demonstrations in Section 6.3 that show which particle trajectories can and cannot be achieved.

6.2 Particle Steering Control Algorithm

We consider neutral particles that are simply carried along by the (vertically averaged) planar fluid flow. Thus, a particle at the location $\vec{x} = (x, y)$ will simply follow the velocity of the fluid at its location

$$\dot{\vec{x}} = \vec{u}(\vec{x}), \tag{6.1}$$

where $\vec{u} = (u, v)$ is the flow field from equation (2.19) (see Section 2.4.1) and the overhead 'dots' denote derivatives with respect to time. The pressure gradient in equation (2.19) can be thought of as a control for manipulating the velocity of the particle or particles. It is by controlling the pressure gradient field inside the droplet, by changing the pressure on the interface via electrode voltages (see equations (2.16), (2.17), (2.18) and Figure 2.6), that we can achieve particle steering using actuators already available in the EWOD device. Steering of multiple particles along complex trajectories requires controlled actuation of the electrode voltages. Therefore, the control problem is to find an electrode voltage sequence that creates a temporally and spatially varying flow field that will carry all the particles along their desired trajectories.

The control problem described above is a trajectory-tracking problem: we seek to find the control inputs that will cause the system (in this case the particle positions) to follow a desired trajectory. A naïve inspection of the equations of motion, especially (6.1) for the particle dynamics, would suggest that the control problem is standard in linear control theory and a linear quadratic regulator (LQR) tracking controller [90] could be used. However, the particle motion depends on the droplet shape and on the number of electrodes that the droplet overlays at any given moment. This information is not known a priori, which means that an LQR cannot be used. For this reason, we do local estimation and control at each time step of our simulation using a least squares framework to compute the necessary pressure boundary conditions and then compute the electrode voltages that will achieve these boundary pressures (see Figures 6.3 and 6.5). Any particle deviation from the desired trajectory that may arise from thermal fluctuations, external disturbances, and actuation errors is corrected using feedback of the particle's position. Figure 6.2 gives a diagram of the closed loop feedback architecture.

Our particle steering algorithm proceeds as follows:

1. Initialization: Represent the desired trajectory of each particle as a set of points connected by straight line segments.

2. Sensing: Feed back the particle position data and the location of the droplet boundary to the control algorithm (as would be provided by the vision sensing system).

3. Control algorithm part A: Choose the desired velocity directions of each particle so that the particles will move towards and then along the desired tra-

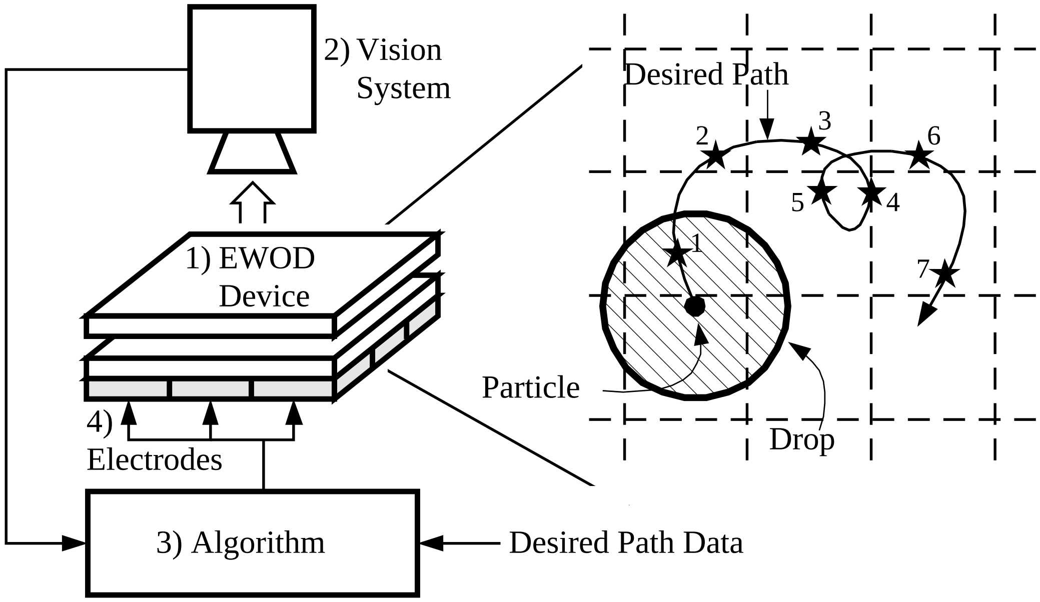


Figure 6.2: Particle steering closed loop feedback control architecture. 1) The EWOD device is observed by 2) an image system (a microscope/camera or an on-chip contact imager) which transmits information to 3) a computer or chip that contains 3a) an image processing algorithm to identify droplet shapes and the location of the particles and 3b) a control algorithm that computes the actuator voltages that will move the particles from where they are to where they should be, and 4) these actuation voltages are then applied on the EWOD device. The loop repeats at each time step to steer the particles along their desired trajectories. The zoomed overhead view of the EWOD device (at right) shows a single droplet with one particle floating inside. The curvy line indicates the desired path of the particle. In our control algorithm, we sample the trajectory by many points (only seven points are shown here; see numbered stars 1-7).

jectories.

4. Control algorithm part B: Solve a least squares problem for the necessary voltage actuation to induce a pressure gradient field that will create a flow field to carry the particles along the desired directions obtained in step 3.

5. Actuate: Apply the computed control voltages at the current time step of our simulation and advance the simulation to the next time step. This updates the droplet shape and particle positions. Then go back to step 2 and repeat the feedback control loop.

The algorithm details are described below.

6.2.1 Algorithm Initialization

We represent the desired trajectory curves for each particle as a fine sampling of points connected by straight lines. The points are indexed in the order in which the particles should follow them (i.e. the trajectory is parameterized; see Figure 6.2). Complicated trajectories are broken up into separate segments for ease of particle tracking (see Section 6.2.3). For simplicity, only one particle and trajectory is considered in the following sections. Multiple particle steering is discussed in Section 6.2.5.

6.2.2 Particle Position and Droplet Boundary Sensing

We need to know the shape and position of the droplet as well as the position of each particle in order to apply our control algorithm. At the beginning of each time step, we obtain the position of the particle and the location of the droplet boundary using feedback through a vision system (see Figure 6.2). The issues of integrating a vision system with an EWOD device are not considered here. For the purposes of this thesis, the particle positions and droplet shape information are taken directly from the simulation.

6.2.3 Compute the Desired Direction of Particle Motion

Next, the desired direction of motion for the particle is chosen to be a unit vector that points from the particle's current coordinates towards one of the trajectory points. Since maximum forcing of the pressure gradient is used to drive the particle in the desired direction (see Figure 6.4), it is necessary to choose a trajectory point that is just out of reach of the particle for the current time step. Otherwise, it is possible that the particle could overshoot trajectory points and trace out an unwanted zigzag path around the trajectory.

Hence, we find the target trajectory point by first finding the closest trajectory point to the particle. Then, using the trajectory parameterization (i.e. the index list; see Figure 6.2), we look ahead after the closest point and choose the target to be the first trajectory point that is out of reach of the particle. This ensures the particle will move forward along the trajectory and not zigzag. If the closest trajectory point is the last point of the trajectory, then the particle aims for the last point.

For a self-intersecting or extremely curvy trajectory, it is possible that the particle could become stuck in a loop and not travel the entire trajectory. We resolve this issue by breaking the trajectory into smooth segments that do not intersect and only allow the particle to 'see' one segment at a time. As a result, the particle follows one piece of the trajectory until it reaches the end, where our algorithm switches to the next segment. Therefore, without loss of generality, we assume in the following subsections that the trajectory consists of just one segment.

The forcing of the particle is created by the pressure gradient. And the desired unit vector discussed above determines the direction of forcing. This unit vector is used in the next section to calculate the pressure boundary conditions needed to realize the pressure gradient that will move the particle in the desired direction.

6.2.4 Least Squares Solution for the Boundary Conditions

Figure 6.3 shows an overhead view of a sample droplet in the EWOD device containing a single particle. The current drop shape overlaps four electrodes, hence four actuators are available to move the single particle. In each of the four cases, only one electrode is on; the other electrodes are off. The arrows inside the droplet show the fluid flow for each of the four voltage actuations. The big dot represents the particle with a thick arrow indicating the negative direction of the pressure gradient at the particle location (note that the fluid flows opposite to the pressure gradient). Our algorithm centers on the idea of taking an appropriate linear combination of pressure gradients in Figure 6.3 to make the particle (or particles) move in the direction(s) we want at a particular time step. This will directly correspond to

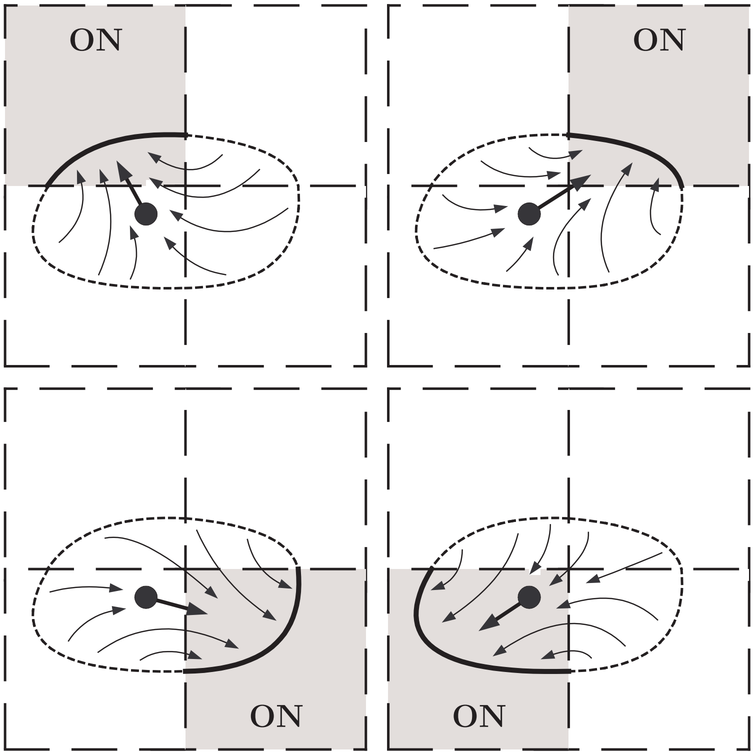


Figure 6.3: Linear combination of pressure gradients for a single droplet overlaying four electrodes (small dashed squares). The diagram above shows a droplet in an EWOD system with four different instances of voltage actuation. In each instance, only one of the four electrodes is on. The particle floating inside the droplet (black dot) has a thick arrow indicating its direction of motion for each single electrode actuation. These arrows actually represent the opposite direction of the pressure gradient when a unit pressure boundary condition is set on the thick curve that overlays the shaded electrode, with zero pressure boundary conditions everywhere else. The thin curvy arrows show the fluid flow inside the droplet. Since the pressure field obeys Laplace's equation (2.15), it is linear and we can make the particle move in any desired direction by taking an appropriate linear combination of the four possible boundary conditions given above.

finding the right combination of electrode voltages at every time step to realize the desired particle motion (or motions).

First, given the current droplet configuration, we solve (2.15) for the pressure field inside the droplet for a single active electrode. The pressure boundary conditions are defined to be 1 on the droplet boundary that lies over the active electrode and 0 everywhere else (see Figure 6.3). From the pressure solution, the pressure gradient at each particle's position is computed. After repeating this for each electrode, we obtain a matrix of pressure gradients

$$G = -\begin{bmatrix} \nabla P_1(\vec{x}_1) & \cdots & \nabla P_N(\vec{x}_1) \\ \vdots & \ddots & \vdots \\ \nabla P_1(\vec{x}_m) & \cdots & \nabla P_N(\vec{x}_m) \end{bmatrix}, \tag{6.2}$$

where $\vec{x}_j = (x_j, y_j)$ are the coordinates for the jth particle. Each column of pressure gradients $\nabla P_k(\vec{x}_j)$ in the matrix corresponds to a single active electrode; each row corresponds to a single particle. The total number of particles is m, and the number of available electrodes is N. The minus sign accounts for the direction of particle motion.

Next, given the desired pressure gradient at each particle's location in the droplet, we wish to find the appropriate boundary conditions to realize it. Because Laplace's equation for the pressure (2.15) is linear regardless of the droplet shape, solutions for single active electrodes can be combined linearly to obtain the pressure gradient field due to many active electrodes. This reduces our problem to solving a linear system

$$G\xi = b \qquad \xi = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_N \end{bmatrix} \qquad b = \begin{bmatrix} \vec{b}_1 \\ \vdots \\ \vec{b}_m \end{bmatrix}, \tag{6.3}$$

where $\vec{b}_j$ is a 2x1 vector representing the desired direction of motion for the jth particle and ξ is the vector of boundary values that will achieve b. We set $\vec{b}_j$ equal to the unit vector in Section 6.2.3 that represents the desired direction of motion for the jth particle. If $2m \geq N$, the number of particle degrees of freedom is greater than the available actuators and (in general) a least squares solution of (6.3) is required to obtain the best fit of actuations ξ. Otherwise, it is a pseudo-inverse problem and has a solution as long as the matrix has full row rank [133].

We solve (6.3) for ξ using the singular value decomposition (SVD) [133]. This is not an expensive computation since the matrix G is not large. In addition, each component of the solution vector must be made to satisfy an inequality constraint

$$\xi_{min} \leq \xi_j \leq \xi_{max}, \;\; 1 \leq j \leq N, \tag{6.4}$$

where ξ_{min} and ξ_{max} are the minimum and maximum values that the pressure boundary condition can be for any electrode. These constraints come from the limitations of varying the contact angle (i.e. contact angle saturation [124], [138], [139], [103]). Hence, ξ_{min} and ξ_{max} are related to the minimum and maximum contact angles achievable in the EWOD device (see equations (2.17) and (2.18)).

$$\begin{aligned} \xi_{min} &= -\frac{L}{H}[\cos(117.0°) + \cos(90.0°)], \\ \xi_{max} &= -\frac{L}{H}[\cos(117.0°) + \cos(117.0°)]. \end{aligned} \tag{6.5}$$

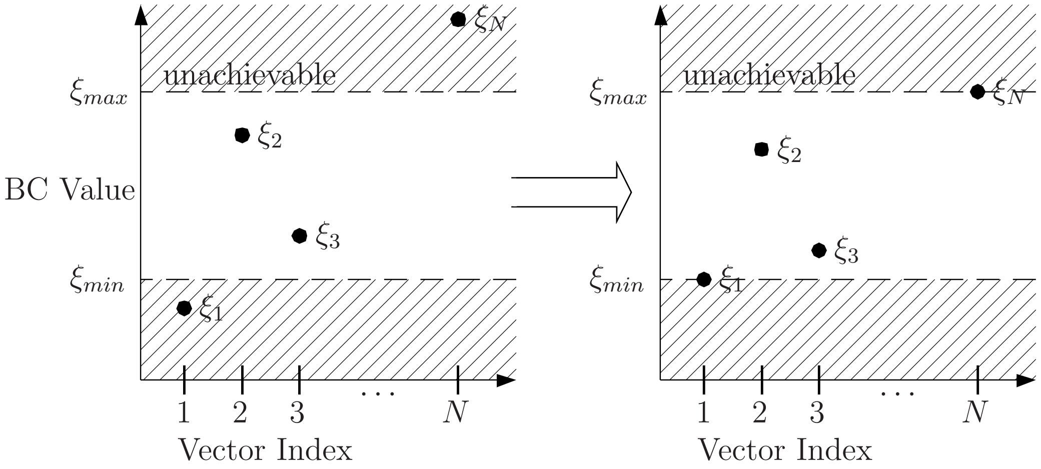


Figure 6.4: Linear transformation of boundary conditions; an example of satisfying the boundary condition constraints. On the left, the components of the solution to (6.3) are plotted with the maximum and minimum constraint bounds denoted by dashed lines (see equation (6.4)). On the right, the components have been linearly mapped to enforce the constraints. This introduces a scaling factor into the linear system (6.3), which affects the magnitude of the desired direction vector b (i.e. the magnitude of the force acting on the particles). In effect, this causes the particle to be forced as much as possible in the desired direction.

In order to satisfy equation (6.4), we take the solution ξ to (6.3) and transform each of its components so that the full dynamic range of boundary forcing is utilized (see Figure 6.4).

With this new transformed ξ, we know what the pressure boundary values should be to realize the desired pressure gradient field. But it is not possible to exactly enforce ξ because we cannot directly control the planar curvature term κ

in (2.16). For a circular droplet, the planar curvature term is constant and has no effect on the pressure gradient field [148]; hence, it can be ignored. Using (2.17) and (2.18), it is straightforward to compute the contact angles needed to implement ξ. For non-circular droplets, we use the same procedure. It is not reasonable to use the planar curvature term in our control algorithm because it involves second derivatives of data that cannot be accurately measured in experiments [112]. Instead, we view it as a small error to the desired directional forcing of the particles. This error grows as the droplet deviates from being a circle. This is not a problem for particle steering for two reasons. First, the linear transformation of the boundary conditions in Figure 6.4 ensures maximum forcing of the particle. Thus, the relative magnitude of the error due to the planar curvature κ is minimized. Second, any particle trajectory tracking errors that may occur are corrected through our feedback system (see numerical simulations in Section 6.3). However, the planar curvature does limit the type of trajectories that the particles can follow (also see Section 6.3).

Finally, given that pressure on the boundary is directly related to the local contact angle (2.18), we use experimental data for the contact angle versus voltage characteristics of the EWOD device [29] to compute the electrode voltages needed to achieve the boundary pressures ξ (see Figure 6.5). In general, there will be some uncertainty about the device parameters. In this thesis, we do not consider adaptive or robust control strategies to deal with uncertain device parameters, but rather demonstrate the potential for particle control and separation in EWOD devices.

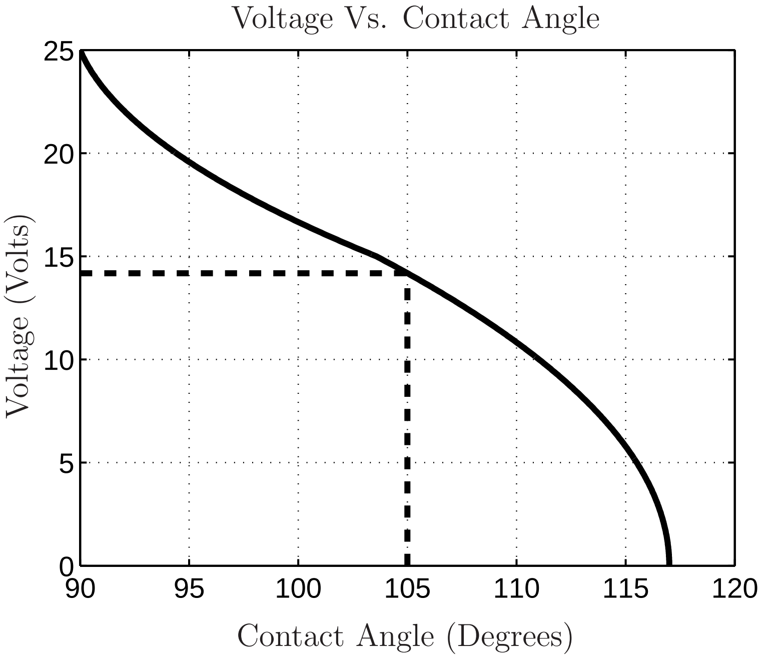


Figure 6.5: Voltage versus contact angle with contact angle saturation. Here we show the inverse mapping of the thick black line in Figure 2.6 that represents the contact angle variability of the EWOD device. The plot shows how to compute the voltage needed to actuate a specific contact angle. The dashed line depicts the mapping from a 105.0° contact angle to a voltage of about 14.2 volts. We use this in our control algorithm for estimating the necessary actuation voltages (see Section 6.2.4).

6.2.5 Advance Simulation and Update Particle Position

Our simulation advances to the next time step after using the voltages computed in Section 6.2.4 to solve for the induced pressure and velocity fields. The velocity field is then used to update the position of the particle (see Figure 6.6). The scaling described in Figure 6.4 ensures the particle will be forced as fast as possible along the desired direction. Our algorithm runs by going back to Section 6.2.2 and repeating for each time step.

Multiple particle steering is easily handled by applying the above discussion to each particle and its respective trajectory. The only change is that the linear system in Section 6.2.4 has more rows to accommodate the extra particles. If the number of electrodes is limited, then this can adversely affect the controllability we have. A single particle can be made to track interesting trajectories with enough electrodes (see Figure 6.7 and Figure 6.10). Two particles can be controlled for simple trajectories as shown in Figure 6.13. For more than two particles, most complex trajectories cannot be tracked. Section 6.4 discusses the merits and limitations of our method.

6.3 Controlling Motion and Splitting

This section presents results demonstrating basic particle steering control using our experimentally validated simulation. A 3x3 electrode grid is used to actuate and control the droplet where each square electrode is 1.4 millimeter on a side. We present seven cases, with different levels of controllability, using both the level set

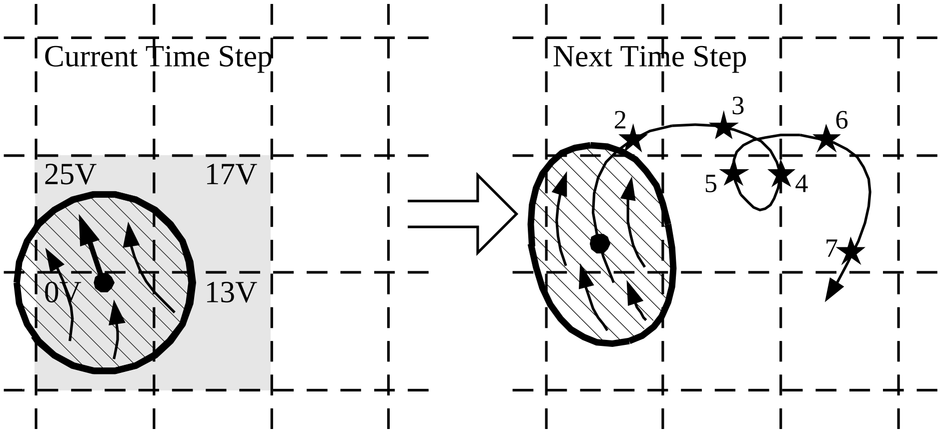


Figure 6.6: EWOD particle steering control algorithm update. The droplet configuration from Figure 6.2 is shown on the left. The direction of motion for the particle is toward the trajectory point that is just out of reach for the current time step. This control strategy ensures the particle will move as fast as possible and stays close to its desired trajectory. On the left, the shaded electrodes contain the voltages needed to move the particle in the desired direction. These voltages are computed by the least squares solution in Section 6.2.4, the constraint map in Figure 6.4, and by the voltage versus contact angle curve in Figure 6.5. The varying voltage grid induces a pressure gradient field inside the droplet such that minus the pressure gradient at the particle's position is pointing along the desired direction of motion. This moves the droplet and particle along the trajectory to the next time step.

method and variational method and make note of their differences. This is followed by a discussion of the possibilities and limits of our control algorithm. The voltages generated by our algorithm are reasonable and are within the limits of the UCLA device discussed in [29].

6.3.1 Figure ‘8’ Path

Figure 6.7 shows a droplet moving in a way that makes a particle floating inside follow a figure ‘8’ path. A circular droplet starts on the center electrode with a particle resting in the center of the droplet. The dashed curve represents the desired trajectory, which is made up of a fine sampling of points. Two segments are used to represent the trajectory because of the self-intersection (see Section 6.2.3). The voltages on the electrode grid are actuated using the algorithm in Section 6.2, which causes the particle to move forward along the trajectory. For this case, the droplet always overlaps enough electrodes to allow it to be controlled in a way that keeps the particle moving on the figure ‘8’ path. The particle never deviates more than 20 micrometers from the desired trajectory.

Figure 6.8 presents the same case using the variational simulation without the pinning model; Figure 6.9 includes the pinning model.

6.3.2 Angular Path

In Figure 6.10, a particle is shown following an angular path that is represented by five separate straight-line segments. This is to prevent the particle from rounding

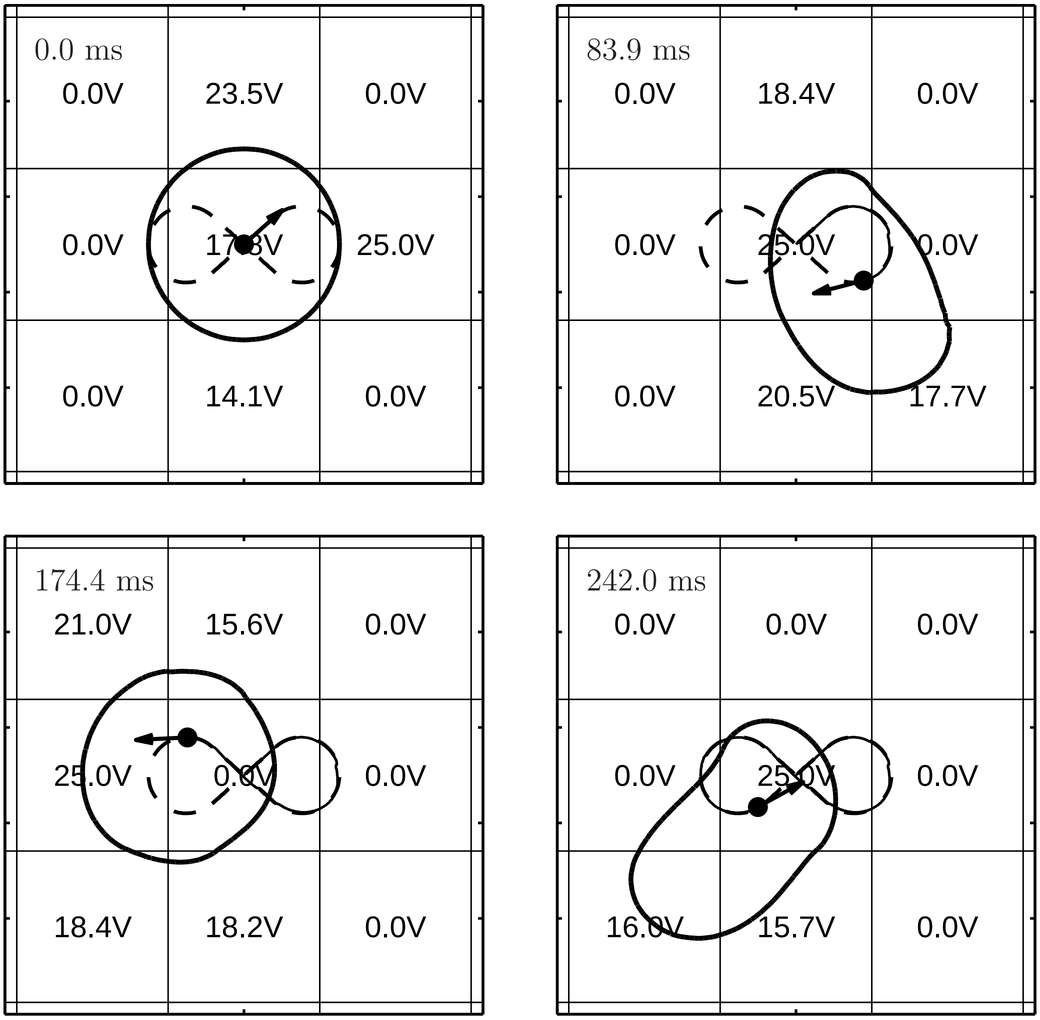


Figure 6.7: Particle following a figure '8' path (level set method). An initially circular droplet (denoted by the closed black curve) lies on a 3x3 grid of electrodes (denoted by straight lines). The dashed figure '8' curve is the desired path, and a large dot represents the particle with an arrow pointing in the desired direction of travel. The light solid curve that overlays the dashed curve is the actual path of the particle. The time-stamp is given in the upper left corner. The voltages on the grid are time varying in such a way as to keep the particle moving along the desired path with less than 20 micrometers deviation.

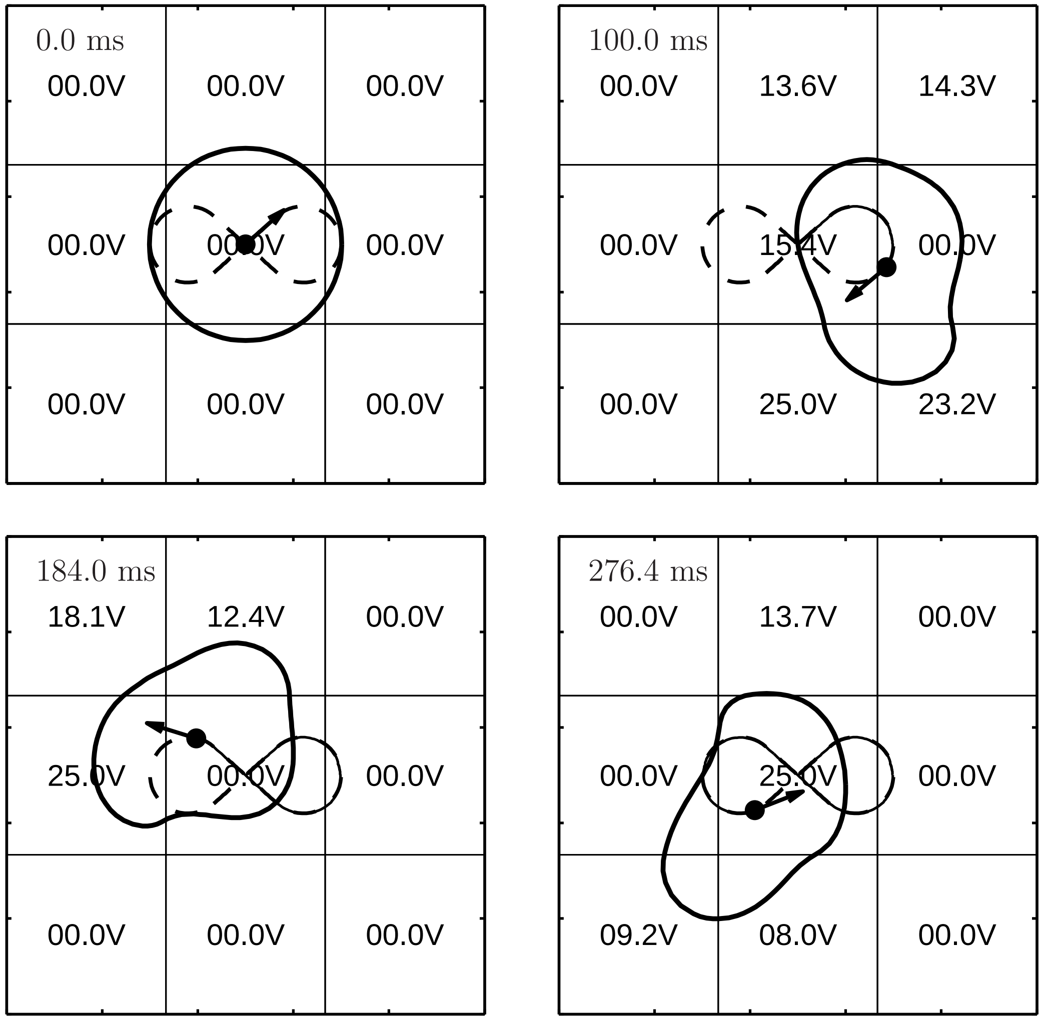


Figure 6.8: Particle following a figure ‘8’ path (variational method without pinning). Same format as Figure 6.7. The results are similar to Figure 6.7 with some differences in the droplet shape and total time to move the particle.

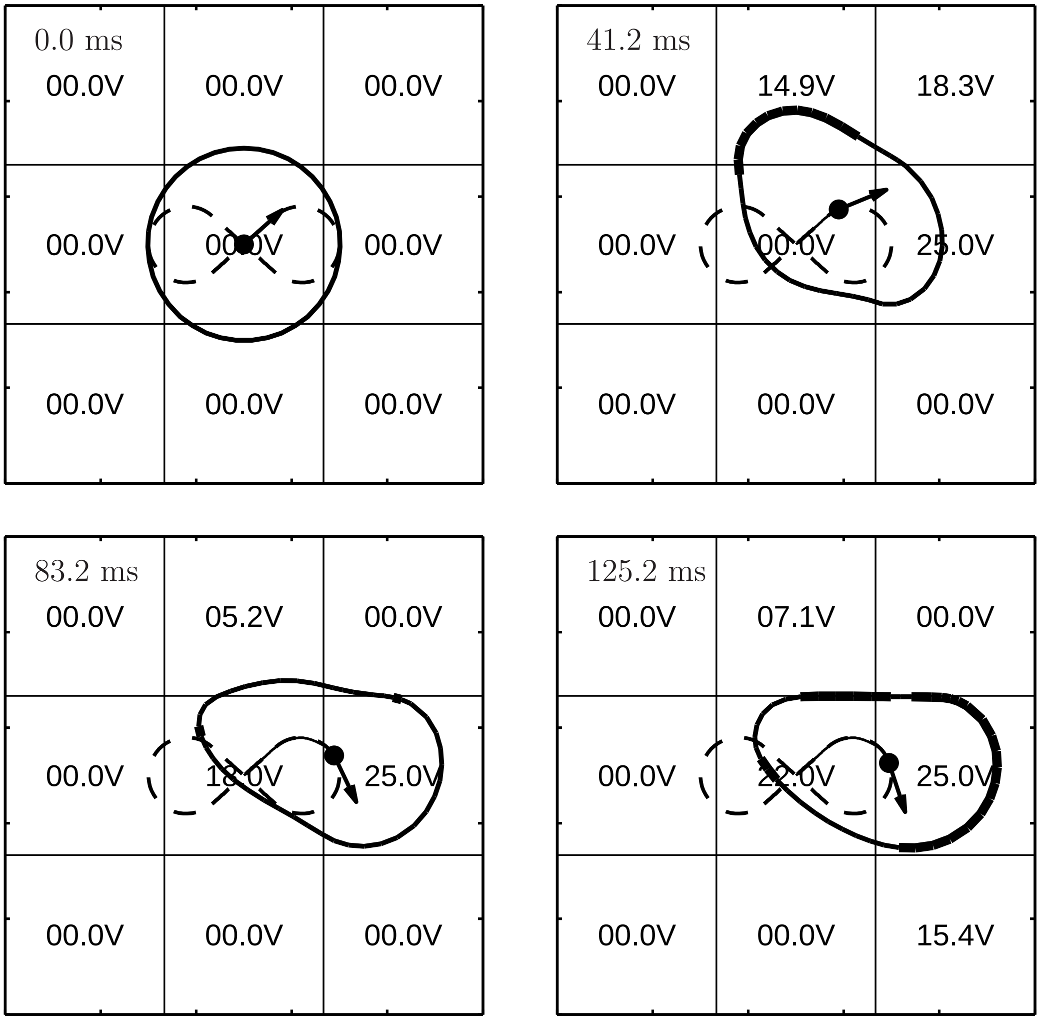


Figure 6.9: Particle following a figure '8' path (variational method with pinning). Same format as Figure 6.8, except the extra bolded parts of the droplet boundary denote pinned regions of the liquid-gas interface. In this case, the droplet becomes nearly stuck (shown in the last frame) and asymptotically approaches a completely pinned state. This is because the particle is slightly off of the desired way-point, so the algorithm is trying to force the particle back on but fails to apply enough forcing. Basically, the control algorithm cannot account for potential pinning of the droplet. It is possible to 'unstick' the droplet but would require a different algorithm.

off the corners as it travels along the trajectory. Just as in Figure 6.7, the droplet always overlaps enough electrodes to keep the particle on the path, with a maximum deviation error of 25 micrometers.

Figure 6.11 presents the same case using the variational simulation without the pinning model; Figure 6.12 includes the pinning model.

6.3.3 Arc and Point Paths

An example of two-particle control is shown in Figure 6.13. One particle is held stationary while the other moves along a circular arc. The trajectory for the stationary particle consists of a single point, which ensures that it stays close to that point. As the particle on the right follows the circular arc trajectory, the stationary particle oscillates around its desired position to within 10 micrometers. The droplet itself becomes deformed because of the limited actuators and the restrictive task of moving one particle and holding another still. This also prevents the particle on the circular arc from moving past the point shown in the last frame of Figure 6.13 and completing the arc.

Figure 6.14 presents the same case using the variational simulation without the pinning model; Figure 6.15 includes the pinning model.

6.3.4 Particle Separation

In Figures 6.16 and 6.17, we demonstrate particle separation. A droplet starts in the first panel with two particles spaced 0.31 millimeters apart. Both particles

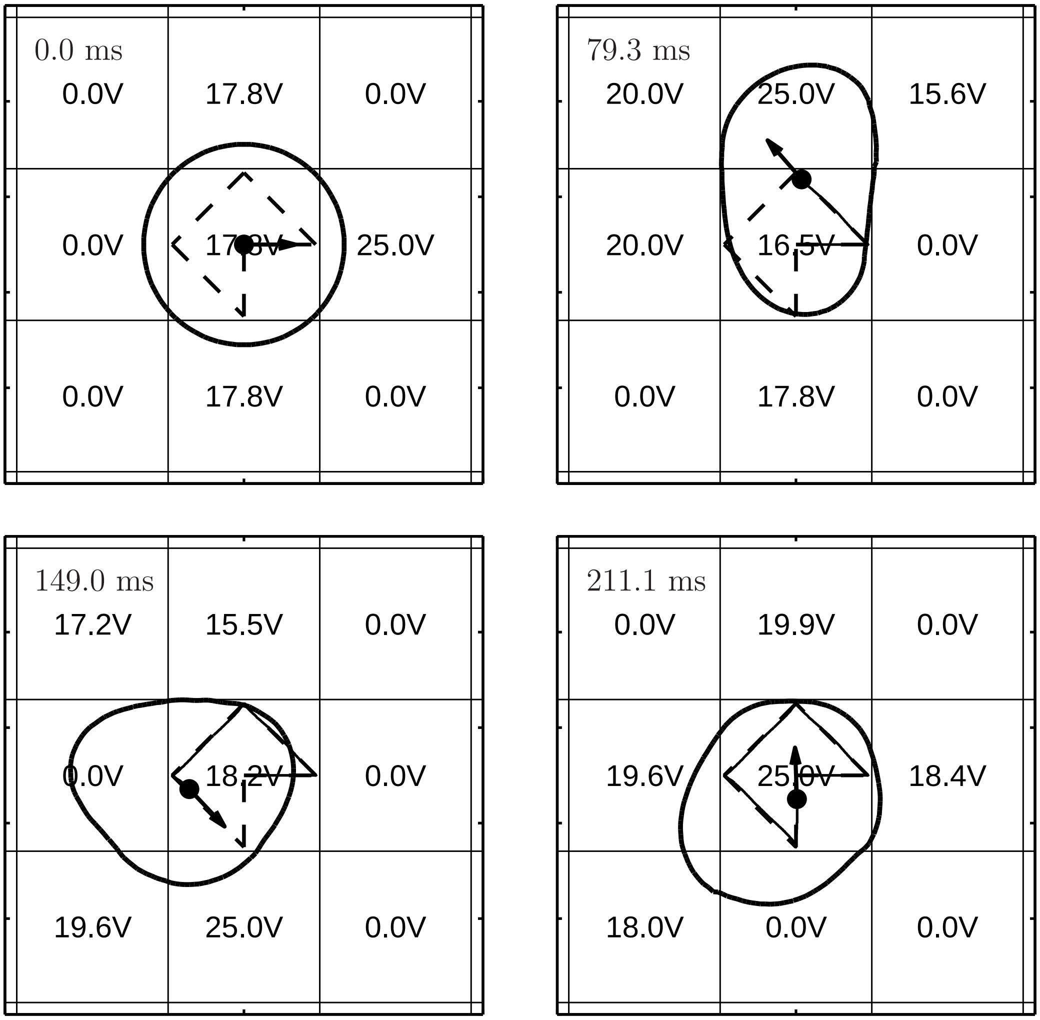


Figure 6.10: Particle following an angular path (level set method). Same format as Figure 6.7. The particle is able to track the trajectory very well, even at the corners.

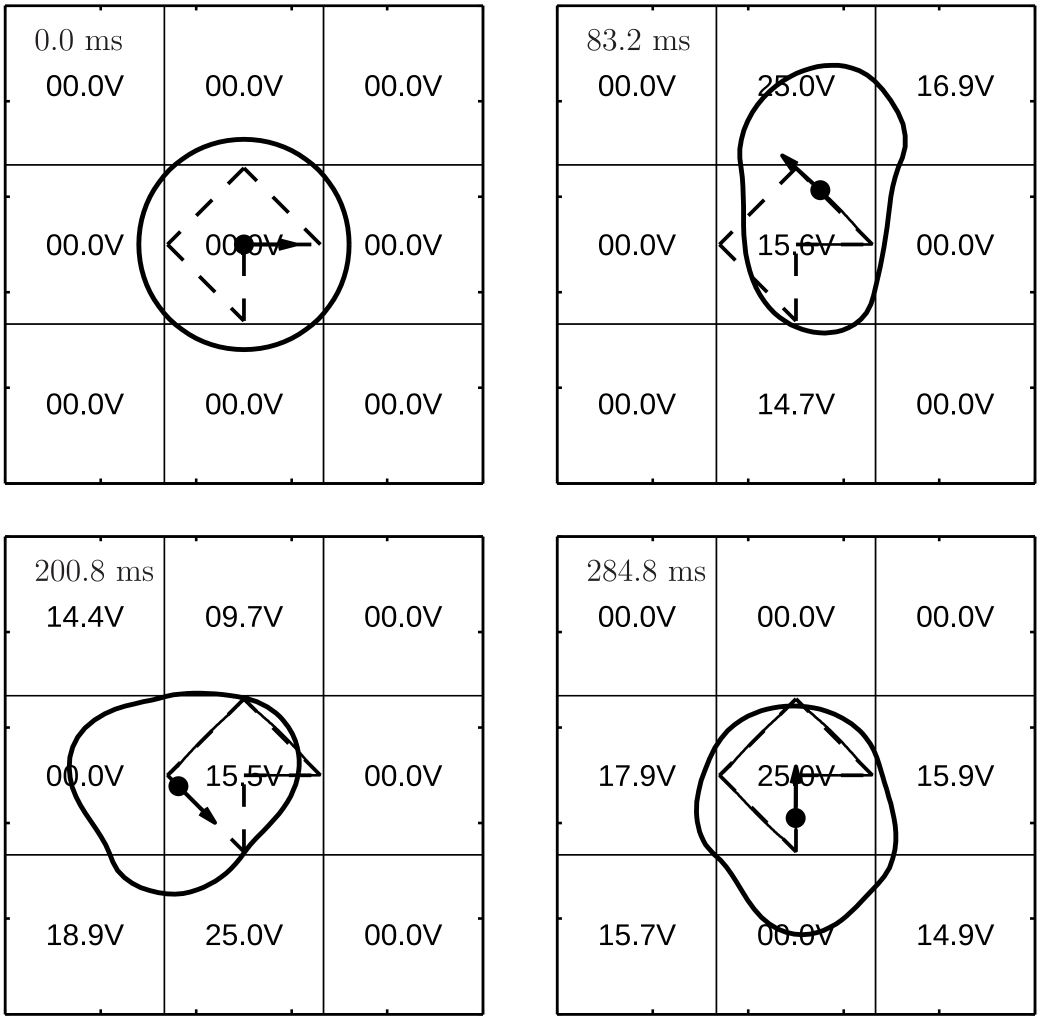


Figure 6.11: Particle following an angular path (variational method without pinning). Same format as Figure 6.8. The path of the particle appears to follow the desired trajectory slightly better than for the level set simulation shown in Figure 6.10.

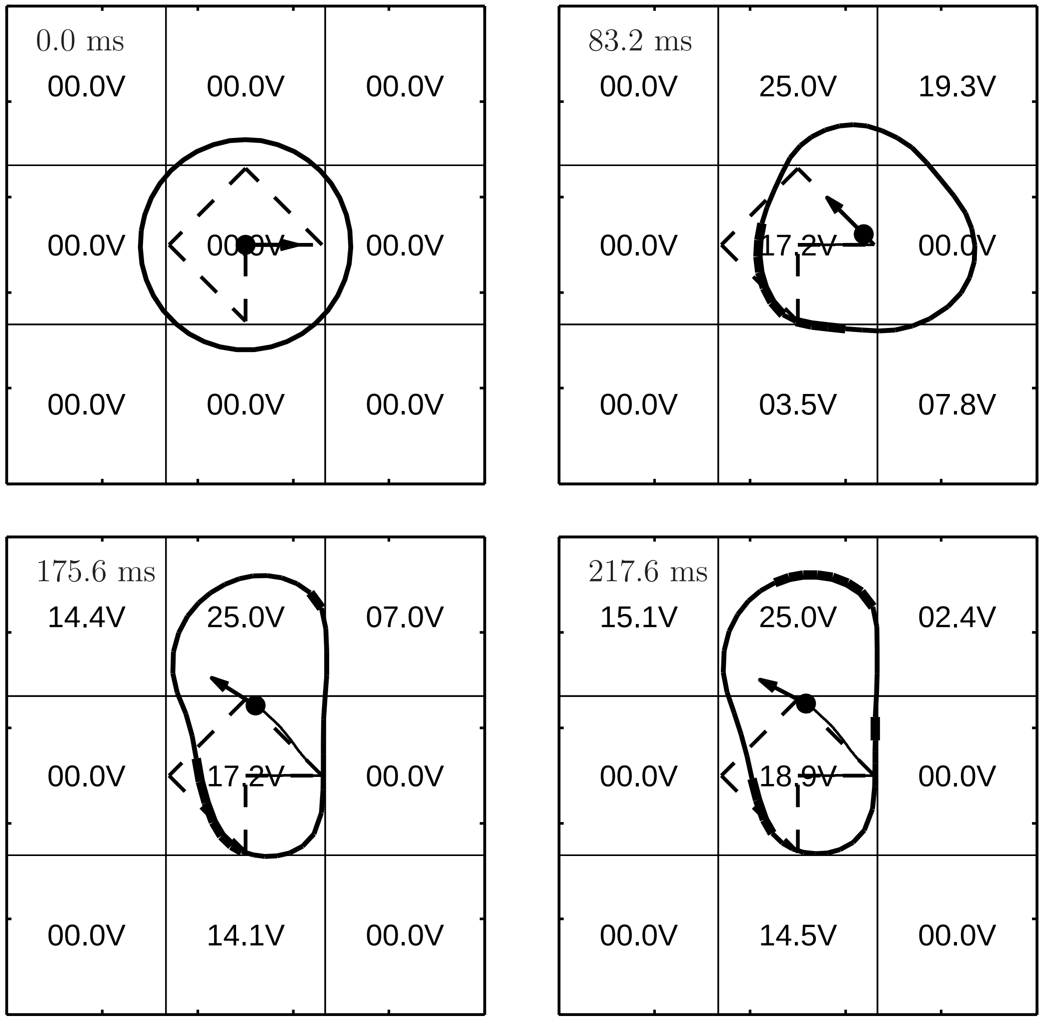


Figure 6.12: Particle following an angular path (variational method with pinning). Same format as Figure 6.9. The particle is able to follow the trajectory at first, but then begins to deviate (see later frames). Eventually, it becomes nearly stuck (shown in the last frame) and asymptotically approaches a completely pinned state. Explanation is the same as that given in Figure 6.9.

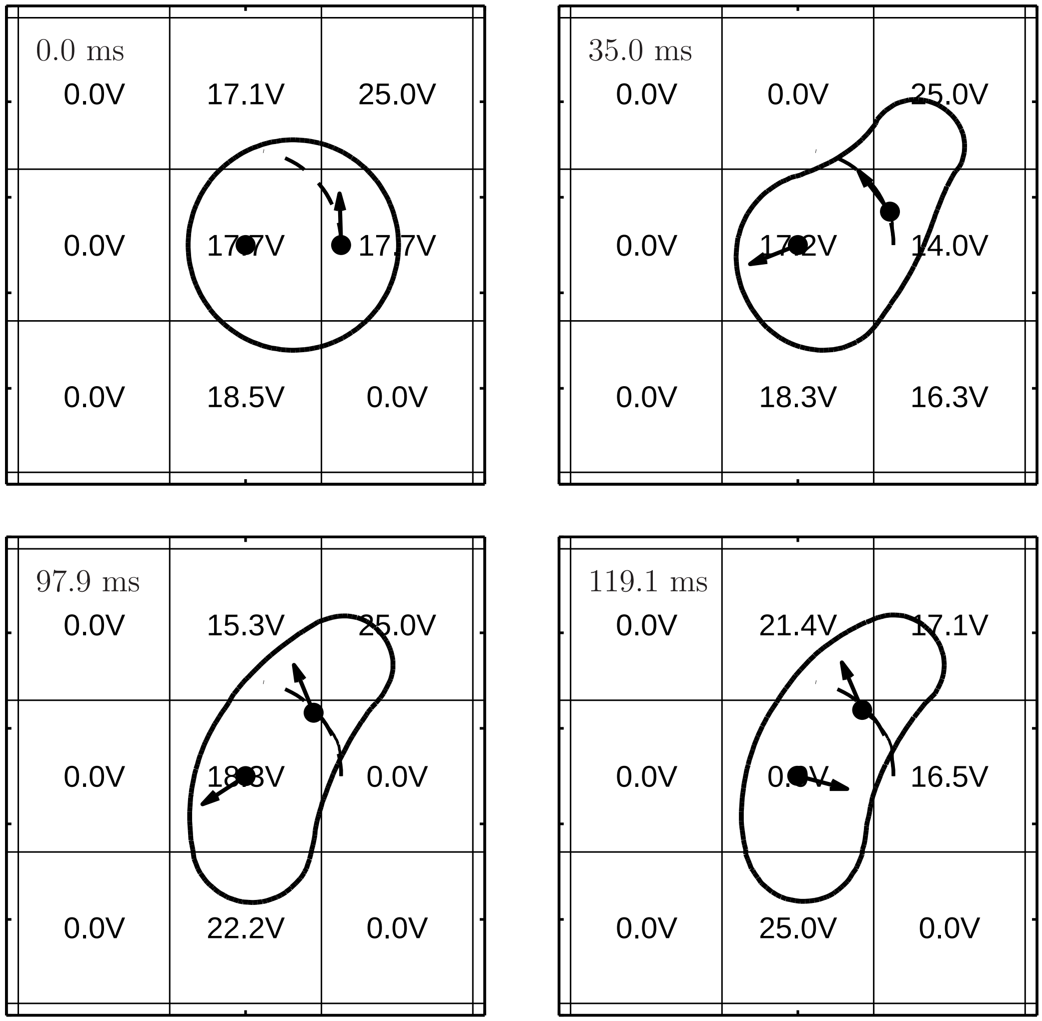


Figure 6.13: Two-particle control: one particle moves on a circular arc, the other is stationary (level set method). Same format as Figure 6.7. The stationary particle's trajectory is a single point. As the particle on the right follows the circular arc, the droplet distorts to accommodate both particle motions. Eventually, the algorithm is unable to continue the particle motion due to the restrictive condition of moving one particle while the other is held stationary, in addition to overcoming the curvature effect of the deformed droplet.

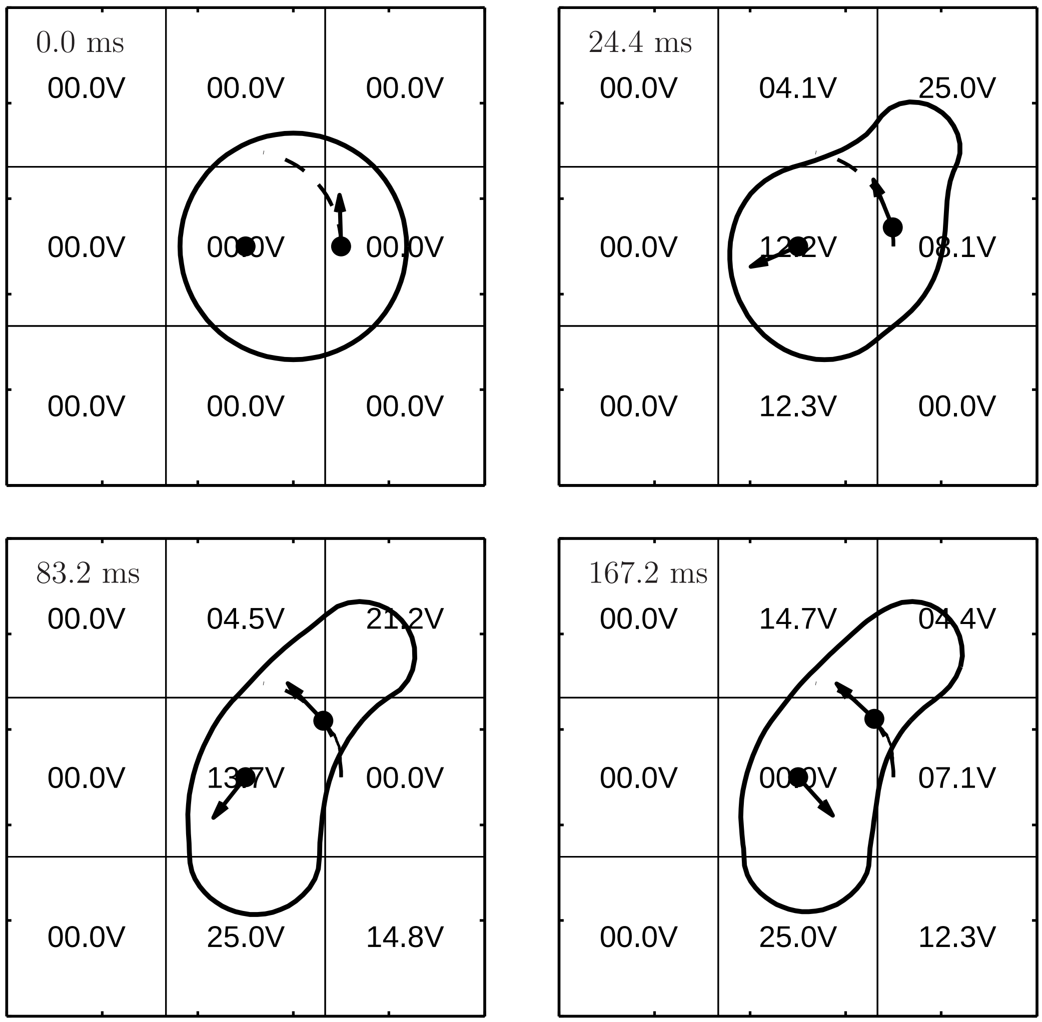


Figure 6.14: Two-particle control: one particle moves on a circular arc, the other is stationary (variational method without pinning). Same format as Figure 6.8. The results are similar to Figure 6.13.

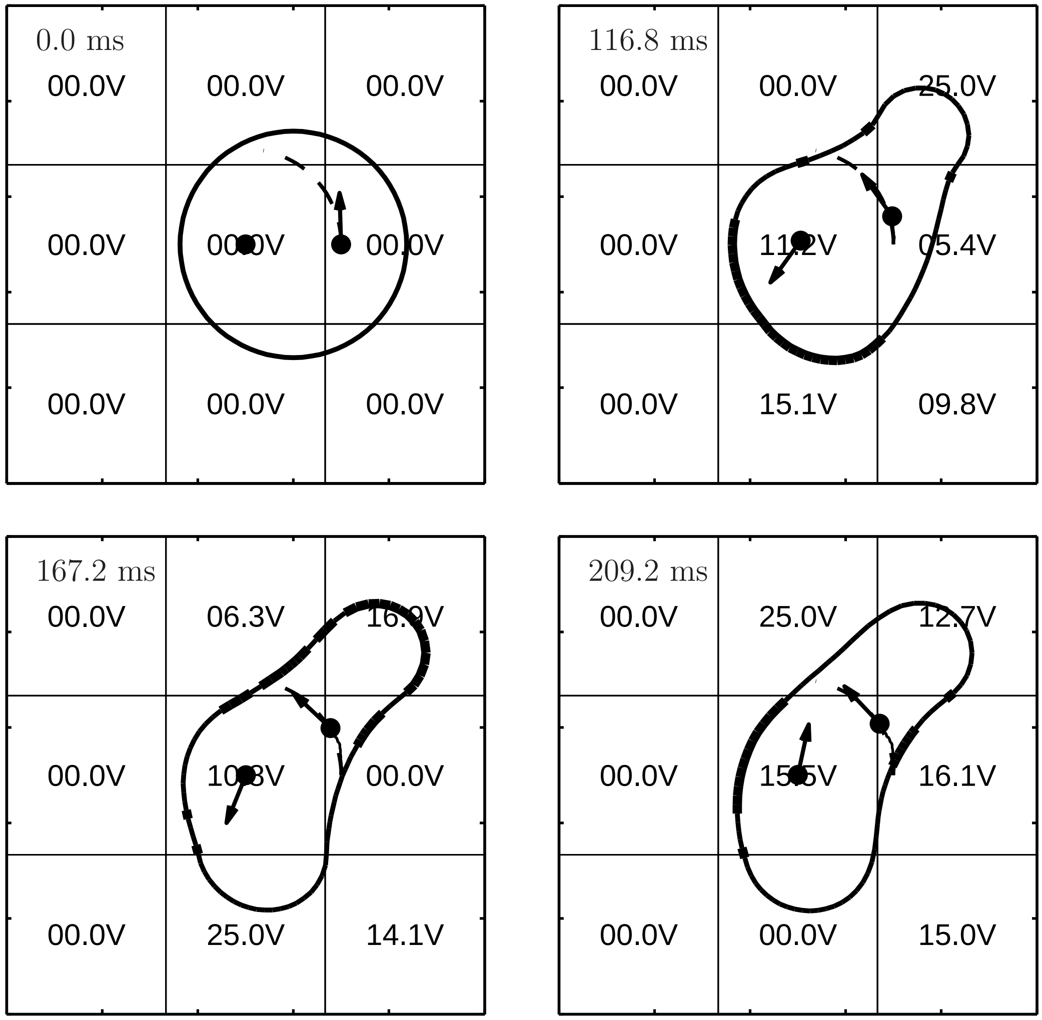


Figure 6.15: Two-particle control: one particle moves on a circular arc, the other is stationary (variational method with pinning). Same format as Figure 6.9. The results are similar to Figure 6.14 with some variation of droplet shape.

follow separate diverging trajectories designed to stretch the droplet and separate the particles. Once the particles near the ends of their trajectories (see third frame), our control algorithm turns off (fourth frame), and we command an open loop voltage of 25 volts on the middle left and right electrodes and zero volts everywhere else. This causes the droplet to split into two smaller drops, each of which contains a single particle. The reason for not using our control algorithm to complete the split is numerical instability. When both particles are in the lobes of the dumbbell shape of the pinching droplet, the available forcing at the particles' positions is fairly weak. This causes the condition number of the G matrix in (6.3) to degenerate and produce errors in the least squares solution. Therefore, we avoid this by commanding open loop voltages that we know will split the droplet (see Figures 5.10 and 5.11). Also, see Figure 6.24 for an example of how this numerical instability can affect particle control.

Figures 6.18 and 6.19 present the same case using the variational simulation without the pinning model.

Figure 6.20 presents a simulation of (attempted) particle separation with the pinning model.

6.3.5 Sine Wave Path

In Figure 6.21, a particle is shown traveling along a sine wave path. The particle is able to track the trajectory very well until near the end where there is a kink in the particle's path. The loss of tracking is because the droplet's shape

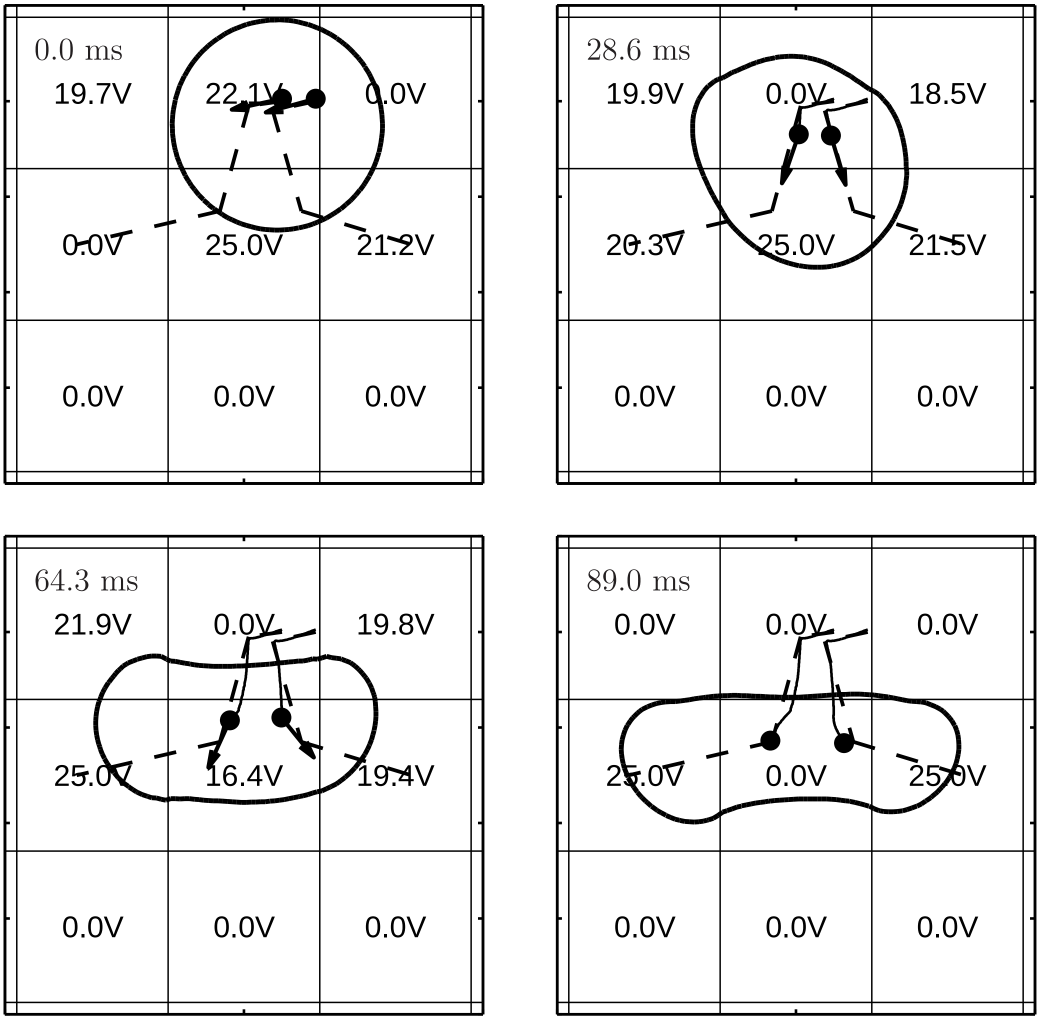


Figure 6.16: Two-particle separation into two satellite drops, part A (level set method). Same format as in Figure 6.7. Each particle first follows a trajectory that takes them away from each other. When there is sufficient distance between the two particles (see last frame), our control algorithm turns off and the separation is completed by applying open loop voltages that split the droplet. Simulation continued in Figure 6.17.

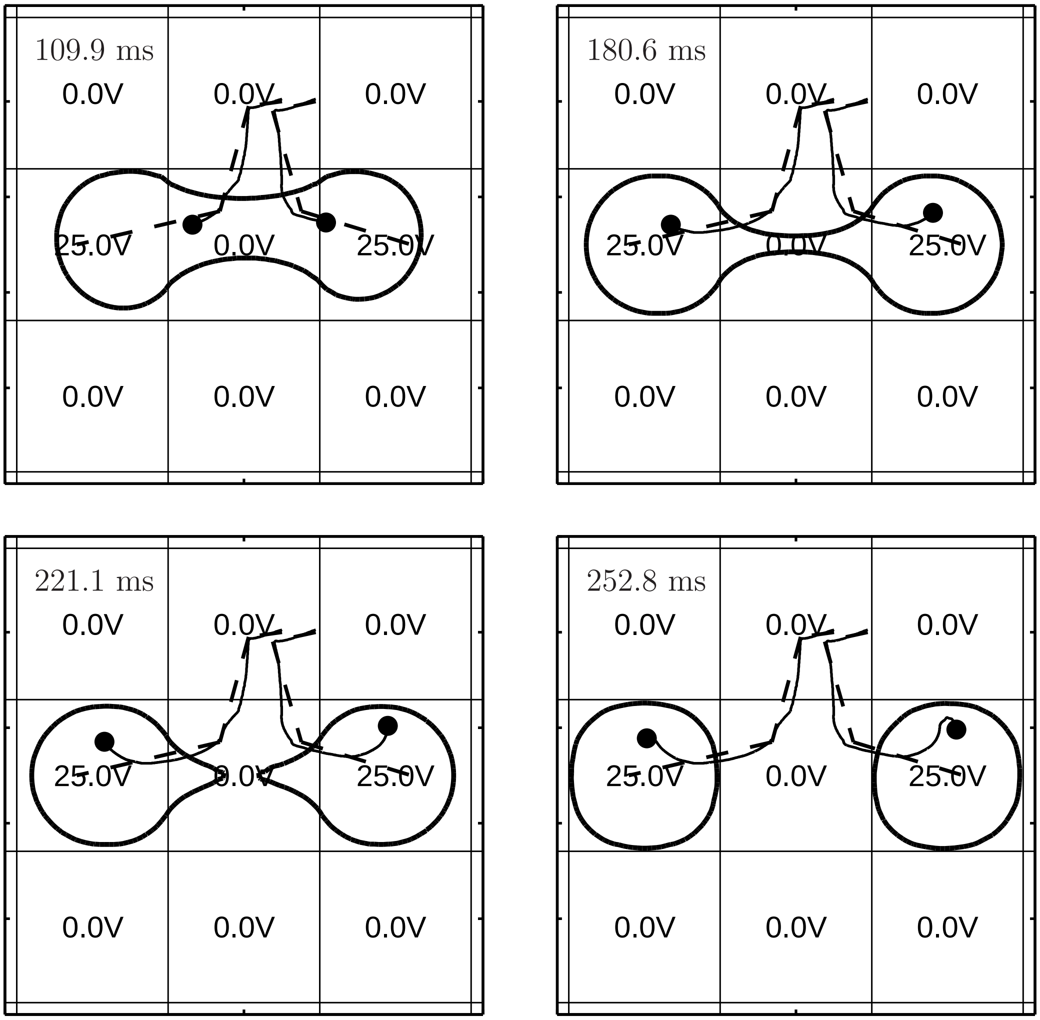


Figure 6.17: Two-particle separation into two satellite drops, part B (level set method). Same format as in Figure 6.7.

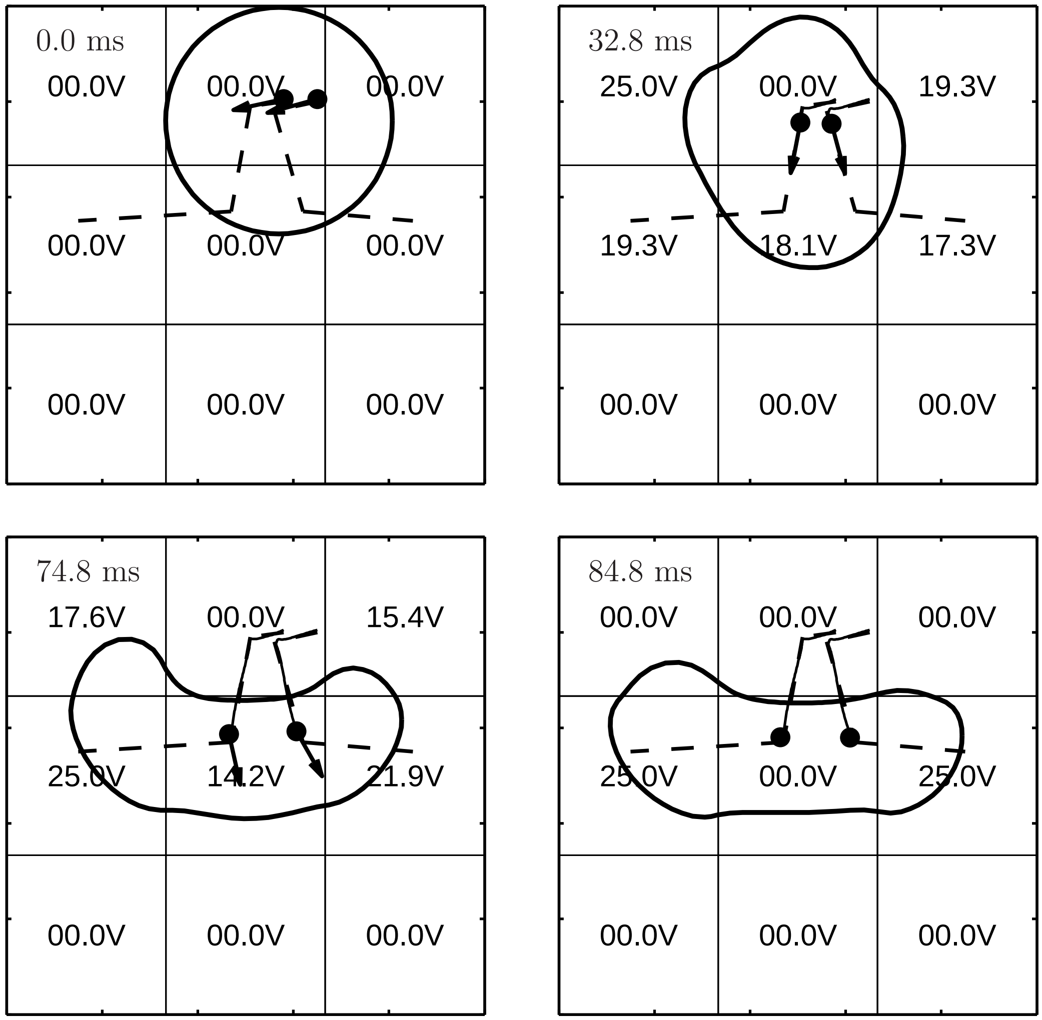


Figure 6.18: Two-particle separation into two satellite drops, part A (variational method without pinning). Same format as Figure 6.8. The paths of the particles follow the desired trajectories much better than for the level set method in Figure 6.16. Eventually, the algorithm turns-off (last frame) and open loop voltages are applied to split the droplet. Simulation continued in Figure 6.19.

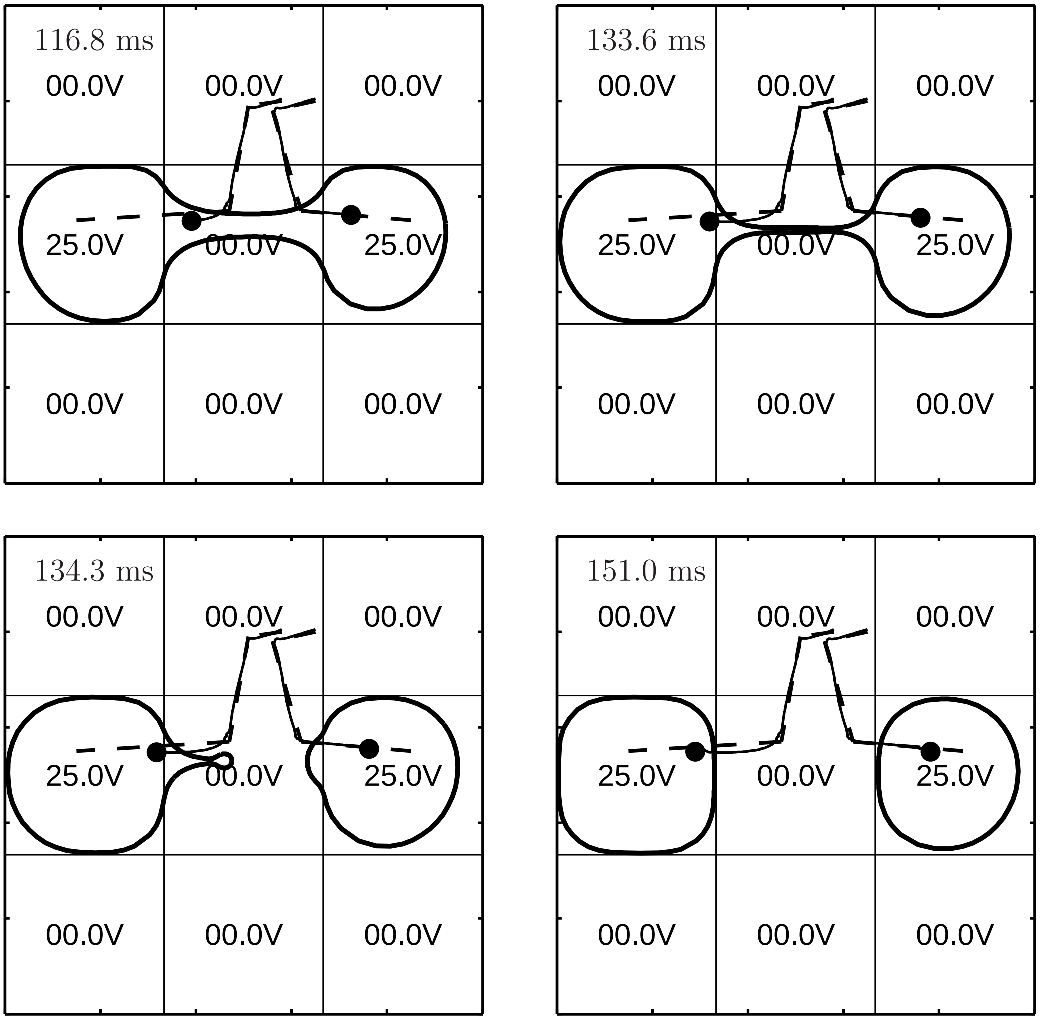


Figure 6.19: Two-particle separation into two satellite drops, part B (variational method without pinning). Same format as Figure 6.8.

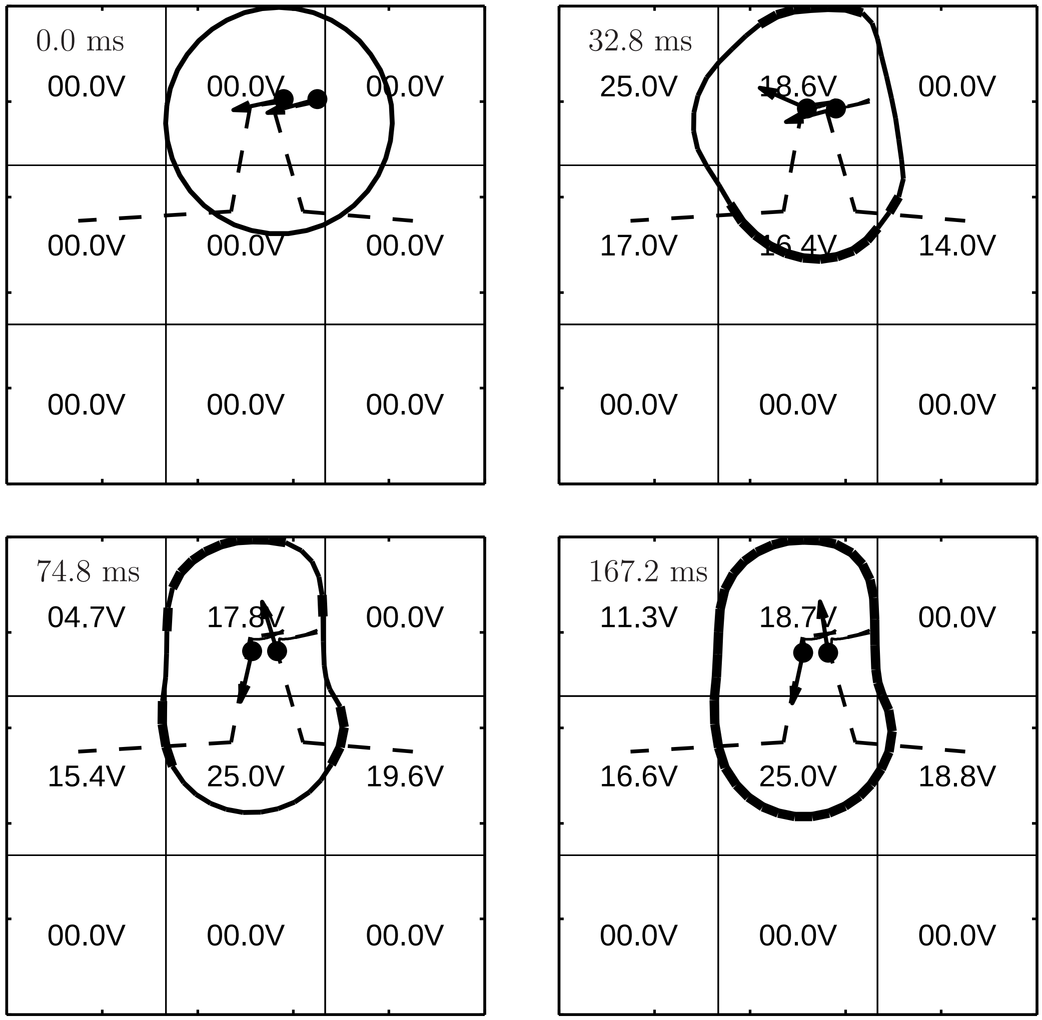


Figure 6.20: Two-particle separation into two satellite drops (variational method with pinning). Same format as Figure 6.9. The particles attempt to follow the splitting trajectories, but quickly become stuck due to contact line pinning (the droplet is fully pinned in the last frame). The algorithm is unaware of the pinned state and does not know how to compensate. See Figure 6.9 for more explanation.

and position is such that the number of available electrodes that overlap the droplet boundary is very limited. It becomes impossible to create a pressure gradient field that will continue moving the particle in the tangential direction of the desired trajectory. Hence, the particle drifts away from the trajectory by more than 100 micrometers. This situation corresponds to (6.3) having no exact solution, which means only a least squares best fit of the desired forcing direction can be computed. Eventually, however, the particle is able to reacquire the trajectory.

Figure 6.22 presents the same case using the variational simulation without the pinning model; Figure 6.23 includes the pinning model.

6.3.6 Unstable Pinching Path

Figure 6.24 shows two initially separate particles trying to come together and touch. The desired motion of the particles induces the droplet to try and pinch together in an effort to have the particles touch. However, when the particles begin to near each other, the droplet ceases its splitting action. Instead, the droplet holds the necking region and begins to oscillate up and down. This is because we are specifying two opposite directions of motion at points very close together leading to a numerical instability in solving (6.3). As the particle positions get closer together, the condition number of the matrix G degenerates causing spurious oscillations in the control voltages. The droplet is unable to bring the particles together, much less pinch, because of the randomly varying electrode voltages.

Figure 6.25 presents the same case using the variational simulation without

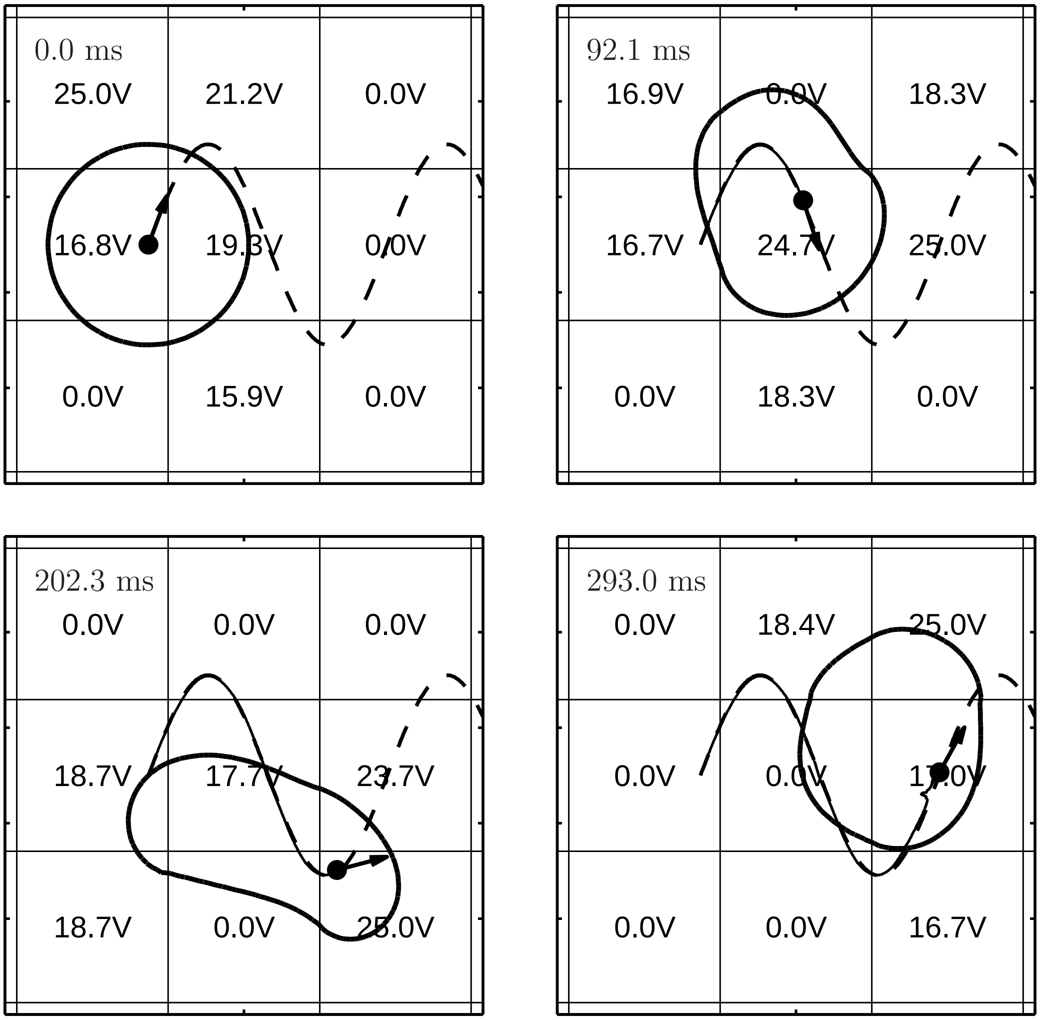


Figure 6.21: Particle traveling on a sine wave (level set method). Same format as Figure 6.7. The particle is able to track the sine wave path until the last time frame where the particle drifts away from the desired trajectory momentarily (see ‘kink’ in the particle path just underneath the particle’s position in the last frame).

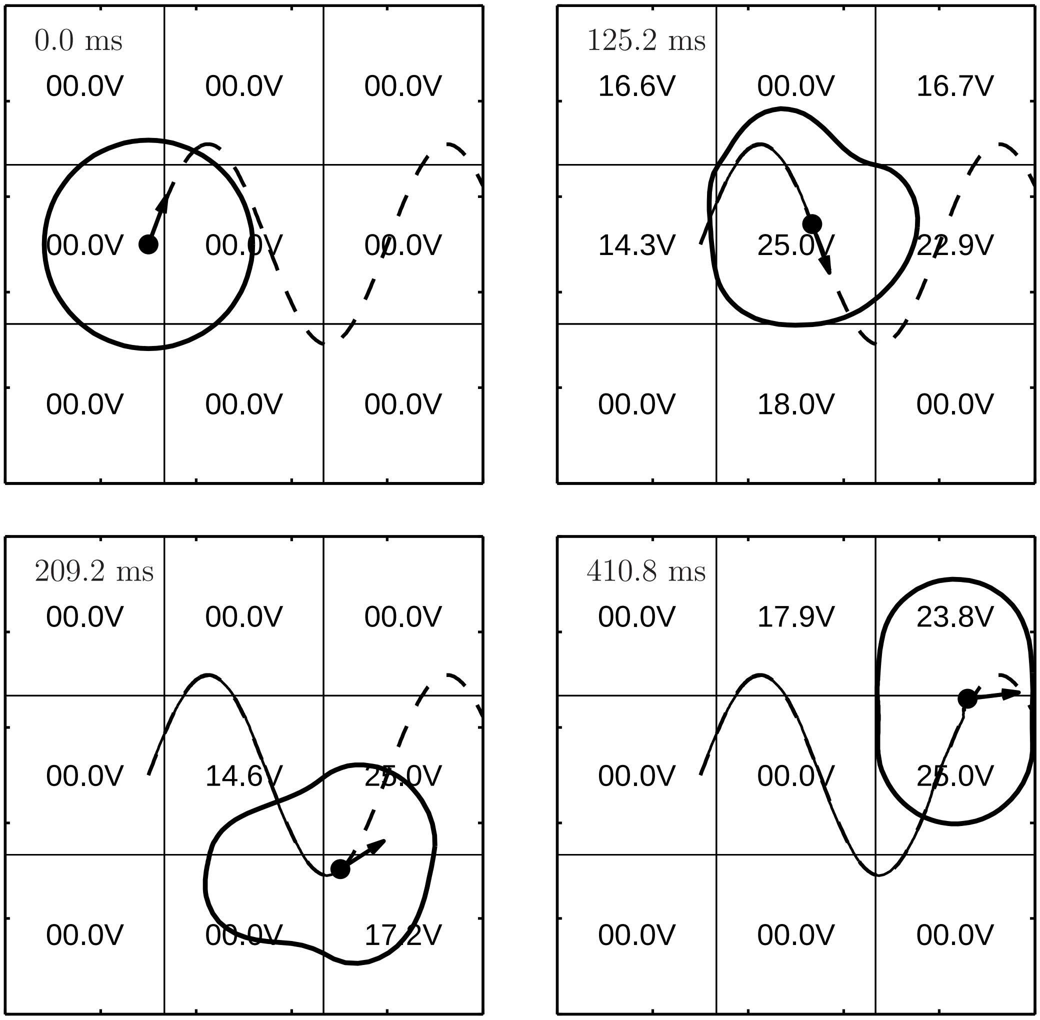


Figure 6.22: Particle traveling on a sine wave (variational method without pinning). Same format as Figure 6.8. The particle follows the sine wave very well, until near the end of the path where it is unable to continue. More electrodes would be needed to continue the motion. Also note that the particle does not drift away from the center of the droplet as much as for the level set simulation (Figure 6.21). It is possible that the excessive drift in Figure 6.21 is a numerical artifact of the level set method.

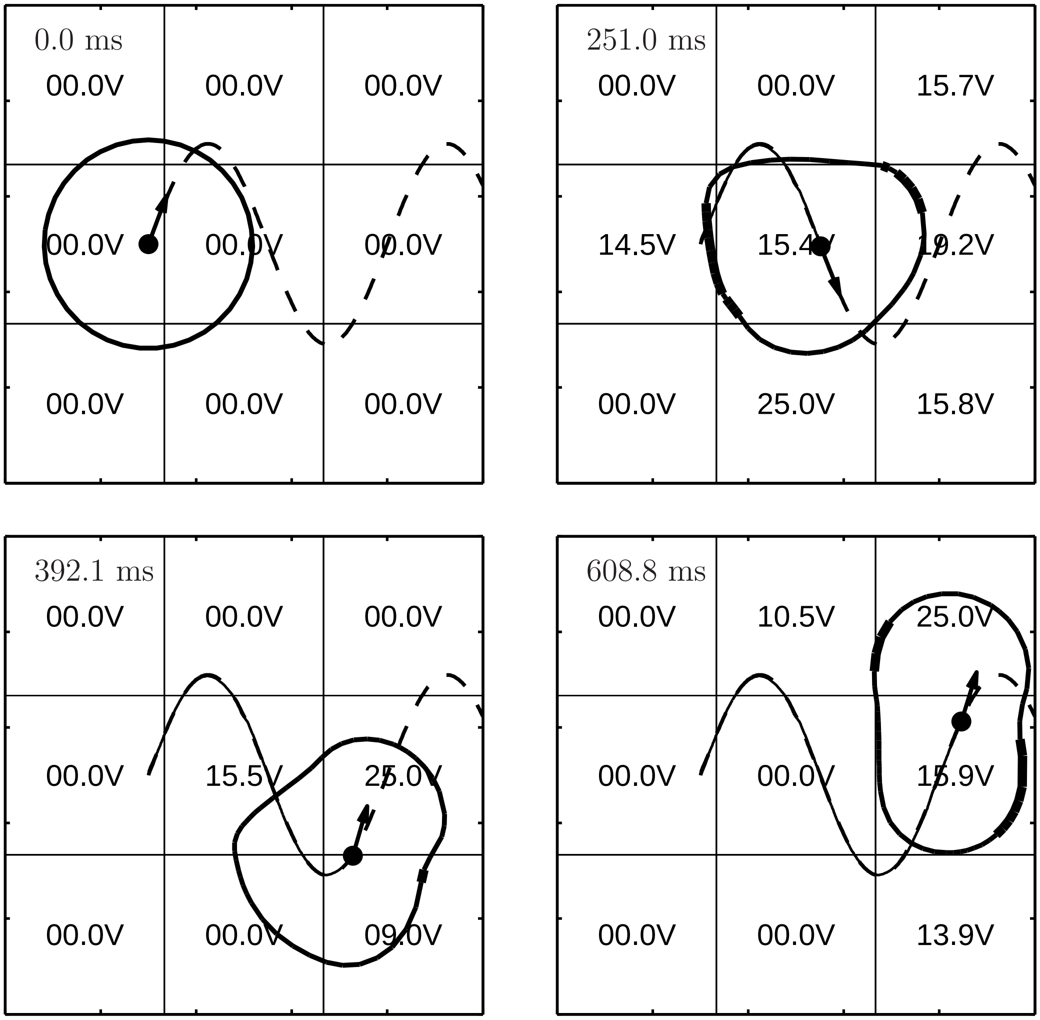


Figure 6.23: Particle traveling on a sine wave (variational method with pinning). Same format as Figure 6.9. The particle is able to follow the trajectory very well until it becomes nearly stuck due to contact line pinning (last frame).

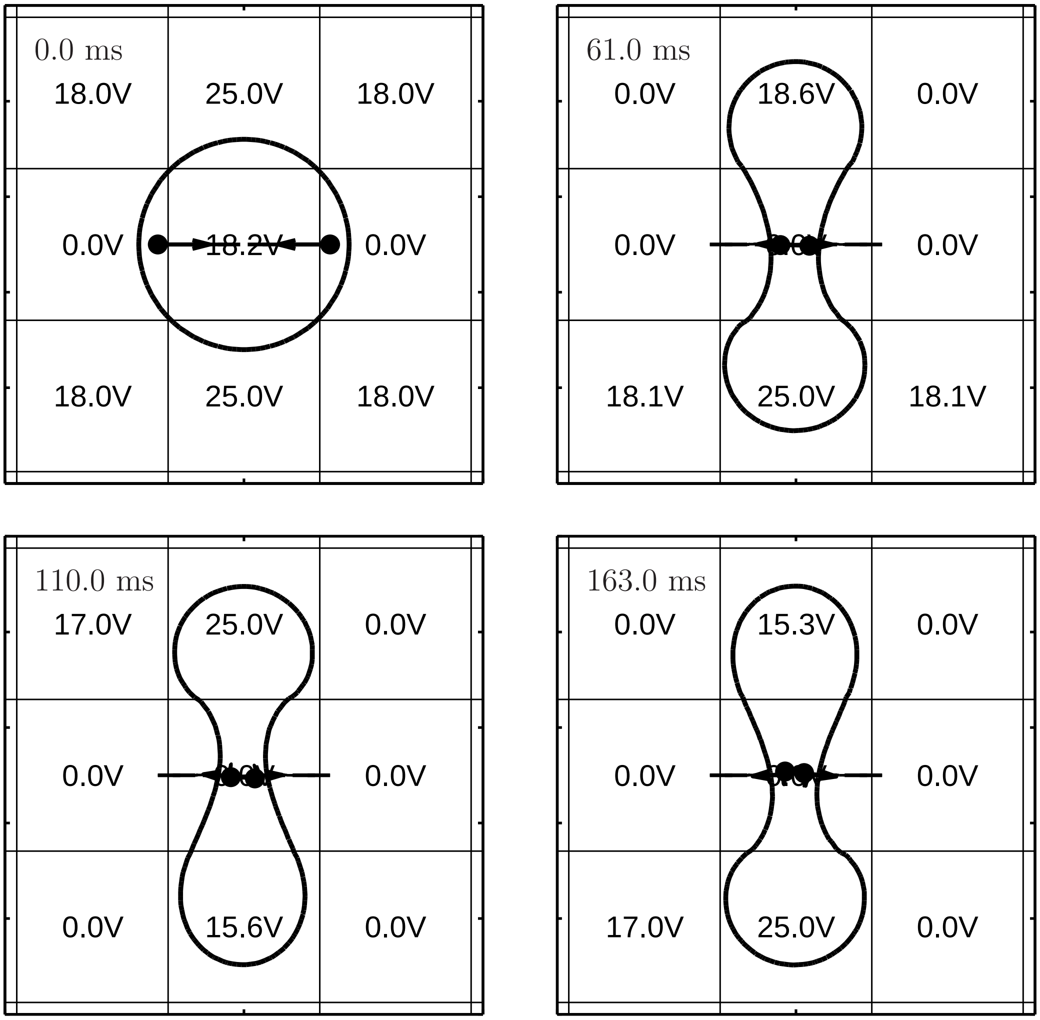


Figure 6.24: Two particles trying to come together and pinch a droplet (level set method). Same format as Figure 6.7. The particles travel on two separate trajectories that would, ideally, bring them together. However, as they come together, numerical instabilities in (6.3) cause random variations in the control voltages. This causes the droplet to hold its shape and move up and down in an undesirable way.

the pinning model; Figure 6.26 includes the pinning model.

6.3.7 Diverging Paths

Figure 6.27 shows two particles trying to follow diverging paths. At first the droplet is able to deform enough to keep the two particles on their respective trajectories. But this quickly fails. The droplet is unable to continue deforming in a way that keeps both particles on track and moving forward. Because the trajectories are just straight lines represented by two points each, the control algorithm keeps the particles moving forward while trying to force them toward the endpoints of the trajectories. The end result is both particles stay roughly parallel with each other and are unable to recover their trajectories. This stems from a lack of available electrodes and the limitations imposed by contact angle saturation.

Figure 6.28 presents the same case using the variational simulation without the pinning model; Figure 6.29 includes the pinning model.

6.4 Discussion

The limitations of achievable particle control arise from having a small number of electrodes available for actuation and from contact angle saturation. Moving several particles in different directions requires many degrees of freedom in adjusting the pressure boundary conditions. As the droplet moves, it must overlap enough electrodes to allow the realization of the pressure gradient field needed to push the individual particles along their trajectories. Hence, a finer electrode grid would allow

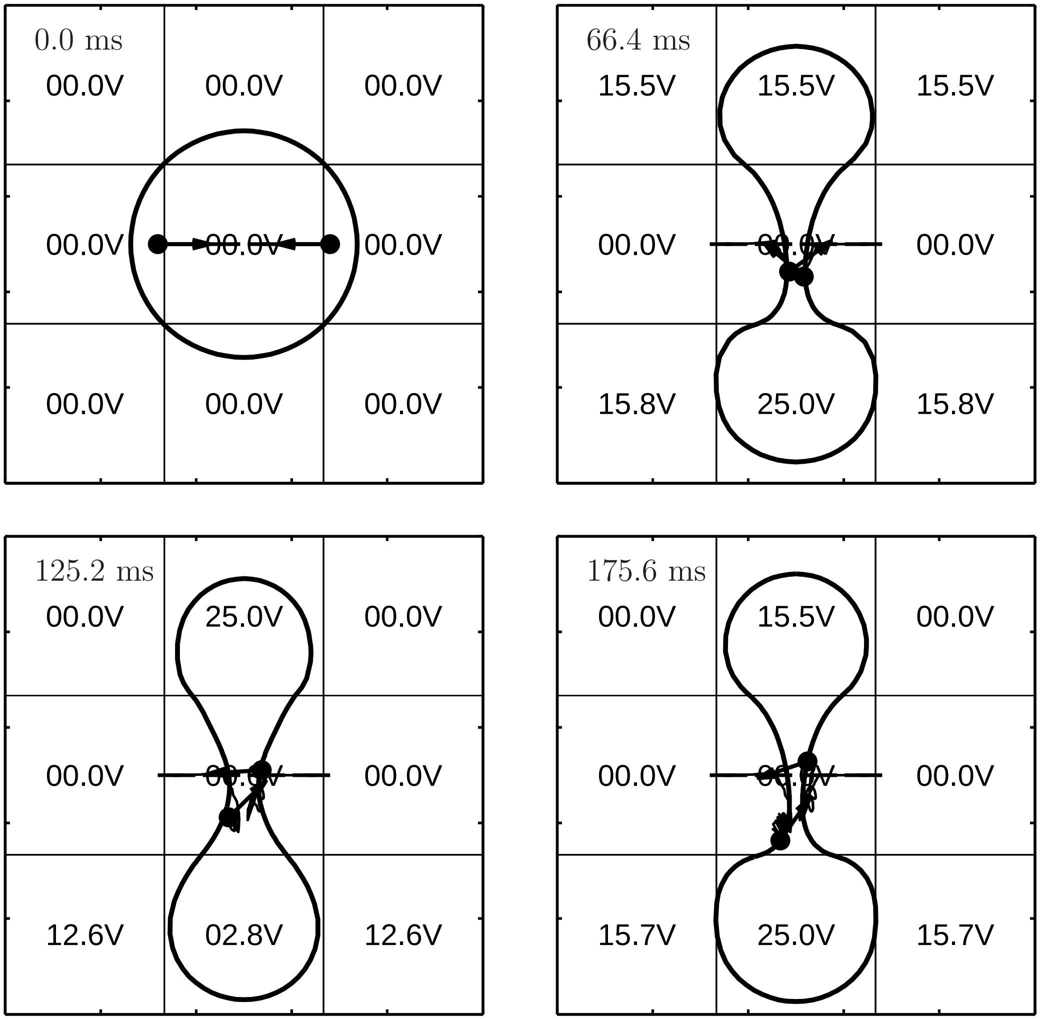


Figure 6.25: Two particles trying to come together and pinch a droplet (variational method without pinning). Same format as Figure 6.8. Results are similar to the level set method (Figure 6.24).

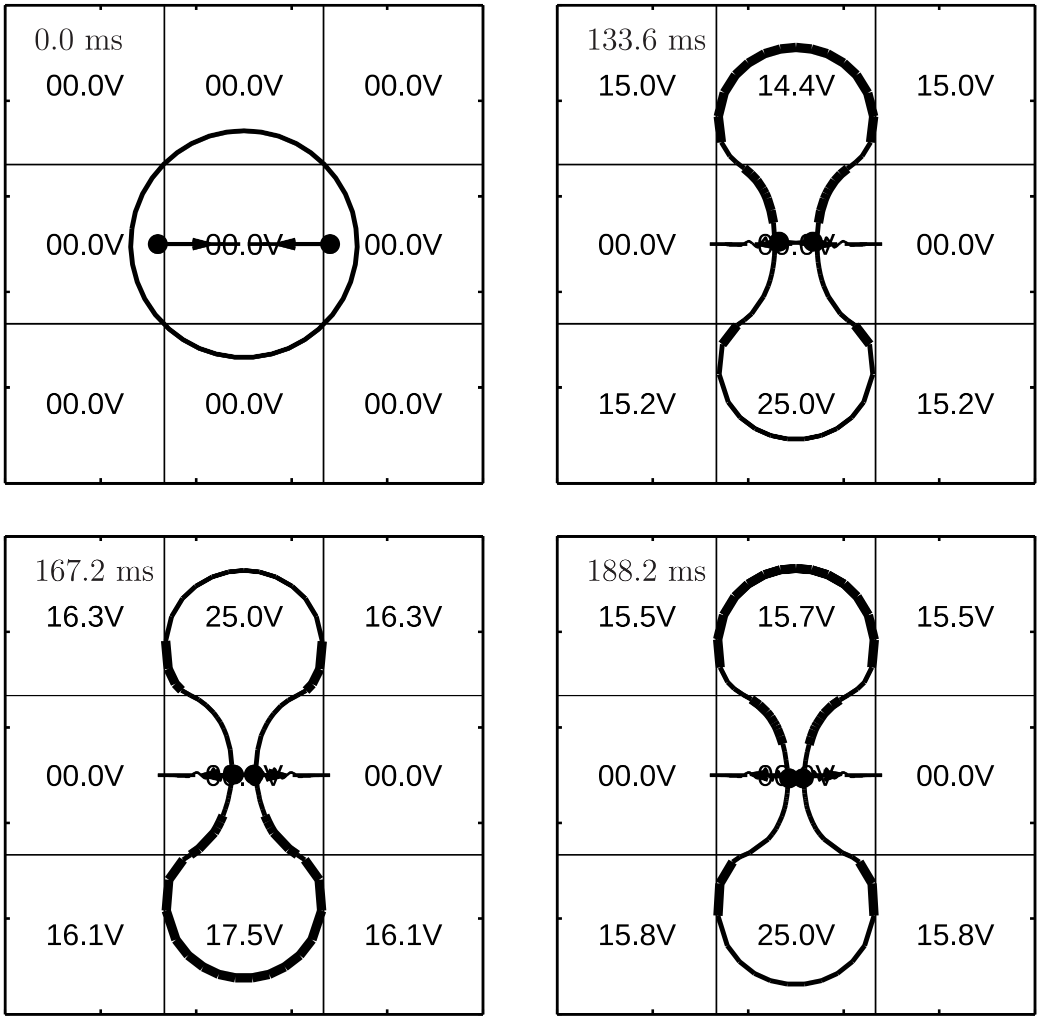


Figure 6.26: Two particles trying to come together and pinch a droplet (variational method with pinning). Same format as Figure 6.9. Results are similar to the non-pinning case (Figure 6.25).

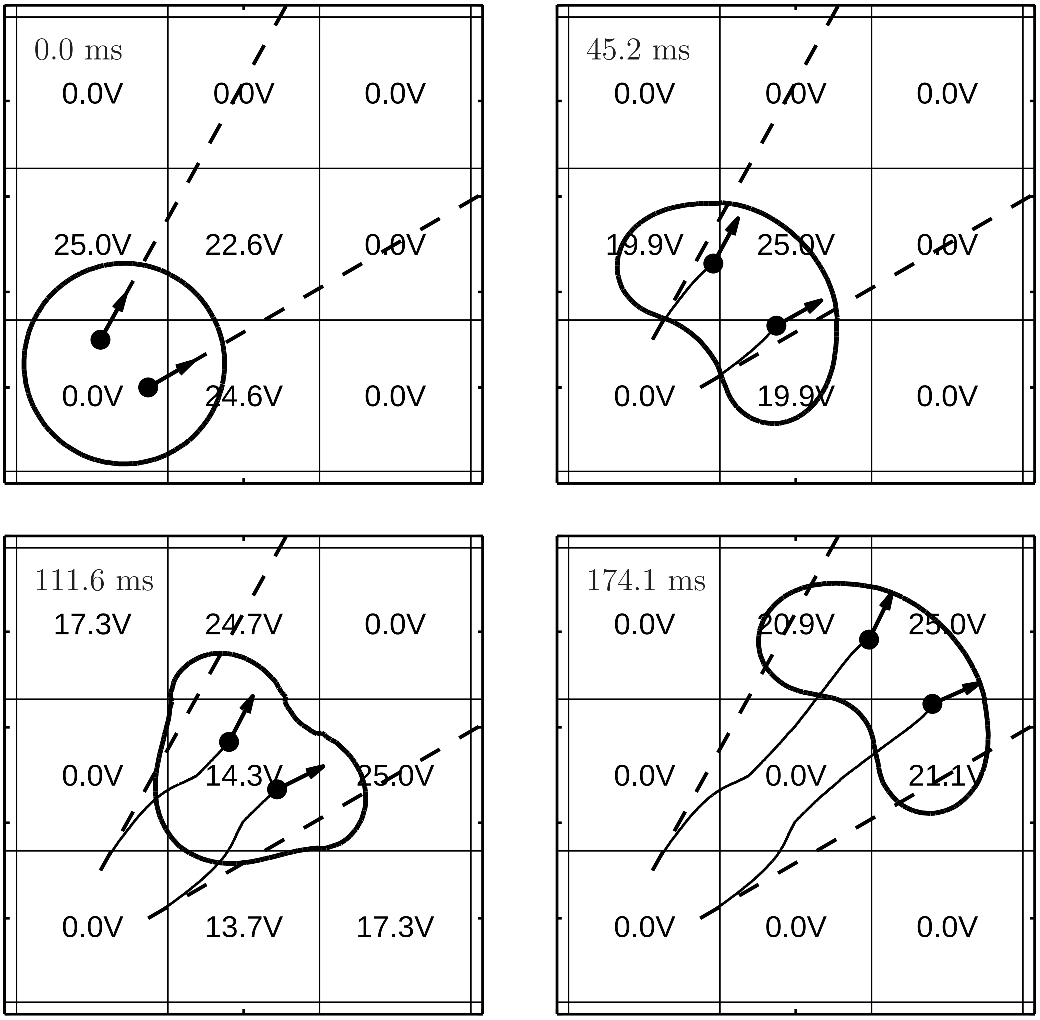


Figure 6.27: Two particles on diverging paths (level set method). Same format as Figure 6.7. Each particle is attempting to follow separate trajectories, both of which lead away from each other. Due to limitations of the pressure boundary actuation and a lack of electrodes, the control algorithm is unable to keep both particles moving on their respective paths.

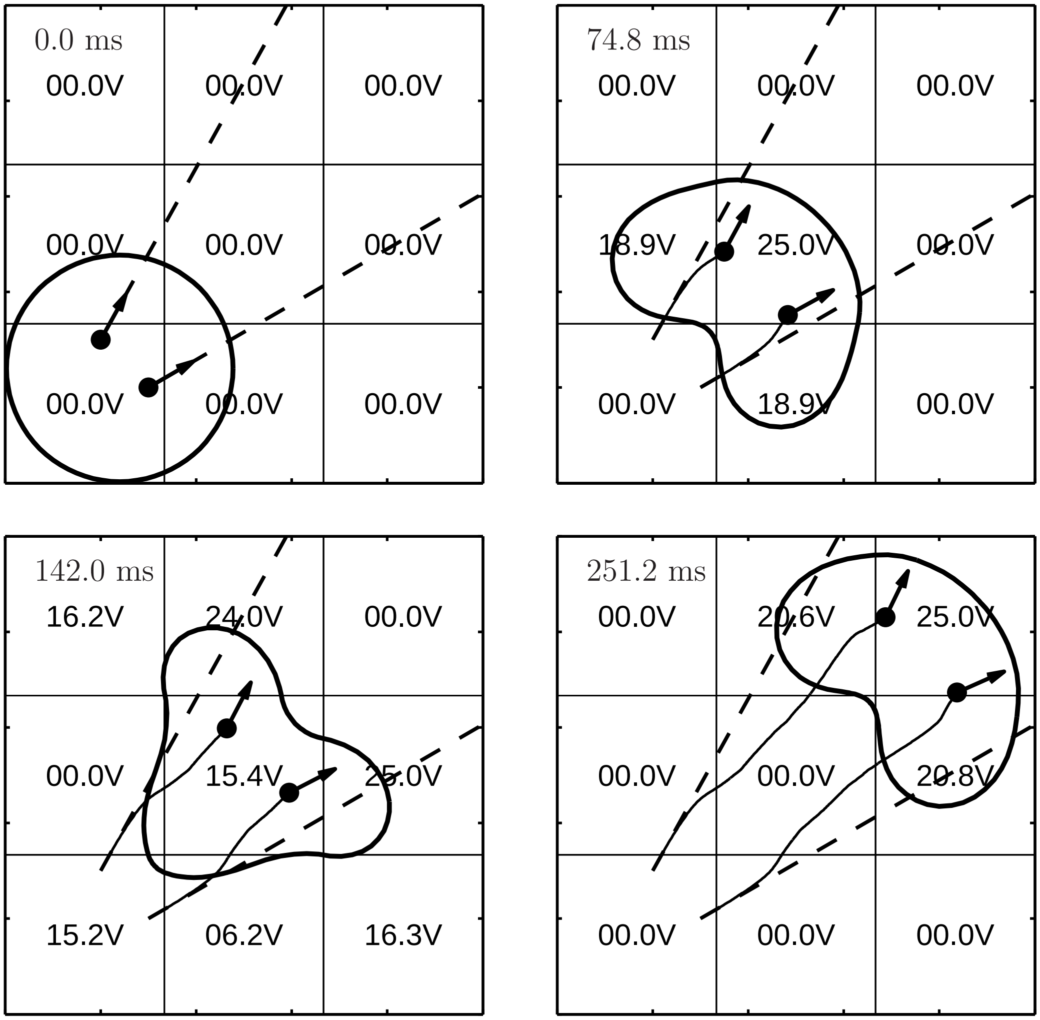


Figure 6.28: Two particles on diverging paths (variational method without pinning). Same format as Figure 6.8. Results are similar to the level set method (Figure 6.27).

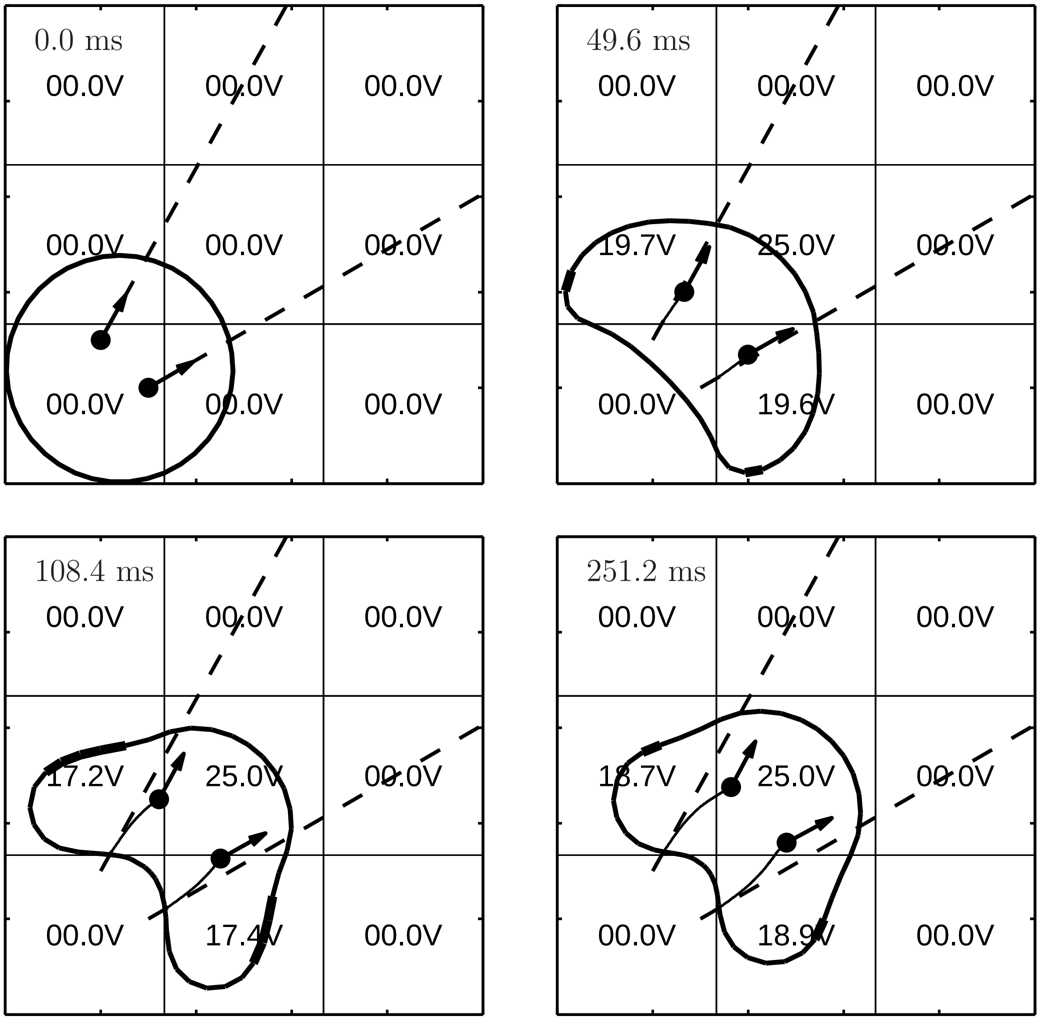


Figure 6.29: Two particles on diverging paths (variational method with pinning). Same format as Figure 6.9. The particles are able to follow their trajectories at first, but eventually becomes nearly stuck (third frame). The last frame shows the droplet asymptotically approaching a completely pinned state. Explanation is the same as that given in Figure 6.9.

more precise control of more particles simultaneously. Also, some trajectories will require the droplet to become extremely distorted and may require it to split into several pieces. To do this, one needs enough dynamic range in the boundary forcing to overcome the droplet's natural tendency to remain in a circular shape (see the planar curvature term κ in (2.16)). Contact angle saturation limits the boundary forcing and the degree of droplet deformation, which can cause controllability to be lost and particles to drift off of their desired trajectories (see Figures 6.21 and 6.27). In addition, if two particles are very close together, it is not possible to force them in arbitrary directions. The limits of boundary forcing and the numerical instability that enters into solving (6.3) inhibit close particle control no matter how many actuators are present (see Figure 6.24). Contact line pinning can also hinder particle steering for the algorithm developed here (see Figures 6.9 and 6.20). However, this could be overcome by modifying the algorithm so that following the desired trajectory could be abandoned in favor of unsticking the droplet. Another improvement that can be made is to solve the linear system (6.3) while taking into account the inequality constraint (6.4) directly (instead of scaling the solution as in Figure 6.4). This could possibly give a solution ξ that produces a higher forcing of the particle in the desired direction of motion, which would be helpful for overcoming droplet sticking.

As of today, it is only feasible to fabricate devices with a few actuators that can control one or two particles. But it is interesting that existing EWOD systems have enough control authority to steer a single particle along complex trajectories and to steer two particles along simple paths.

Chapter 7

Conclusions

We give a summary of the contributions of this work followed by some discussion.

7.1 Contributions of This Work

This section is broken into four subsections: 1) modeling of EWOD flow, 2) implementation of the implicit front-tracking level set method, 3) derivation of the explicit-front tracking variational method, and 4) performing particle control in the EWOD system. Each subsection lists and describes its main contributions. Note: some of the contributions here have been published in [142] and [141].

7.1.1 Modeling EWOD Flow

1. Non-dimensionalized the problem and performed an order of magnitude analysis to derive the governing equations.

2. Demonstrated, computationally, the need to include contact line friction effects to better match experimental data for EWOD-driven droplet motion.

3. Introduced a constant scaling factor (hysteresis constant) that approximately models the frictional effects induced by contact angle hysteresis. This was

necessary in order to adjust for the mismatch in time scales when comparing simulations to experimental splitting times because our prior model had ignored 3-phase contact line friction. This introduced a single parameter into the simulation that had to be fit to the experimental data. But this scaling factor is similar to other contact line friction models from the literature.

4. Compared simulations to experiments at UCLA. This is clearly important to demonstrate the usefulness of our computational tool to the electrowetting community.

5. Developed a phenomenological model of contact line pinning to account for sticking of the three-phase contact line of a droplet, which is a noticeable effect in the UCLA EWOD system.

7.1.2 Implementing the Level Set Method

1. Applied the level set method to simulating droplets moving in an EWOD system.

2. Used a method-of-lines approach to evolve the velocity field at each time step.

3. Introduced a diffusion method for extending the velocity field when updating the level set function. This is a necessary step when using the level set method, and our diffusion method is not the usual technique for extending the velocity field.

4. Modified the level set method so that it conserves mass while still being com-

putationally efficient and able to evolve through topological changes.

5. Implemented the method in MATLAB with some routines written in C for computational efficiency.

7.1.3 Deriving the Variational Method

1. Recast the EWOD fluid equations into a mixed variational formulation and applied FEM. This was done to overcome the following problems in our level set method: lack of accurate (local) mass conservation and noisy curvature due to our previous explicit calculation that involved differentiating numerical data.

2. Included our phenomenological line-pinning model into the variational formulation using a variational inequality. This was another instance where the variational method proved to be superior to the level set implementation.

3. Mathematically proved the well-posedness of the time-discrete version of the EWOD governing equations (with contact line pinning), as well as for the fully discrete version.

4. Developed a stable and accurate algorithm for the solution of the FEM linear system that includes the contact line pinning inequality constraint.

5. Implemented the method in MATLAB with some routines written in C for computational efficiency.

7.1.4 Particle Control in the EWOD System

1. Developed an algorithm for actuating the voltages in an EWOD device for controlling the trajectory of a floating particle just by using EWOD actuation. This demonstrated to the electrowetting community that its systems could be used to steer particles, and was new and unexpected.

2. Developed the algorithm capable of dealing with the non-linear aspect of the evolution of the droplet shape. As a droplet moves across the electrode grid, the forcing available for particle control changes in a way that cannot be accounted for a priori. This requires the algorithm to compute the available forcing at each time step of the evolution in order to control the particle motion.

3. Accounted for constraints of voltage actuation to ensure maximal speed of particle motion. The forcing available for particle control is limited by contact angle saturation in the EWOD system. Therefore, we included a step in the control algorithm that ensures the controlling voltages satisfy the constraints while maximizing the particle speed along the given trajectory.

7.2 About the Model and Numerics

This thesis has presented a model and numerical simulation of droplet motion inside an electrowetting device. Starting from the full Navier-Stokes equations we obtained a reduced order model, similar to Hele-Shaw type flow, that captures the bulk dynamic behavior of EWOD-driven droplets in 2-D. The key part of our analysis is including contact angle saturation, a simple model of hysteresis, and a phenomenological model of contact line pinning in order to match the experimental data. Our simulation results show how the liquid-gas interface physics affect the motion of the droplet. When all effects are included, our simulations compare favorably with the experiments but are not an exact match. The main reasons for this are:

1. contact line pinning and contact angle hysteresis are not modeled exactly;
2. inadequate modeling of the detailed fluid dynamics near the liquid-gas interface;
3. uncertainty in the experimental conditions.

Some improvements on item (1) can be made by considering a more realistic contact line friction model. For example, the pressure boundary condition could be modified to include a 'viscous' damping term:

$$p = \kappa + E + P_{pin}\mathrm{sgn}(\vec{u} \cdot \vec{n}) + D_{viscous}(\vec{u} \cdot \vec{n}),$$

where the extra term creates a boundary pressure that pushes back with increasing normal velocity ($D_{viscous}$ is a damping coefficient). This could replace the hysteresis constant K_{hys} and be more 'physical', however $D_{viscous}$ would still be a fitting parameter for the simulation. Including this extra term into the variational framework is quite easy. One only needs to include another integral term into the $a(\cdot, \cdot)$ bilinear form, namely:

$$D_{viscous} \int_{\Gamma} (\vec{u} \cdot \vec{n})(\vec{v} \cdot \vec{n}).$$

None of the function spaces would change and the well-posedness of the formulation would be the same. In fact, adding this extra term would improve the variational inequality solver because it adds another positive term. This is also another instance where the variational/front tracking method is more flexible than the level set method; it is not clear how to include a 'boundary viscous term' with level sets.

As for item (2), the current model does not accurately capture boundary effects because they are ignored in the derivation of Hele-Shaw flow. One possible remedy would be to vary the parabolic velocity profile assumption when near the liquid-gas interface (e.g. make the velocity profile along the channel spacing be more uniform rather than parabolic). However, this would complicate the resulting governing equations. Moreover, it may be necessary to account for the 3-D fluid dynamics near the interface in order to markedly improve the model (see [70] for an instance where this was important).

Improving item (3) would require more controlled experiments and is beyond the scope of this thesis.

The numerical methods introduced here are fast, and readily lend themselves to control algorithm design and device optimization. Computing time of the level set simulations (in MATLAB) varied between 3 and 6 minutes. Our variational method (also in MATLAB) was comparable, with some simulations requiring up to 10 minutes to complete when simulating contact line pinning. We note here that the

Uzawa iterative variational inequality solver can be slow to converge. This is mainly due to the fact that our Uzawa method is essentially a steepest descent method, as opposed to a conjugate gradient method which would be much faster. Deriving a fast solver for our contact line pinning model is still an active area of research.

As for the performance of the two methods, the main advantage of the level set method is that it has a simple grid structure and can go through topological changes fairly automatically. But it is our belief that the variational method is superior with respect to accurately computing boundary conditions, namely:

- being able to represent the boundary accurately;
- computing curvature accurately;
- simulating contact line pinning.

In addition, our variational method conserves mass better than our level set implementation and without including an ad hoc correction. The only drawback is in computing through topological changes. But the method in [140] is able to account for this. Therefore, our variational method is a good tool for simulating the fluid motion of EWOD driven droplets.

7.3 On Particle Control

A control method has been developed to steer individual particles inside electrowetting systems. Using only existing EWOD electrode actuators, the algorithm can steer particles to specific locations, hold one particle stationary while another particle is moved along a path, and steer and sort particles within and across individual droplets. The particle steering is achieved by creating fluid flows that carry all the particles from where they are to where they should be at each time step. It is possible to steer a single particle along interesting trajectories with a small (3x3) number of electrodes. Steering two particles independently inside a single drop is also possible for simple particle motions. A finer grid of electrodes, or the use of larger liquid drops, would allow simultaneous steering of more particles along more complicated trajectories inside a single drop of liquid. The particle steering results outlined in Chapter 6 are based on the EWOD model described in Chapter 2 and the experimentally validated simulation methods detailed in Chapters 3 and 4. An experimental demonstration of particle steering in the UCLA devices will require the implementation of a real time vision system, a real time control algorithm realization, image sensing, control computation, and EWOD device system integration. These items have been demonstrated for an electro-osmotically driven micro-fluidic system (Shapiro, et al [4], [5]), so it is reasonable that the same could be done for the EWOD system.

Appendix A

Some Differential Geometry

This appendix states the basic definitions of differential geometry and derives some formulas used in the main text. For more details on this subject, see the references in [37], [7], [145], [38], [116]. Note: some of the notation in this appendix supersedes the notation in the rest of the thesis.

A.1 Main Definitions

Differential geometry is the study of the *intrinsic* geometry or shape of a surface. The surface can be a 1-D curve in a 2-D or 3-D space ($\mathbb{R}^2$ or $\mathbb{R}^3$) or a 2-D surface in 3-D space (see Figures A.1 and A.2). For example, a 1-D surface could represent the liquid-gas interface of a 2-D droplet in an EWOD device. All differential geometric relations are true regardless of the way the surface is parameterized (i.e. the way the surface is mapped or labeled).

For our purposes, we will use an explicit parameterization of the surface in deriving the differential geometric relations. And for the sake of generality, we will compute most of the relations for a 2-D surface (embedded in a 3-D space), but all formulas derived here are true for a 1-D surface (curve) also. This is because a 1-D curve can be interpreted as a cross-section of a cylindrical 2-D surface (see Figure A.3).

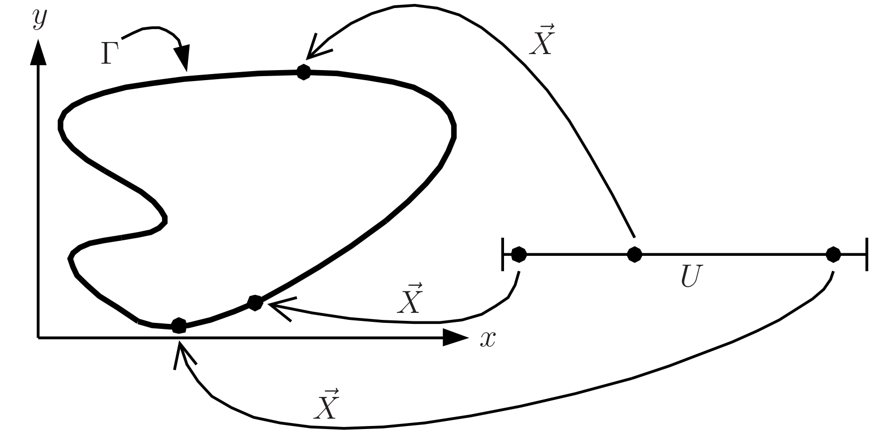


Figure A.1: A 1-D closed curve Γ with mapping $\vec{X}$. The mapping is defined on a single reference domain U, which is just an interval. Only one reference domain is needed for 1-D curves.

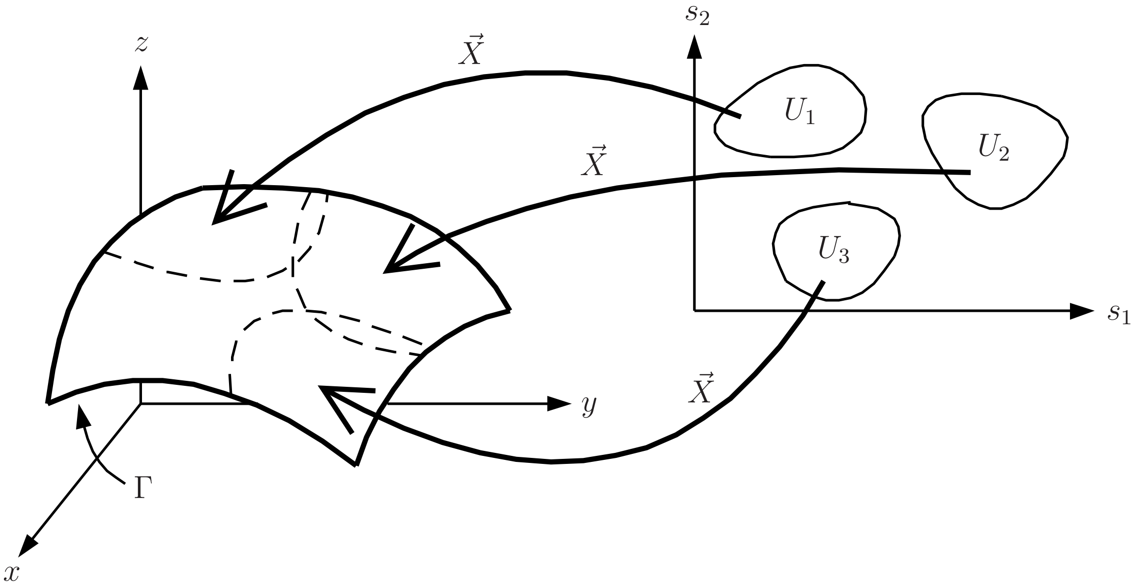


Figure A.2: Section of a 2-D surface Γ with mapping $\vec{X}$. The mapping is defined on multiple open sets $\{U_1, U_2, U_3\}$ (reference domains) that are disjoint. Each U_i is mapped to a small patch (denoted by a dashed curve) on the surface Γ. More than one reference domain is required to cover the whole surface Γ if it is closed.

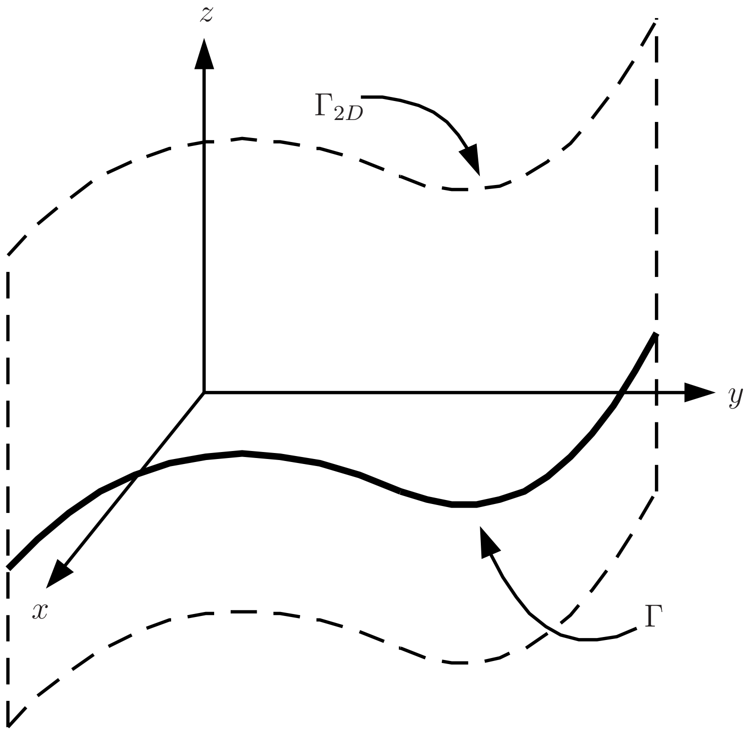


Figure A.3: A 1-D curve Γ (in the x-y plane) shown as a cross-section of a 2-D surface Γ_{2D}. All differential geometric formulas derived in this appendix are for a 2-D surface in an ambient 3-D space. But these formulas also hold for a 1-D curve in a 2-D ambient space. This can be seen by noting that a 1-D curve can be interpreted as a cross-section of a cylindrical surface (shown here).

A.1.1 Parameterization

Let Γ denote the set of points (in $\mathbb{R}^3$) that defines the surface, and let $\vec{X} : U_i \rightarrow \Gamma \subset \mathbb{R}^3$ be a mapping that parameterizes a 'patch' of Γ (i.e. $\vec{X}(U_i) \subset \Gamma$) for each i in some finite index set (see Figure A.2). Each disjoint open set U_i is a reference domain in $\mathbb{R}^2$ with local 2-D variables s_1 and s_2 (i.e. $\vec{X}(U_i) = \{\vec{X}(s_1, s_2) : (s_1, s_2) \in U_i\}$). Furthermore, let $\vec{X}$ satisfy $\bigcup_i \vec{X}(U_i) = \vec{X}(\cup_i U_i) = \Gamma$. Hence, $\vec{X}(\cdot) = \Gamma$ is a total surface parameterization using local charts $\{U_i\}$ [37]. Note that $\vec{X}$ is a vector function with coordinate functions denoted by $\vec{X} = (X_1, X_2, X_3)$.

We also define the unit normal vector $\vec{n}$ (in terms of local coordinates) using the surface tangent vectors $\partial_{s_1}\vec{X}$ and $\partial_{s_2}\vec{X}$

$$\vec{n} = \frac{(\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{X})}{|\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{X}|}, \tag{A.1}$$

where the parameterization is chosen such that $\vec{n}$ is an <u>outward</u> pointing normal vector when Γ is a closed surface. Note that $|\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{X}| = \sqrt{\det(g)}$ (see (A.2) below). In the following, we assume that $\vec{X}$ is a *smooth* function (i.e. at least C^2).

A.1.2 Fundamental Forms

The fundamental forms from differential geometry [37] are defined in the following way. The first fundamental form of differential geometry is given by a metric which, for a 2-D surface, is a 2x2 matrix denoted g:

$$g = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}, \tag{A.2}$$

where the coefficients g_{ij} are given by certain derivatives of the parameterization

$$g_{ij} = \partial_{s_i}\vec{X} \cdot \partial_{s_j}\vec{X}, \text{ for } 1 \leq i, j \leq 2. \tag{A.3}$$

Note that g is symmetric. The inverse of the matrix g is given by

$$g^{-1} = \begin{bmatrix} g^{11} & g^{12} \\ g^{21} & g^{22} \end{bmatrix} = \frac{1}{\det(g)} \begin{bmatrix} g_{22} & -g_{12} \\ -g_{21} & g_{11} \end{bmatrix}, \tag{A.4}$$

where we denote the coefficients of the inverse with a superscript for the indices. Of course, we have the following property because $g\, g^{-1} = I$:

$$\delta_i^j = \sum_{k=1}^{2} g_{ik} g^{kj} = \sum_{k=1}^{2} g_{ik} g^{jk}, \tag{A.5}$$

$$\begin{aligned} \delta_i^j &= 1, i = j, \\ \delta_i^j &= 0, i \neq j. \end{aligned} \tag{A.6}$$

The second fundamental form h is also a 2x2 matrix:

$$h = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}, \tag{A.7}$$

where the coefficients h_{ij} are given by [37]

$$h_{ij} = -\partial_{s_i}\vec{n} \cdot \partial_{s_j}\vec{X} = \vec{n} \cdot \partial_{s_i}\partial_{s_j}\vec{X}, \text{ for } 1 \leq i, j \leq 2. \tag{A.8}$$

Using (A.2) and (A.7), the total and Gaussian curvatures (both scalar) are given by (respectively):

$$\kappa = \kappa_1 + \kappa_2 = -\sum_{i,j=1}^{2} g^{ij} h_{ij}, \tag{A.9}$$

$$\kappa_G = \kappa_1 \kappa_2 = \frac{\det(h)}{\det(g)}, \tag{A.10}$$

where κ_1 and κ_2 are the *principal curvatures* of the surface Γ. There are two curvatures because Γ is a 2-D surface (note: for a 1-D curve there is only one curvature, denoted κ). The minus sign in (A.9) ensures that the total curvature is positive for a closed *elliptic* surface [37] when the normal vector $\vec{n}$ is defined to point outwards.

The vector curvature (or total curvature vector) is simply the product of the scalar total curvature with the normal vector: $\kappa\vec{n}$.

A.2 Surface Derivative Operators

We define operators on the surface Γ that are directly analogous to the gradient, divergence, and Laplace operators in standard calculus.

A.2.1 Surface Gradient Operator

Let $\omega : \Gamma \to \mathbb{R}$ be a scalar function defined on the surface Γ. Then the surface gradient $\nabla_\Gamma(\cdot)$ of ω in local coordinates is defined by

$$[\nabla_\Gamma \omega] \circ \vec{X} = \sum_{i,j=1}^{2} g^{ij} \partial_{s_i} \tilde{\omega} \partial_{s_j} \vec{X}, \tag{A.11}$$

where $\tilde{\omega} = \omega \circ \vec{X}$ is in local coordinates. Note that $[\nabla_\Gamma \omega] \circ \vec{X}$ is a 1x3 row vector.

Let $\vec{\varphi} : \Gamma \to \mathbb{R}^3$ be a vector function on Γ, and let φ_k denote the coordinate functions of $\vec{\varphi}$ (i.e. $\vec{\varphi} = (\varphi_1, \varphi_2, \varphi_3)$). Let $\tilde{\vec{\varphi}} = \vec{\varphi} \circ \vec{X}$ and $\tilde{\varphi}_k = \varphi_k \circ \vec{X}$ denote the functions in local coordinates. Then we define the surface gradient of a vector function (in local coordinates) by

$$[\nabla_\Gamma \vec{\varphi}] \circ \vec{X} = \begin{bmatrix} [\nabla_\Gamma \varphi_1] \circ \vec{X} \\ [\nabla_\Gamma \varphi_2] \circ \vec{X} \\ [\nabla_\Gamma \varphi_3] \circ \vec{X} \end{bmatrix}, \tag{A.12}$$

where

$$[\nabla_\Gamma \varphi_k] \circ \vec{X} = \sum_{i,j=1}^{2} g^{ij} \partial_{s_i} \tilde{\varphi}_k \partial_{s_j} \vec{X}. \tag{A.13}$$

Note that $[\nabla_\Gamma \vec{\varphi}] \circ \vec{X}$ is a 3x3 matrix. We will also have need of the following quantity as well

$$[\nabla_\Gamma (\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} = \sum_{i,j=1}^{2} g^{ij} \partial_{s_i} \vec{X} \otimes \partial_{s_j} \vec{X}, \tag{A.14}$$

where $\vec{X} \circ \vec{X}^{-1}$ is the identity map on Γ. But we prefer to keep it as written.

A.2.2 Surface Divergence Operator

The usual divergence operator is defined as $\nabla \cdot \vec{v} = \mathrm{trace}(\nabla \vec{v})$. We define the surface divergence operator in a similar way (again in local coordinates)

$$[\nabla_\Gamma \cdot \vec{\varphi}] \circ \vec{X} = \sum_{i,j=1}^{2} g^{ij} \partial_{s_i} \tilde{\vec{\varphi}} \cdot \partial_{s_j} \vec{X} = \sum_{i,j=1}^{2} g^{ij} \sum_{k=1}^{3} \partial_{s_i} \tilde{\varphi}_k \partial_{s_j} X_k, \tag{A.15}$$

where $[\nabla_\Gamma \cdot \vec{\varphi}] \circ \vec{X}$ is a scalar function.

Surface Divergence of the Identity Map

From standard multi-variable calculus, we know that $\nabla \cdot \vec{x}$ is equal to the dimension of the space. We show a similar result for the surface divergence

Proposition A.2.1. *If* Γ *is a 2-D surface, then*

$$\nabla_\Gamma \cdot (\vec{X} \circ \vec{X}^{-1}) = 2. \tag{A.16}$$

And for a 1-D surface, we have

$$\nabla_\Gamma \cdot (\vec{X} \circ \vec{X}^{-1}) = 1. \tag{A.17}$$

Proof. Applying (A.15) to the identity map $(\vec{X} \circ \vec{X}^{-1})$ gives

$$[\nabla_\Gamma \cdot (\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} = \sum_{i,j=1}^{2} g^{ij} \sum_{k=1}^{3} \partial_{s_i} X_k \partial_{s_j} X_k = \sum_{i,j=1}^{2} g^{ij} \partial_{s_i} \vec{X} \cdot \partial_{s_j} \vec{X}.$$

And using the definition of the first fundamental form (A.3) and properties (A.5) and (A.6), reduces this to

$$[\nabla_\Gamma \cdot (\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} = \sum_{i,j=1}^{2} g^{ij} g_{ij} = \sum_{i=1}^{2} \delta_i^i = \sum_{i=1}^{2} 1 = 2,$$

giving the assertion for a 2-D surface. Formula (A.17) comes from the fact that the metric for a 1-D surface is a 1x1 matrix $g = \partial_s \vec{X}(s) \cdot \partial_s \vec{X}(s)$ (where s is the only parameterization variable) and the surface divergence is just

$$[\nabla_\Gamma \cdot (\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} = \frac{\partial_s \vec{X}(s)}{g} \cdot \partial_s \vec{X}(s) = 1.$$

□

A.2.3 Surface Laplacian Operator

Before defining the surface Laplacian operator, we compute some useful formulas.

Intermediate Formulas

Proposition A.2.2.

$$\frac{1}{\sqrt{\det(g)}}\partial_{s_q}\sqrt{\det(g)} = \frac{1}{2}\sum_{i,j=1}^{2} g^{ij}\partial_{s_q}(g_{ij}).$$

Proof. Recall that $\det(g) = g_{11}g_{22} - g_{12}g_{21}$, so we have

$$\partial_{s_q}\sqrt{\det(g)} = \frac{1}{2\sqrt{\det(g)}}(g_{22}\partial_{s_q}g_{11} + g_{11}\partial_{s_q}g_{22} - g_{21}\partial_{s_q}g_{12} - g_{12}\partial_{s_q}g_{21}).$$

After a slight manipulation,

$$\begin{aligned}\partial_{s_q}\sqrt{\det(g)} = \frac{\sqrt{\det(g)}}{2}\Big(&\frac{g_{22}}{\det(g)}\partial_{s_q}g_{11} + \frac{g_{11}}{\det(g)}\partial_{s_q}g_{22} + \\ &-\frac{g_{21}}{\det(g)}\partial_{s_q}g_{12} - \frac{g_{12}}{\det(g)}\partial_{s_q}g_{21}\Big).\end{aligned}$$

After using (A.4), and the fact that g is symmetric, we get

$$\begin{aligned}\partial_{s_q}\sqrt{\det(g)} &= \frac{\sqrt{\det(g)}}{2}(g^{11}\partial_{s_q}g_{11} + g^{22}\partial_{s_q}g_{22} + g^{12}\partial_{s_q}g_{12} + g^{21}\partial_{s_q}g_{21}), \\ &= \frac{\sqrt{\det(g)}}{2}\sum_{i,j=1}^{2} g^{ij}\partial_{s_q}(g_{ij})\end{aligned}$$

which implies the assertion. □

Proposition A.2.3.

$$\sum_{i,j=1}^{2} g^{ij}(\partial_{s_q}\partial_{s_i}\vec{X}) \cdot (\partial_{s_j}\vec{X}) = \frac{1}{\sqrt{\det(g)}}\partial_{s_q}\sqrt{\det(g)}.$$

Proof. Start by writing out the sum

$$\begin{aligned}\sum_{i,j=1}^{2} g^{ij}(\partial_{s_q}\partial_{s_i}\vec{X}) \cdot (\partial_{s_j}\vec{X}) = g^{11}(\partial_{s_q}\partial_{s_1}\vec{X}) \cdot (\partial_{s_1}\vec{X}) + g^{12}(\partial_{s_q}\partial_{s_1}\vec{X}) \cdot (\partial_{s_2}\vec{X}) \\ +g^{21}(\partial_{s_q}\partial_{s_2}\vec{X}) \cdot (\partial_{s_1}\vec{X}) + g^{22}(\partial_{s_q}\partial_{s_2}\vec{X}) \cdot (\partial_{s_2}\vec{X}).\end{aligned}$$

Using the standard product rule, and the fact that $g^{12} = g^{21}$, we get

$$\begin{aligned}\sum_{i,j=1}^{2} g^{ij}(\partial_{s_q}\partial_{s_i}\vec{X}) \cdot (\partial_{s_j}\vec{X}) =& \frac{g^{11}}{2}\partial_{s_q}(\partial_{s_1}\vec{X} \cdot \partial_{s_1}\vec{X}) + \frac{g^{12}}{2}\partial_{s_q}(\partial_{s_1}\vec{X} \cdot \partial_{s_2}\vec{X}) \\ &+\frac{g^{21}}{2}\partial_{s_q}(\partial_{s_2}\vec{X} \cdot \partial_{s_1}\vec{X}) + \frac{g^{22}}{2}\partial_{s_q}(\partial_{s_2}\vec{X} \cdot \partial_{s_2}\vec{X}).\end{aligned}$$

Plugging in (A.3) gives

$$\begin{aligned}\sum_{i,j=1}^{2} g^{ij}(\partial_{s_q}\partial_{s_i}\vec{X}) \cdot (\partial_{s_j}\vec{X}) &= \frac{1}{2}\sum_{i,j=1}^{2} g^{ij}\partial_{s_q}(g_{ij}), \\ &= \frac{1}{\sqrt{\det(g)}}\partial_{s_q}\sqrt{\det(g)},\end{aligned}$$

where the last equality follows from Proposition A.2.2. □

Define Laplace-Beltrami Operator

We define the surface Laplacian (or Laplace-Beltrami) operator in a similar way as for the usual Laplacian:

$$\Delta_\Gamma \omega := \nabla_\Gamma \cdot \nabla_\Gamma \omega, \tag{A.18}$$

where ω is a scalar function on Γ. Applying the surface divergence (A.15) to the surface gradient of ω (A.11) gives (in local coordinates)

$$[\Delta_\Gamma \omega] \circ \vec{X} = \sum_{i,j=1}^{2} g^{ij}\partial_{s_i}\left\{\sum_{p,q=1}^{2} g^{pq}\,\partial_{s_p}\tilde{\omega}\,\partial_{s_q}\vec{X}\right\} \cdot \partial_{s_j}\vec{X}. \tag{A.19}$$

However, there is a more convenient form in which we can put (A.19). Expanding slightly and applying the product rule gives

$$\begin{aligned}[\Delta_\Gamma \omega] \circ \vec{X} = \sum_{i,j=1}^{2}\sum_{p,q=1}^{2} g^{ij}\{&(\partial_{s_i}g^{pq})(\partial_{s_p}\tilde{\omega})\partial_{s_q}\vec{X} + g^{pq}(\partial_{s_i}\partial_{s_p}\tilde{\omega})\partial_{s_q}\vec{X} + \\ &+ g^{pq}(\partial_{s_p}\tilde{\omega})(\partial_{s_i}\partial_{s_q}\vec{X})\} \cdot \partial_{s_j}\vec{X}.\end{aligned}$$

This can be simplified slightly using (A.3):

$$[\Delta_\Gamma \omega] \circ \vec{X} = \sum_{i,j=1}^{2} \sum_{p,q=1}^{2} \{(\partial_{s_i} g^{pq})(\partial_{s_p} \tilde{\omega}) g^{ij} g_{qj} + g^{pq}(\partial_{s_i}\partial_{s_p}\tilde{\omega}) g^{ij} g_{qj} +$$

$$+ g^{pq} g^{ij} (\partial_{s_p}\tilde{\omega})(\partial_{s_i}\partial_{s_q}\vec{X}) \cdot \partial_{s_j}\vec{X}\},$$

followed by plugging in (A.5) gives

$$\begin{aligned}[\Delta_\Gamma \omega] \circ \vec{X} = & \sum_{i=1}^{2} \sum_{p,q=1}^{2} \{(\partial_{s_i} g^{pq})(\partial_{s_p}\tilde{\omega})\delta^i_q + g^{pq}(\partial_{s_i}\partial_{s_p}\tilde{\omega})\delta^i_q\} + \\ & + \sum_{i,j=1}^{2} \sum_{p,q=1}^{2} \{g^{pq} g^{ij} (\partial_{s_p}\tilde{\omega})(\partial_{s_i}\partial_{s_q}\vec{X}) \cdot \partial_{s_j}\vec{X}\}.\end{aligned}$$

After using the definition of the 'Kronecker delta' (A.6), renaming certain indices, and rearranging slightly, we get

$$\begin{aligned}[\Delta_\Gamma \omega] \circ \vec{X} = & \sum_{i,j=1}^{2} \{(\partial_{s_j} g^{ij})(\partial_{s_i}\tilde{\omega}) + g^{ij}(\partial_{s_j}\partial_{s_i}\tilde{\omega})\} + \\ & + \sum_{p,q=1}^{2} g^{pq}(\partial_{s_p}\tilde{\omega}) \sum_{i,j=1}^{2} g^{ij}(\partial_{s_q}\partial_{s_i}\vec{X}) \cdot \partial_{s_j}\vec{X}.\end{aligned}$$

Plugging in Proposition A.2.3 simplifies to

$$\begin{aligned}[\Delta_\Gamma \omega] \circ \vec{X} = & \sum_{i,j=1}^{2} \{(\partial_{s_j} g^{ij})(\partial_{s_i}\tilde{\omega}) + g^{ij}(\partial_{s_j}\partial_{s_i}\tilde{\omega})\} + \\ & + \sum_{p,q=1}^{2} g^{pq}(\partial_{s_p}\tilde{\omega}) \frac{1}{\sqrt{\det(g)}} \partial_{s_q}\sqrt{\det(g)}.\end{aligned}$$

Relabeling p, q as i, j and factoring out $\frac{1}{\sqrt{\det(g)}}$ gives

$$[\Delta_\Gamma \omega] \circ \vec{X} = \frac{1}{\sqrt{\det(g)}} \sum_{i,j=1}^{2} \{\sqrt{\det(g)}(\partial_{s_j} g^{ij})(\partial_{s_i}\tilde{\omega}) + \sqrt{\det(g)} g^{ij}(\partial_{s_j}\partial_{s_i}\tilde{\omega}) +$$

$$+ (\partial_{s_j}\sqrt{\det(g)}) g^{ij}(\partial_{s_i}\tilde{\omega})\}.$$

Applying the product rule in reverse gives the alternate form of the surface Laplacian applied to a scalar function in local coordinates:

$$[\Delta_\Gamma \omega] \circ \vec{X} = \frac{1}{\sqrt{\det(g)}} \sum_{i,j=1}^{2} \partial_{s_j}[\sqrt{\det(g)}\ g^{ij}(\partial_{s_i}\tilde{\omega})]. \tag{A.20}$$

And the surface Laplacian of a 3x1 vector $\vec{\varphi}$ is computed by applying Δ_Γ to each component of the vector

$$[\Delta_\Gamma \vec{\varphi}] \circ \vec{X} = \begin{bmatrix} [\Delta_\Gamma \varphi_1] \circ \vec{X} \\ [\Delta_\Gamma \varphi_2] \circ \vec{X} \\ [\Delta_\Gamma \varphi_3] \circ \vec{X} \end{bmatrix}, \tag{A.21}$$

which is also a 3x1 vector.

A.3 Alternate Curvature Formulas

We would like to have another formula to compute the total curvature κ and Gaussian curvature κ_G. But first we make note of some basic vector identities that are valid pointwise:

$$\vec{a} \times \vec{b} = -\vec{b} \times \vec{a}, \tag{A.22}$$

$$(\vec{a} \times \vec{b}) \cdot (\vec{c} \times \vec{d}) = (\vec{a} \cdot \vec{c})(\vec{b} \cdot \vec{d}) - (\vec{a} \cdot \vec{d})(\vec{b} \cdot \vec{c}), \tag{A.23}$$

$$\vec{a} \times (\vec{b} \times \vec{c}) = \vec{b}(\vec{a} \cdot \vec{c}) - \vec{c}(\vec{a} \cdot \vec{b}), \tag{A.24}$$

where $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ are 3-D vectors and '×' is the cross product.

A.3.1 Alternate Total Curvature

Proposition A.3.1.

$$[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} = \frac{(\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{X}) + (\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{n})}{\sqrt{\det(g)}}, \tag{A.25}$$

$$\begin{aligned} [-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} \quad &= \kappa\, \vec{n}, \\ [-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} \cdot \vec{n} &= \kappa. \end{aligned} \tag{A.26}$$

Proof. First, we compute $-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})$ in local coordinates. According to (A.20) and (A.21), and using the definition (A.4) for the inverse metric, we have

$$[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} = \frac{-1}{\sqrt{\det(g)}} \Big\{ \partial_{s_1} \left(\frac{(\partial_{s_2}\vec{X} \cdot \partial_{s_2}\vec{X})\partial_{s_1}\vec{X} - (\partial_{s_1}\vec{X} \cdot \partial_{s_2}\vec{X})\partial_{s_2}\vec{X}}{\sqrt{\det(g)}} \right) + \\ + \partial_{s_2} \left(\frac{(\partial_{s_1}\vec{X} \cdot \partial_{s_1}\vec{X})\partial_{s_2}\vec{X} - (\partial_{s_1}\vec{X} \cdot \partial_{s_2}\vec{X})\partial_{s_1}\vec{X}}{\sqrt{\det(g)}} \right) \Big\}.$$

Now, by the vector identity (A.24), this becomes

$$[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} = \frac{-1}{\sqrt{\det(g)}} \Big\{ \partial_{s_1} \left(\frac{\partial_{s_2}\vec{X} \times (\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{X})}{\sqrt{\det(g)}} \right) + \\ + \partial_{s_2} \left(\frac{\partial_{s_1}\vec{X} \times (\partial_{s_2}\vec{X} \times \partial_{s_1}\vec{X})}{\sqrt{\det(g)}} \right) \Big\},$$

which, after using the definition of the normal vector (A.1) and the fact that $\sqrt{\det(g)} = |\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{X}|$, simplifies to

$$[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} = \frac{-1}{\sqrt{\det(g)}} \{ \partial_{s_1}(\partial_{s_2}\vec{X} \times \vec{n}) - \partial_{s_2}(\partial_{s_1}\vec{X} \times \vec{n}) \}.$$

Expanding this further gives

$$[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} = \frac{-1}{\sqrt{\det(g)}} \{ (\partial_{s_1}\partial_{s_2}\vec{X} \times \vec{n}) + (\partial_{s_2}\vec{X} \times \partial_{s_1}\vec{n}) + \\ - (\partial_{s_2}\partial_{s_1}\vec{X} \times \vec{n}) - (\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{n}) \},$$

followed by canceling and rearranging proves (A.25):

$$[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} = \frac{(\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{X}) + (\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{n})}{\sqrt{\det(g)}}.$$

Note that the vector field given by (A.25) is orthogonal to the tangent vectors $\partial_{s_1}\vec{X}$ and $\partial_{s_2}\vec{X}$; hence (A.25) is parallel to the normal vector $\vec{n}$. We use this to simplify

(A.25) further:

$$
\begin{aligned}
[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} \cdot \vec{n} &= \frac{(\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{X}) + (\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{n})}{\sqrt{\det(g)}} \cdot \vec{n} \\
&= \frac{1}{\det(g)}\{(\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{X}) \cdot (\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{X}) + \\
&\qquad + (\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{X}) \cdot (\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{n})\},
\end{aligned}
$$

where we have used (A.1). By using the vector identity (A.23), we can transform this into

$$
\begin{aligned}
[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} \cdot \vec{n} = \frac{1}{\det(g)}\{&(\partial_{s_1}\vec{X} \cdot \partial_{s_1}\vec{n})(\partial_{s_2}\vec{X} \cdot \partial_{s_2}\vec{X}) \\
&-(\partial_{s_1}\vec{X} \cdot \partial_{s_2}\vec{X})(\partial_{s_2}\vec{X} \cdot \partial_{s_1}\vec{n}) \\
&+(\partial_{s_1}\vec{X} \cdot \partial_{s_1}\vec{X})(\partial_{s_2}\vec{X} \cdot \partial_{s_2}\vec{n}) \\
&-(\partial_{s_1}\vec{X} \cdot \partial_{s_2}\vec{n})(\partial_{s_2}\vec{X} \cdot \partial_{s_1}\vec{X})\},
\end{aligned}
$$

and by definitions (A.3) and (A.8) this reduces to

$$
[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} \cdot \vec{n} = \frac{-h_{11}g_{22} + g_{12}h_{12} - g_{11}h_{22} + h_{21}g_{21}}{\det(g)}.
$$

Finally, using the formula for the inverse metric (A.4) gives

$$
[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} \cdot \vec{n} = -(g^{11}h_{11} + g^{12}h_{12} + g^{22}h_{22} + g^{21}h_{21}),
$$

which, by the equation for the total curvature (A.9), proves the alternate scalar curvature formula in (A.26)

$$
[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X} \cdot \vec{n} = \kappa.
$$

Because we know that $[-\Delta_\Gamma(\vec{X} \circ \vec{X}^{-1})] \circ \vec{X}$ is parallel to $\vec{n}$, we have that the total curvature vector (in terms of local coordinates) is given by the first equation in (A.26). □

A.3.2 Alternate Gaussian Curvature

This section derives another way of writing the Gaussian curvature that is similar in spirit to Proposition A.3.1.

Proposition A.3.2. *The Gaussian curvature κ_G (in local coordinates) can be written as*

$$\kappa_G = \vec{n} \cdot \frac{\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{n}}{\sqrt{\det(g)}}. \tag{A.27}$$

Proof. By the definition of the normal vector (A.1) and the vector identity (A.23), we have

$$\begin{aligned} \vec{n} \cdot \frac{\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{n}}{\sqrt{\det(g)}} &= \frac{(\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{X}) \cdot (\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{n})}{\det(g)}, \\ &= \frac{(\partial_{s_1}\vec{X} \cdot \partial_{s_1}\vec{n})(\partial_{s_2}\vec{X} \cdot \partial_{s_2}\vec{n}) - (\partial_{s_1}\vec{X} \cdot \partial_{s_2}\vec{n})(\partial_{s_2}\vec{X} \cdot \partial_{s_1}\vec{n})}{\det(g)}. \end{aligned}$$

Then, by the definition of the second fundamental form (A.8),

$$\vec{n} \cdot \frac{\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{n}}{\sqrt{\det(g)}} = \frac{h_{11}h_{22} - h_{21}h_{12}}{\det(g)},$$

which is just (by (A.10))

$$\vec{n} \cdot \frac{\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{n}}{\sqrt{\det(g)}} = \frac{\det(h)}{\det(g)} =: \kappa_G.$$

This proves the assertion. □

A.4 Integration by Parts

Next, we derive some integration by parts formulas.

A.4.1 Preliminary Formulas

Proposition A.4.1. *Let Γ be a smooth, closed surface. Let κ be the total scalar curvature (see* (A.9)*) of Γ and assume it is positive if Γ is elliptic (e.g. a sphere). Also assume the normal vector $\vec{n}$ points outward (away from the interior). Then the following relation is true*

$$\int_\Gamma \nabla_\Gamma \omega = \int_\Gamma \omega \kappa \vec{n} \tag{A.28}$$

for all smooth scalar $\omega : \Gamma \to \mathbb{R}$.

Proof. Let ω have compact support on Γ such that $\tilde{\omega} := \omega \circ \vec{X}$ has compact support in some open set U_i (recall that $\{U_i\}$ are local charts or reference domains). We start on the left side of (A.28) and do a change of variables, followed by plugging in the definition of the surface gradient operator (A.11):

$$\begin{aligned} \int_\Gamma \nabla_\Gamma \omega &= \int_{U_i} [\nabla_\Gamma \omega] \circ \vec{X} \sqrt{\det(g)} ds_1 ds_2, \\ &= \int_{U_i} \sqrt{\det(g)} \sum_{i,j=1}^{2} g^{ij} \partial_{s_i} \tilde{\omega} \partial_{s_j} \vec{X} ds_1 ds_2. \end{aligned}$$

Writing out the sum gives

$$\begin{aligned} \int_\Gamma \nabla_\Gamma \omega = \int_{U_i} \sqrt{\det(g)} [\partial_{s_1} \tilde{\omega} (g^{11} \partial_{s_1} \vec{X} + g^{12} \partial_{s_2} \vec{X}) + \\ \partial_{s_2} \tilde{\omega} (g^{21} \partial_{s_1} \vec{X} + g^{22} \partial_{s_2} \vec{X})] ds_1 ds_2, \end{aligned}$$

and after using the definition of the inverse metric (A.4) and rearranging slightly we get

$$\begin{aligned} \int_\Gamma \nabla_\Gamma \omega = \int_{U_i} \frac{1}{\sqrt{\det(g)}} \Big[\partial_{s_1} \tilde{\omega} \big((\partial_{s_2} \vec{X} \cdot \partial_{s_2} \vec{X}) \partial_{s_1} \vec{X} - (\partial_{s_1} \vec{X} \cdot \partial_{s_2} \vec{X}) \partial_{s_2} \vec{X} \big) + \\ + \partial_{s_2} \tilde{\omega} \big((\partial_{s_1} \vec{X} \cdot \partial_{s_1} \vec{X}) \partial_{s_2} \vec{X} - (\partial_{s_1} \vec{X} \cdot \partial_{s_2} \vec{X}) \partial_{s_1} \vec{X} \big) \Big] ds_1 ds_2. \end{aligned}$$

Applying the vector identity (A.24) simplifies this to

$$\int_{\Gamma} \nabla_{\Gamma} \omega = \int_{U_i} \frac{1}{\sqrt{\det(g)}} \Big[\partial_{s_1} \tilde{\omega} \big(\partial_{s_2} \vec{X} \times (\partial_{s_1} \vec{X} \times \partial_{s_2} \vec{X}) \big) +$$

$$+ \partial_{s_2} \tilde{\omega} \big(\partial_{s_1} \vec{X} \times (\partial_{s_2} \vec{X} \times \partial_{s_1} \vec{X}) \big) \Big] ds_1 ds_2,$$

and using the definition of the normal vector $\vec{n}$ (A.1) and making note of (A.22) reduces this further

$$\int_{\Gamma} \nabla_{\Gamma} \omega = \int_{U_i} \Big[\partial_{s_1} \tilde{\omega} \big(\partial_{s_2} \vec{X} \times \vec{n} \big) - \partial_{s_2} \tilde{\omega} \big(\partial_{s_1} \vec{X} \times \vec{n} \big) \Big] ds_1 ds_2.$$

Next, we integrate by parts the two terms in the integrand

$$\int_{\Gamma} \nabla_{\Gamma} \omega = \int_{U_i} \Big[- \tilde{\omega} \partial_{s_1} (\partial_{s_2} \vec{X} \times \vec{n}) + \tilde{\omega} \partial_{s_2} (\partial_{s_1} \vec{X} \times \vec{n}) \Big] ds_1 ds_2,$$

where there are no boundary terms because $\tilde{\omega}$ has compact support in U_i. After applying the product rule and rearranging, we get

$$\int_{\Gamma} \nabla_{\Gamma} \omega = \int_{U_i} \tilde{\omega} \frac{[(\partial_{s_1} \vec{n} \times \partial_{s_2} \vec{X}) + (\partial_{s_1} \vec{X} \times \partial_{s_2} \vec{n})]}{\sqrt{\det(g)}} \sqrt{\det(g)} ds_1 ds_2.$$

Using (A.26) for the total curvature vector in local coordinates gives

$$\int_{\Gamma} \nabla_{\Gamma} \omega = \int_{U_i} [\omega \circ \vec{X}] [(\kappa \vec{n}) \circ \vec{X}] \sqrt{\det(g)} ds_1 ds_2.$$

Finally, we apply a change of variables back to the surface Γ

$$\int_{\Gamma} \nabla_{\Gamma} \omega = \int_{\Gamma} \omega \kappa \vec{n},$$

which holds for any ω with compact support on Γ. But this formula is true for any scalar ω because it is always possible to decompose ω using a partition of unity. This allows ω to be written as a sum of scalar functions ω_i, each of which has compact support and the above formula is true for each individual ω_i. The assertion then follows (for general ω) by adding up the integrals. □

Proposition A.4.1 can be extended to vector functions.

Proposition A.4.2. *Under the same assumptions as in Proposition A.4.1, we have the following relations*

$$\int_\Gamma \nabla_\Gamma \vec{\varphi} = \int_\Gamma \kappa\, \vec{n} \otimes \vec{\varphi}, \tag{A.29}$$

$$\int_\Gamma \nabla_\Gamma \cdot \vec{\varphi} = \int_\Gamma \kappa\, \vec{n} \cdot \vec{\varphi}, \tag{A.30}$$

for all smooth vector functions $\vec{\varphi} : \Gamma \to \mathbb{R}^3$.

Proof. Equation (A.29) is obtained by applying (A.28) to each component of $\vec{\varphi}$ and (A.30) comes from taking the trace of the matrix equation (A.29). □

A.4.2 Main Integration by Parts Formula

Proposition A.4.3. *Let* φ *and* η *be smooth scalar functions on* Γ *(i.e.* $\varphi, \eta : \Gamma \to \mathbb{R}$*). Then the following integration by parts formula is true*

$$-\int_\Gamma \varphi \Delta_\Gamma \eta = \int_\Gamma \nabla_\Gamma \varphi \cdot \nabla_\Gamma \eta. \tag{A.31}$$

Proof. Because $\nabla_\Gamma \cdot (\varphi \nabla_\Gamma \eta) = \nabla_\Gamma \varphi \cdot \nabla_\Gamma \eta + \varphi \Delta_\Gamma \eta$, we have

$$\int_\Gamma (\nabla_\Gamma \varphi \cdot \nabla_\Gamma \eta + \varphi \Delta_\Gamma \eta) = \int_\Gamma \nabla_\Gamma \cdot (\varphi \nabla_\Gamma \eta) = \int_\Gamma \kappa (\vec{n} \cdot \nabla_\Gamma \eta) \varphi,$$

where the last equality follows from (A.30) applied to the vector $\varphi \nabla_\Gamma \eta$. But $\nabla_\Gamma \eta$ is tangent to the surface Γ, so $\vec{n} \cdot \nabla_\Gamma \eta = 0$. This proves the assertion. □

Proposition A.4.3 also extends to vector functions.

Proposition A.4.4. *Let $\vec{\varphi} = (\varphi_1, \varphi_2, \varphi_3)$ and $\vec{\eta} = (\eta_1, \eta_2, \eta_3)$ be smooth vector functions on Γ (i.e. $\vec{\varphi}, \vec{\eta} : \Gamma \to \mathbb{R}^3$). Then* (A.31) *generalizes to*

$$- \int_{\Gamma} \vec{\varphi} \cdot \Delta_{\Gamma} \vec{\eta} = \int_{\Gamma} \nabla_{\Gamma} \vec{\varphi} \cdot \nabla_{\Gamma} \vec{\eta}, \tag{A.32}$$

where the 'dot' $(\cdot)$ in $\nabla_{\Gamma} \vec{\varphi} \cdot \nabla_{\Gamma} \vec{\eta}$ means

$$\nabla_{\Gamma} \vec{\varphi} \cdot \nabla_{\Gamma} \vec{\eta} := \sum_{k=1}^{3} \nabla_{\Gamma} \varphi_k \cdot \nabla_{\Gamma} \eta_k, \tag{A.33}$$

(i.e. the 'double dot product' of two matrices).

Proof. Clearly, we can apply (A.31) to the vector components

$$- \int_{\Gamma} \varphi_k \Delta_{\Gamma} \eta_k = \int_{\Gamma} \nabla_{\Gamma} \varphi_k \cdot \nabla_{\Gamma} \eta_k,$$

for $k = 1, 2, 3$. Then by summing over k, we get the assertion. □

A.5 Other Equality Relations

Here we derive some formulas that will be useful in Section A.6.

A.5.1 Preliminary Calculations

Proposition A.5.1. *Let $\vec{\varphi} = (\varphi_1, \varphi_2, \varphi_3)$ and $\vec{\eta} = (\eta_1, \eta_2, \eta_3)$ be smooth vector functions on Γ (i.e. $\vec{\varphi}, \vec{\eta} : \Gamma \to \mathbb{R}^3$), and let $\tilde{\vec{\varphi}} = \vec{\varphi} \circ \vec{X}$, $\tilde{\vec{\eta}} = \vec{\eta} \circ \vec{X}$ be the vector functions in local coordinates. Then the following is true*

$$\{[\nabla_{\Gamma} \vec{\varphi}] \circ \vec{X}\} \cdot \{[\nabla_{\Gamma} \vec{\eta}] \circ \vec{X}\} = \sum_{i,j=1}^{2} g^{ij} \partial_{s_i} \tilde{\vec{\varphi}} \cdot \partial_{s_j} \tilde{\vec{\eta}}.$$

Proof. By equations (A.12) and (A.13), we have

$$\begin{aligned}\{[\nabla_\Gamma \vec{\varphi}] \circ \vec{X}\} \cdot \{[\nabla_\Gamma \vec{\eta}] \circ \vec{X}\} &= \sum_{k=1}^{3} \{[\nabla_\Gamma \varphi_k] \circ \vec{X}\} \cdot \{[\nabla_\Gamma \eta_k] \circ \vec{X}\} =, \\ &= \sum_{k=1}^{3} \sum_{i,j=1}^{2} \sum_{p,q=1}^{2} g^{ij} g^{pq} \partial_{s_i} \tilde{\varphi}_k \partial_{s_p} \tilde{\eta}_k \partial_{s_j} \vec{X} \cdot \partial_{s_q} \vec{X}, \\ &= \sum_{i,j=1}^{2} \sum_{p,q=1}^{2} g^{ij} g^{pq} g_{jq} \sum_{k=1}^{3} \partial_{s_i} \tilde{\varphi}_k \partial_{s_p} \tilde{\eta}_k,\end{aligned}$$

where we used the definition of the first fundamental form (A.3) in the last step. By properties (A.5) and (A.6), the above equation simplifies to

$$\{[\nabla_\Gamma \vec{\varphi}] \circ \vec{X}\} \cdot \{[\nabla_\Gamma \vec{\eta}] \circ \vec{X}\} = \sum_{i,j=1}^{2} g^{ij} \partial_{s_i} \tilde{\vec{\varphi}} \cdot \partial_{s_j} \tilde{\vec{\eta}},$$

which is the assertion we want. □

A.5.2 Surface Gradient of the Normal Vector

Now we will get a relation between the curvature and the surface gradient of the normal vector.

Proposition A.5.2. *Let $\vec{n}$ be the normal vector. Then the following is true*

$$|\nabla_\Gamma \vec{n}|^2 = \kappa^2 - 2\kappa_G. \qquad \text{(A.34)}$$

If Γ is a 1-D closed curve, then $\kappa_G = 0$ and

$$|\nabla_\Gamma \vec{n}| = |\kappa|. \qquad \text{(A.35)}$$

Proof. We start on the right side of (A.34) and use Proposition A.3.1 to get

$$\begin{aligned}\kappa^2 = (\kappa\vec{n}) \cdot (\kappa\vec{n}) = \frac{1}{\det(g)} &[(\partial_{s_1} \vec{n} \times \partial_{s_2} \vec{X}) + (\partial_{s_1} \vec{X} \times \partial_{s_2} \vec{n})] \cdot \\ &\cdot [(\partial_{s_1} \vec{n} \times \partial_{s_2} \vec{X}) + (\partial_{s_1} \vec{X} \times \partial_{s_2} \vec{n})],\end{aligned}$$

which becomes

$$\kappa^2 = \frac{1}{\det(g)}[(\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{X}) \cdot (\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{X}) +$$
$$+2(\partial_{s_1}\vec{n} \times \partial_{s_2}\vec{X}) \cdot (\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{n}) +$$
$$+(\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{n}) \cdot (\partial_{s_1}\vec{X} \times \partial_{s_2}\vec{n})].$$

Using the vector identity (A.23) transforms this into

$$\kappa^2 = \frac{1}{\det(g)}[(\partial_{s_1}\vec{n} \cdot \partial_{s_1}\vec{n})(\partial_{s_2}\vec{X} \cdot \partial_{s_2}\vec{X}) - (\partial_{s_1}\vec{n} \cdot \partial_{s_2}\vec{X})(\partial_{s_2}\vec{X} \cdot \partial_{s_1}\vec{n}) +$$
$$+2(\partial_{s_1}\vec{n} \cdot \partial_{s_1}\vec{X})(\partial_{s_2}\vec{X} \cdot \partial_{s_2}\vec{n}) - 2(\partial_{s_1}\vec{n} \cdot \partial_{s_2}\vec{n})(\partial_{s_2}\vec{X} \cdot \partial_{s_1}\vec{X}) +$$
$$+(\partial_{s_1}\vec{X} \cdot \partial_{s_1}\vec{X})(\partial_{s_2}\vec{n} \cdot \partial_{s_2}\vec{n}) - (\partial_{s_1}\vec{X} \cdot \partial_{s_2}\vec{n})(\partial_{s_2}\vec{n} \cdot \partial_{s_1}\vec{X})],$$

and using definitions (A.3) and (A.8) changes this to

$$\kappa^2 = \frac{1}{\det(g)}[(\partial_{s_1}\vec{n} \cdot \partial_{s_1}\vec{n})g_{22} - h_{12}h_{21} +$$
$$+2h_{11}h_{22} - 2(\partial_{s_1}\vec{n} \cdot \partial_{s_2}\vec{n})g_{12} +$$
$$+g_{11}(\partial_{s_2}\vec{n} \cdot \partial_{s_2}\vec{n}) - h_{12}h_{21}].$$

Note that $h_{12} = h_{21}$. Rearranging gives

$$\kappa^2 = \frac{[(\partial_{s_1}\vec{n} \cdot \partial_{s_1}\vec{n})g_{22} - 2(\partial_{s_1}\vec{n} \cdot \partial_{s_2}\vec{n})g_{12} + g_{11}(\partial_{s_2}\vec{n} \cdot \partial_{s_2}\vec{n})]}{\det(g)} +$$
$$+2\frac{h_{11}h_{22} - h_{12}h_{21}}{\det(g)},$$

and plugging in the definition of the inverse metric (A.4) gives

$$\kappa^2 = [(\partial_{s_1}\vec{n} \cdot \partial_{s_1}\vec{n})g^{11} + (\partial_{s_1}\vec{n} \cdot \partial_{s_2}\vec{n})g^{12} + (\partial_{s_2}\vec{n} \cdot \partial_{s_1}\vec{n})g^{21} + g^{22}(\partial_{s_2}\vec{n} \cdot \partial_{s_2}\vec{n})] +$$
$$+2\frac{\det(h)}{\det(g)}.$$

Using (A.10) and writing more compactly, we have

$$\kappa^2 = \sum_{i,j=1}^{2} g^{ij}(\partial_{s_i}\vec{n} \cdot \partial_{s_j}\vec{n}) + 2\kappa_G.$$

Finally, by Proposition A.5.1, we get the assertion

$$\begin{aligned}\kappa^2 - 2\kappa_G &= \{[\nabla_\Gamma \vec{n}] \circ \vec{X}\} \cdot \{[\nabla_\Gamma \vec{n}] \circ \vec{X}\}, \\ &= |\nabla_\Gamma \vec{n}|^2,\end{aligned}$$

where there is a slight abuse of notation on the last line because we left off the mapping $\vec{X}$. □

Before continuing, we make note of a modified version of (A.35) for the tangent vector. Suppose Γ is a closed 1-D curve in the plane, and let $\vec{t} = (t_1, t_2)$ be the tangent vector with components t_1, t_2 and right-handed orientation. Then the outward pointing normal vector $\vec{n} = (n_1, n_2)$ is related to the tangent vector by a 90 degree rotation:

$$(n_1, n_2) = (t_2, -t_1).$$

Then, by (A.35), we have the following identity

$$\begin{aligned}|\nabla_\Gamma \vec{t}\,|^2 &= |\nabla_\Gamma t_1|^2 + |\nabla_\Gamma t_2|^2, \\ &= |\nabla_\Gamma n_2|^2 + |\nabla_\Gamma n_1|^2 = |\nabla_\Gamma \vec{n}|^2 = \kappa^2.\end{aligned}$$

Hence,

$$|\nabla_\Gamma \vec{t}\,| = \kappa. \tag{A.36}$$

A.6 Some Inequalities

This section derives some inequalities that are used in Chapter 4.

A.6.1 Scalar Inequality

Lemma A.6.1. *Let $\phi : \Gamma \to \mathbb{R}$ be a scalar function defined on Γ. Then we have the following inequality:*

$$\|\phi\|_{0,\Gamma} \leq C(\|\phi\,\kappa\|_{0,\Gamma} + \|\nabla_\Gamma \phi\|_{0,\Gamma}), \tag{A.37}$$

where $\|\cdot\|_{0,\Gamma}$ denotes the L^2 norm on Γ and $C > 0$ is a constant that depends on the diameter of Γ in the ambient space.

Proof. Let d be the dimension of the surface Γ and let $\vec{x}_0$ be the coordinates of the centroid of Γ (in the $d+1$ dimensional ambient space); so $\vec{x}_0$ is a $d+1$ dimensional constant vector. Then we have by Proposition A.2.1

$$\begin{aligned} \|\phi\|^2_{0,\Gamma} &= \int_\Gamma \phi^2 = \frac{1}{d}\int_\Gamma d\phi^2 = \frac{1}{d}\int_\Gamma \nabla_\Gamma \cdot (\vec{X} \circ \vec{X}^{-1})\phi^2, \\ &= \frac{1}{d}\int_\Gamma \nabla_\Gamma \cdot [(\vec{X} \circ \vec{X}^{-1}) - \vec{x}_0]\phi^2, \end{aligned}$$

because the surface divergence of a constant vector field is zero. By the product rule we get

$$\|\phi\|^2_{0,\Gamma} = \frac{1}{d}\int_\Gamma \nabla_\Gamma \cdot \{\phi^2[(\vec{X} \circ \vec{X}^{-1}) - \vec{x}_0]\} - \nabla_\Gamma(\phi^2) \cdot [(\vec{X} \circ \vec{X}^{-1}) - \vec{x}_0],$$

and using the integration by parts formula (A.30) on the first part of the integrand gives

$$\|\phi\|^2_{0,\Gamma} = \frac{1}{d}\int_\Gamma \phi^2[(\vec{X} \circ \vec{X}^{-1}) - \vec{x}_0] \cdot \vec{n}\kappa - \int_\Gamma 2\phi\nabla_\Gamma\phi \cdot [(\vec{X} \circ \vec{X}^{-1}) - \vec{x}_0],$$

where we replaced $\nabla_\Gamma(\phi^2)$ by $2\phi\nabla_\Gamma\phi$. After taking the absolute value and bounding the right-hand side, we get

$$\begin{aligned} \|\phi\|^2_{0,\Gamma} &\leq \int_\Gamma \phi^2\big|(\vec{X} \circ \vec{X}^{-1}) - \vec{x}_0\big||\vec{n}|\,|\kappa| + \int_\Gamma 2|\phi\nabla_\Gamma\phi|\,\big|(\vec{X} \circ \vec{X}^{-1}) - \vec{x}_0\big|, \\ &\leq C\int_\Gamma \phi^2|\kappa| + C\int_\Gamma |\phi\nabla_\Gamma\phi|, \end{aligned} \tag{A.38}$$

where $C = 2R$ with R being the 'maximal radius' of Γ:

$$R := \max_{\vec{x} \in \Gamma} \big| [\vec{x} - \vec{x}_0] \big|.$$

By Hölder's inequality, we have that

$$\int_\Gamma \phi^2 |\kappa| = \int_\Gamma \phi(\phi|\kappa|) \leq \|\phi\|_{0,\Gamma} \|\phi\kappa\|_{0,\Gamma},$$
$$\int_\Gamma |\phi \nabla_\Gamma \phi| = \int_\Gamma |\phi| \; |\nabla_\Gamma \phi| \leq \|\phi\|_{0,\Gamma} \|\nabla_\Gamma \phi\|_{0,\Gamma}.$$

Therefore, (A.38) becomes

$$\|\phi\|^2_{0,\Gamma} \leq C \left(\|\phi\|_{0,\Gamma} \|\phi\kappa\|_{0,\Gamma} + \|\phi\|_{0,\Gamma} \|\nabla_\Gamma \phi\|_{0,\Gamma} \right).$$

Ergo, we obtain the assertion (A.37). □

A.6.2 Poincaré For Tangential Vector Field

Next, we derive a Poincaré type inequality for a vector field on a 1-D surface that is purely tangential.

Lemma A.6.2. *Let Γ be a 1-D closed surface, and let $\vec{v}$ be a vector field on Γ that is purely tangential (i.e. $\vec{v} = \phi \vec{t}$, where $\phi : \Gamma \to \mathbb{R}$ and $\vec{t}$ is the unit tangent vector of Γ). Then we have the following inequality*

$$\|\vec{v}\|_{0,\Gamma} \leq C \|\nabla_\Gamma \vec{v}\|_{0,\Gamma}, \tag{A.39}$$

where $C > 0$ is a constant that depends on the diameter of Γ in the ambient space.

Proof. We begin with some preliminary computations:

$$\begin{aligned}|\nabla_\Gamma(\phi\vec{t}\,)|^2 &= (\vec{t}\otimes\nabla_\Gamma\phi+\phi\nabla_\Gamma\vec{t}\,)\cdot(\vec{t}\otimes\nabla_\Gamma\phi+\phi\nabla_\Gamma\vec{t}\,),\\ &= |\vec{t}\otimes\nabla_\Gamma\phi|^2+\phi^2|\nabla_\Gamma\vec{t}\,|^2+2(\vec{t}\otimes\nabla_\Gamma\phi)\cdot(\phi\nabla_\Gamma\vec{t}\,),\\ &= |\vec{t}\,|^2|\nabla_\Gamma\phi|^2+\phi^2|\nabla_\Gamma\vec{t}\,|^2+2\phi\nabla_\Gamma\phi\cdot(\vec{t}\cdot\nabla_\Gamma\vec{t}\,),\\ &= |\nabla_\Gamma\phi|^2+\phi^2|\nabla_\Gamma\vec{t}\,|^2,\end{aligned}$$

where the third term is the zero vector because

$$|\vec{t}\,|^2=\vec{t}\cdot\vec{t}=1 \;\Rightarrow\; \nabla_\Gamma(\vec{t}\cdot\vec{t})=2\vec{t}\cdot\nabla_\Gamma\vec{t}=0.$$

And by (A.36), we get

$$|\nabla_\Gamma(\phi\vec{t}\,)|^2=|\nabla_\Gamma\phi|^2+\phi^2\kappa^2,$$

which after integrating over Γ, and plugging in $\vec{v}=\phi\vec{t}$, becomes

$$\|\nabla_\Gamma\vec{v}\|_{0,\Gamma}^2=\|\nabla_\Gamma\phi\|_{0,\Gamma}^2+\|\phi\kappa\|_{0,\Gamma}^2. \tag{A.40}$$

But computing the L^2 norm of $\vec{v}$ gives

$$\begin{aligned}\|\vec{v}\|_{0,\Gamma}^2=\int_\Gamma\vec{v}\cdot\vec{v}=\int_\Gamma\phi^2&=\|\phi\|_{0,\Gamma}^2,\\ &\le 2C^2(\|\phi\kappa\|_{0,\Gamma}^2+\|\nabla_\Gamma\phi\|_{0,\Gamma}^2),\end{aligned}$$

by Lemma A.6.1 and use of a Cauchy inequality. The constant C is taken from Lemma A.6.1 and depends on the diameter of Γ in the ambient space. Finally, by (A.40), we have the assertion,

$$\|\vec{v}\|_{0,\Gamma}\le\sqrt{2}C\|\nabla_\Gamma\vec{v}\|_{0,\Gamma}.$$

□

Appendix B

Solving the Divergence Equation

Let Ω be a domain in $\mathbb{R}^2$ with boundary $\Gamma := \partial\Omega$. In the following proposition Ω only needs to be Lipschitz in order to define the function spaces $H^{1/2}(\Gamma)$ or $H^1(\Gamma)$. In particular, Proposition B.0.1 certainly holds for a smooth domain, as well as a continuous piecewise C^1 domain (see pages 84, 163, 234 of [1] and page 34 of [93]). This would be the case when Ω is a polygon or when curved edges are used (as for iso-parametric elements).

Proposition B.0.1. (Solve the Divergence Equation) *Let Ω be Lipschitz and Γ be its boundary, and let $\vec{n}$ be the unit outer normal to Γ. Let $\vec{r} \in [H^{1/2}(\Gamma)]^2$, $f \in L^2(\Omega)$, and suppose that the following compatibility condition holds: $\int_\Gamma \vec{r} \cdot \vec{n} = \int_\Omega f$. Then, there exists a $\vec{v} \in [H^1(\Omega)]^2$ such that*

$$\nabla \cdot \vec{v} = f, \text{ in } \Omega,$$

$$\vec{v} = \vec{r}, \text{ on } \Gamma,$$

and $\vec{v}$ satisfies the following bound

$$\|\vec{v}\|_{1,\Omega} \leq c(\|\vec{r}\|_{1/2,\Gamma} + \|f\|_{0,\Omega}),$$

for some constant $c > 0$.

Proof. First, let $\vec{z} \in [H^1(\Omega)]^2$ solve the following vector Laplace problem:

$$\Delta \vec{z} = 0, \text{ in } \Omega,$$

$$\vec{z} = \vec{r}, \text{ on } \Gamma,$$

where $\|\vec{z}\|_{1,\Omega} = \|\vec{r}\|_{1/2,\Gamma}$ by the definition of the $H^{1/2}(\Gamma)$ norm.

Next, by the surjectivity of the divergence operator [18], [42], [59] and the fact that $\int_\Omega f - \int_\Omega \nabla \cdot \vec{z} = \int_\Omega f - \int_\Gamma \vec{r} \cdot \vec{n} = 0$, there exists a $\vec{w} \in [H_0^1(\Omega)]^2$ that satisfies

$$\begin{aligned} \nabla \cdot \vec{w} &= f - \nabla \cdot \vec{z}, \text{ in } \Omega, \\ \vec{w} &= 0, \qquad\qquad \text{on } \Gamma, \end{aligned}$$

with a bound given by

$$\|\vec{w}\|_{1,\Omega} \le c_1 \|f - \nabla \cdot \vec{z}\|_{0,\Omega} \le c_2(\|f\|_{0,\Omega} + \|\vec{z}\|_{1,\Omega}) = c_2(\|f\|_{0,\Omega} + \|\vec{r}\|_{1/2,\Gamma}).$$

Therefore, by letting $\vec{v} = \vec{w} + \vec{z}$ we get that

$$\nabla \cdot \vec{v} = f, \text{ in } \Omega,$$

$$\vec{v} = \vec{r}, \text{ on } \Gamma,$$

with the following bound

$$\|\vec{v}\|_{1,\Omega} \le (1 + c_2)(\|\vec{r}\|_{1/2,\Gamma} + \|f\|_{0,\Omega}).$$

□

Appendix C

Construction of a Continuous Normal Vector

In Section 4.3.3, we use a special vector field (defined on the boundary Γ) denoted $\vec{n}_s$. This object is crucial to the well-posedness of the variational formulation, so its construction is detailed here.

C.1 Properties of $\vec{n}_s$

First, we replace Γ by Γ_h to denote the dependence of the domain boundary on the discretization parameter h (i.e. the mesh size), and let $\vec{n}$ be the unit normal vector of Γ_h. Then, there exists a vector field $\vec{n}_s$ that satisfies the following properties:

1. it is continuous and piecewise differentiable on Γ_h;
2. $\vec{n}_s \cdot \vec{n} = 1$ everywhere on Γ_h;
3. $\|\vec{n}_s\|_{1,\Gamma_h} \leq C$, where $C > 0$ is uniform in some sense (see subsequent sections).

Essentially, $\vec{n}_s$ approximates $\vec{n}$. If Γ_h were smooth, then $\vec{n}_s := \vec{n}$ and the above properties are clearly satisfied. However, if Γ_h is a polygon (as it would be for a triangulated domain) or a 'curved polygon' (when using iso-parametric elements), then it is not obvious what $\vec{n}_s$ should be. The following sections explain how to construct $\vec{n}_s$ for a regular polygon and a polygon with curved edges.

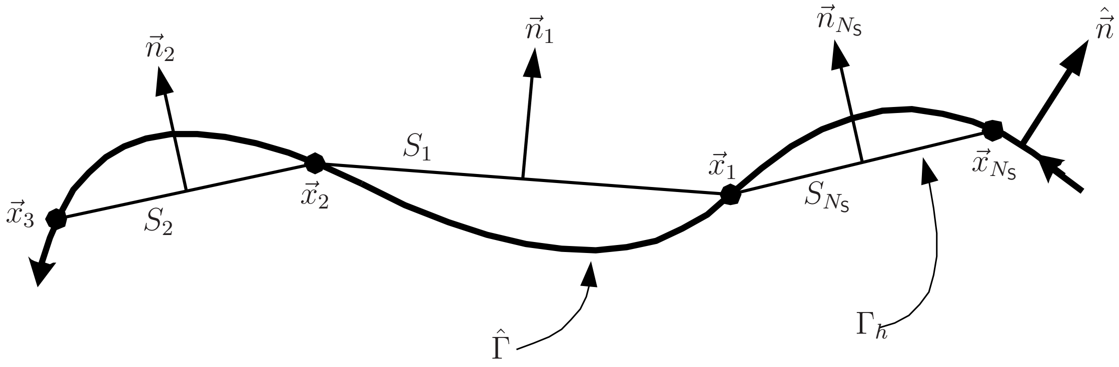


Figure C.1: Polygonal boundary Γ_h approximating a closed smooth curve $\hat{\Gamma}$. The polygon consists of a set of straight sides $\{S_k\}$ with vertices $\{\vec{x}_i\}$ denoted by thick dots (note: all vertices lie on the smooth boundary $\hat{\Gamma}$). The outward pointing normal vector of Γ_h is $\vec{n}$, and on each side S_k it is labeled $\vec{n}_k$. The normal vector of the smooth domain is $\hat{\vec{n}}$. Because Γ_h is closed, the vertex $\vec{x}_{N_\mathsf{S}}$ precedes $\vec{x}_1$.

C.2 Construction of $\vec{n}_s$ for a Polygon

Let Γ_h be a polygon that approximates a smooth boundary $\hat{\Gamma}$, and let S_k denote the kth side (see Figure C.1) with constant normal vector $\vec{n}_k$. Label the number of sides as N_S and let S_{Γ_h} be the set of sides:

$$\mathsf{S}_{\Gamma_h} := \{S_k : 1 \leq k \leq N_\mathsf{S}\}. \tag{C.1}$$

Next, let $\{\vec{x}_i\}_{i=1}^{N_\mathsf{S}}$ be the set of vertices of the polygon Γ_h, and let $\{\phi_i\}_{i=1}^{N_\mathsf{S}}$ be continuous piecewise linear 'hat' functions defined on Γ_h (i.e. each ϕ_i is in $\mathcal{P}_1(S_k)$ for $1 \leq k \leq N_\mathsf{S}$) such that:

$$\phi_i(\vec{x}_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \tag{C.2}$$

Now define the continuous vector field $\vec{n}_s$:

$$\vec{n}_s := \sum_{i=1}^{N_{\mathsf{S}}} \frac{\vec{n}_i + \vec{n}_{i-1}}{1 + \vec{n}_i \cdot \vec{n}_{i-1}} \phi_i, \tag{C.3}$$

where

$$\begin{aligned} \vec{x}_0 &:= \vec{x}_{N_{\mathsf{S}}}, \\ \vec{n}_0 &:= \vec{n}_{N_{\mathsf{S}}}, \\ \vec{x}_{N_{\mathsf{S}}+1} &:= \vec{x}_1, \\ \vec{n}_{N_{\mathsf{S}}+1} &:= \vec{n}_1. \end{aligned} \tag{C.4}$$

Note $\vec{n}_s$ is bounded in $L^\infty(\Gamma_h)$ as long as the angles of the polygon at the vertices are strictly bounded away from 0°.

Property (1) is satisfied because the set $\{\phi_i\}_{i=1}^{N_{\mathsf{S}}}$ is a continuous basis. To prove property (2), we first define some notation. Let ψ be some function defined on Γ_h, and let ψ_k be the restriction of ψ to the side S_k. Then we define $\psi(\vec{x}_k^+) := \psi_k(\vec{x}_k)$ and $\psi(\vec{x}_k^-) := \psi_{k-1}(\vec{x}_k)$, for any vertex $\vec{x}_k$. This is important if ψ is only continuous over each side S_k and discontinuous at each vertex.

Therefore, because $\vec{n}$ is piecewise constant (discontinuous) over Γ_h, note the following

$$\begin{aligned} &\vec{n}(\vec{x}_k^-) = \vec{n}_{k-1}, \\ &\vec{n}(\vec{x}_k^+) = \vec{n}_k, \\ &\phi_i(\vec{x}_k^-) = \phi_i(\vec{x}_k) = \phi_i(\vec{x}_k^+), \text{ for each } i, \end{aligned} \tag{C.5}$$

and then compute $\vec{n}_s \cdot \vec{n}$ on the left and right sides of each vertex:

$$\begin{aligned} \vec{n}_s(\vec{x}_k^-) \cdot \vec{n}(\vec{x}_k^-) &= \sum_{i=1}^{N_{\mathsf{S}}} \frac{\vec{n}(\vec{x}_k^-) \cdot (\vec{n}_i + \vec{n}_{i-1})}{1 + \vec{n}_i \cdot \vec{n}_{i-1}} \phi_i(\vec{x}_k^-), \\ &= \frac{\vec{n}_{k-1} \cdot (\vec{n}_k + \vec{n}_{k-1})}{1 + \vec{n}_k \cdot \vec{n}_{k-1}} = \frac{\vec{n}_{k-1} \cdot \vec{n}_k + 1}{1 + \vec{n}_k \cdot \vec{n}_{k-1}} = 1, \end{aligned} \tag{C.6}$$

$$\begin{aligned}\vec{n}_s(\vec{x}_k^+) \cdot \vec{n}(\vec{x}_k^+) &= \sum_{i=1}^{N_{\mathsf{S}}} \frac{\vec{n}(\vec{x}_k^+) \cdot (\vec{n}_i + \vec{n}_{i-1})}{1 + \vec{n}_i \cdot \vec{n}_{i-1}} \phi_i(\vec{x}_k^+), \\ &= \frac{\vec{n}_k \cdot (\vec{n}_k + \vec{n}_{k-1})}{1 + \vec{n}_k \cdot \vec{n}_{k-1}} = \frac{1 + \vec{n}_k \cdot \vec{n}_{k-1}}{1 + \vec{n}_k \cdot \vec{n}_{k-1}} = 1,\end{aligned} \tag{C.7}$$

where we used definition (C.2). Ergo, by (C.6) and (C.7)

$$(\vec{n}_s \cdot \vec{n})(\vec{x}_k) = 1, \text{ for all } k. \tag{C.8}$$

Now let $S_k \in \mathsf{S}_{\Gamma_h}$ be a side of Γ_h (i.e. $S_k \subset \Gamma_h$) for some k. Then by (C.3) (and the definition of $\{\phi_i\}$), $\vec{n}_s$ is a linear function on S_k, so $\vec{n}_s \cdot \vec{n}$ is also linear on S_k because $\vec{n}$ is constant on S_k. Combining with (C.8) implies that $\vec{n}_s \cdot \vec{n} = 1$ everywhere on S_k. Finally, since k was arbitrary we get that $\vec{n}_s \cdot \vec{n} = 1$ everywhere on Γ_h, thereby proving property (2).

Property (3) will be proved in Section C.4.

C.3 Construction of $\vec{n}_s$ for Curved Edges

Let Γ_h be a polygon with *curved* sides that approximates a smooth boundary $\hat{\Gamma}$, and let S_k denote the kth side (see Figure C.2) with normal vector $\vec{n}_k$ (note: $\vec{n}_k$ is not constant on each side S_k). Label the number of sides as N_{S}, and let S_{Γ_h} be defined by (C.1). Next, let $\{\vec{x}_i\}_{i=1}^{N_{\mathsf{S}}}$ be the set of vertices of the curved polygon Γ_h, with midpoints denoted $\{\vec{m}_i\}_{i=1}^{N_{\mathsf{S}}}$, and let $\{\phi_i\}_{i=1}^{N_{\mathsf{S}}}$ be the set of continuous piecewise linear 'hat' functions defined on Γ_h (i.e. each ϕ_i is in $\mathcal{P}_1(S_k)$ for $1 \leq k \leq N_{\mathsf{S}}$) such that (C.2) is satisfied. And for convenience, given $\vec{n}_k$, define $\vec{n}_k^1 := \vec{n}_k(\vec{x}_{k+1})$ and $\vec{n}_k^0 := \vec{n}_k(\vec{x}_k)$ at the endpoints of S_k.

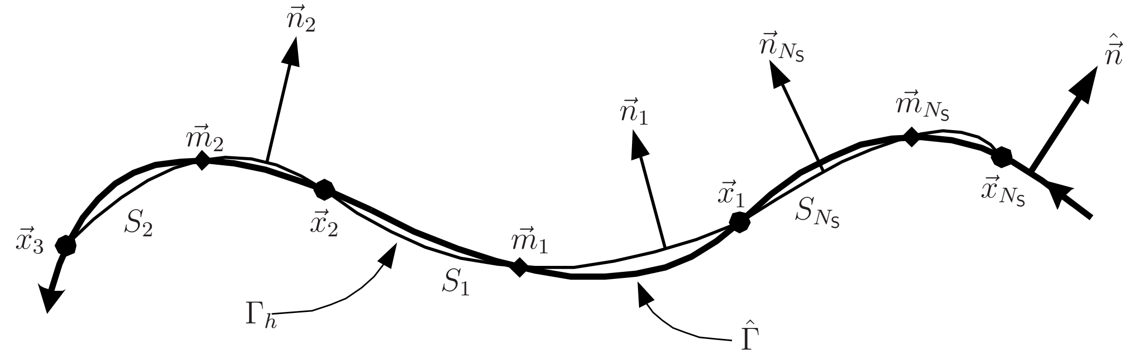


Figure C.2: Curved polygonal boundary Γ_h approximating a closed smooth curve $\hat{\Gamma}$ (denoted by the thicker curve). Γ_h consists of a set of curved sides $\{S_k\}$ with vertices $\{\vec{x}_i\}$ shown as thick dots and midpoints $\{\vec{m}_i\}$ shown as black diamonds (note: all vertices and midpoints lie on the smooth boundary $\hat{\Gamma}$). The outward pointing normal vector of Γ_h is $\vec{n}$, and on each side S_k it is labeled $\vec{n}_k$. The normal vector of the smooth domain is $\hat{\vec{n}}$. Because Γ_h is closed, the vertex $\vec{x}_{N_\mathsf{S}}$ precedes $\vec{x}_1$.

Now define a continuous vector field $\vec{r}$ on the curved polygon Γ_h:

$$\vec{r} := \sum_{i=1}^{N_{\mathsf{S}}} \frac{\vec{n}_i^0 + \vec{n}_{i-1}^1}{1 + \vec{n}_i^0 \cdot \vec{n}_{i-1}^1} \phi_i, \tag{C.9}$$

where we use definition (C.4). Note that $\vec{r}$ is bounded in $L^\infty(\Gamma_h)$ as long as the angle made by each pair of consecutive sides is strictly bounded away from 0° (i.e. $\vec{n}_i^0 \cdot \vec{n}_{i-1}^1 > c > -1$).

Just as in Section C.2, we have that

$$\begin{aligned} \vec{r}(\vec{x}_k) \cdot \vec{n}(\vec{x}_k^-) = \vec{r}(\vec{x}_k) \cdot \vec{n}_{k-1}(\vec{x}_k) = \vec{r}(\vec{x}_k) \cdot \vec{n}_{k-1}^1 = 1, \\ \vec{r}(\vec{x}_k) \cdot \vec{n}(\vec{x}_k^+) = \vec{r}(\vec{x}_k) \cdot \vec{n}_k(\vec{x}_k) = \vec{r}(\vec{x}_k) \cdot \vec{n}_k^0 = 1, \\ \Rightarrow \vec{r}(\vec{x}_k) \cdot \vec{n}(\vec{x}_k) = 1, \end{aligned} \tag{C.10}$$

for $1 \leq k \leq N_{\mathsf{S}}$. Therefore, $\vec{r} \cdot \vec{n}$ is a continuous function on Γ_h and $\vec{r} \cdot \vec{n} > c > 0$ assuming that the curved edges of Γ_h are not extremely distorted. Now we define $\vec{n}_s$ as

$$\vec{n}_s := \frac{\vec{r}}{\vec{r} \cdot \vec{n}}, \tag{C.11}$$

which is continuous piecewise C^1 because it is composed of continuous piecewise C^1 functions. Hence, property (1) is satisfied, and property (2) is trivially verified. Again, the proof of property (3) is delayed until Section C.4.

C.4 Proof of Bounded $H^1(\Gamma_h)$ Norm

The continuous vector $\vec{n}_s$ is used in Section 4.3.3 to prove the INF-SUP condition for the time-discrete problem. The INF-SUP constant in that proof depends on $1/\|\vec{n}_s\|_{1,\Gamma_h}$; ergo it is necessary that the $H^1(\Gamma_h)$ norm be bounded. Of course, for a fixed boundary Γ_h, this is true. But if the discrete domain (i.e. the polygon or curved polygon) is refined repeatedly, it is conceivable that the bound on $\|\vec{n}_s\|_{1,\Gamma_h}$ may increase as the mesh size h decreases. This would cause the INF-SUP constant to degrade and would be undesirable.

However, if Γ_h approximates a *smooth* closed curve $\hat{\Gamma}$, then $\|\vec{n}_s\|_{1,\Gamma_h}$ should be bounded uniformly with respect to h. This requires that the vertices (and midpoints when curved sides are used) of Γ_h always lie on the smooth curve $\hat{\Gamma}$. So, $\|\vec{n}_s\|_{1,\Gamma_h}$ should only depend on the true domain geometry being approximated.

In the subsequent sections, this is proven for the polygon and curved polygon case. The assumption here will be that the smooth curve $\hat{\Gamma}$ is at least C^2 regular. And the symbol h denotes the maximum length of all sides in Γ_h. Thus, Γ_h depends on the boundary mesh size h.

C.4.1 Polygon Case

The proof for a polygon uses the following estimate for the difference between the normal vector of a side of the approximating polygon and the normal vector of the smooth curve.

Proposition C.4.1. *Let $\hat{\Gamma}$ be a C^2 regular closed curve with approximating polygon Γ_h, where h denotes the maximum length of all sides of the polygon. Suppose that Γ_h approximates $\hat{\Gamma}$ such that $\hat{\Gamma}$ can be represented as the graph of a function f near the side $S_k \subset \Gamma_h$ (see Figure C.3). Hence, the derivative is bounded by a constant* *(i.e. $|f'| \leq C_f$). Then, the following estimate is true:*

$$
\begin{aligned}
|\vec{n}_k - \hat{\vec{n}}_{k+1}| &\leq C_0 h_k \|f''\|_{\infty,[0,h_k]} \leq C_1 h_k \kappa_k, \\
|\vec{n}_k - \hat{\vec{n}}_k| &\leq C_0 h_k \|f''\|_{\infty,[0,h_k]} \leq C_1 h_k \kappa_k,
\end{aligned}
\tag{C.12}
$$

where $\hat{\vec{n}}_k$ is the unit normal vector of $\hat{\Gamma}$ at the point $\vec{x}_k$, $\hat{\vec{n}}_{k+1}$ corresponds to $\vec{x}_{k+1}$, $\vec{n}_k$ is the unit normal vector of S_k, h_k is the length of S_k, κ_k is the maximum curvature of the portion of $\hat{\Gamma}$ between $\vec{x}_k$ and $\vec{x}_{k+1}$, and $C_0, C_1 > 0$ are constants.

Proof. Let θ be the angle between the unit vectors $\vec{n}_k$ and $\hat{\vec{n}}_{k+1}$ (as shown in Figure C.3). Then by basic geometry [86], we have that

$$
\tan\left(\frac{\theta}{2}\right) = \frac{|\vec{n}_k - \hat{\vec{n}}_{k+1}|/2}{|\vec{n}_k + \hat{\vec{n}}_{k+1}|/2} = \frac{|\vec{n}_k - \hat{\vec{n}}_{k+1}|}{|\vec{n}_k + \hat{\vec{n}}_{k+1}|}.
$$

Turning this around gives

$$
\begin{aligned}
|\vec{n}_k - \hat{\vec{n}}_{k+1}| &= |\vec{n}_k + \hat{\vec{n}}_{k+1}| \tan\left(\frac{\theta}{2}\right), \\
&\leq 2\left|\tan\left(\frac{\theta}{2}\right)\right| \leq 2|\tan\theta|,
\end{aligned}
\tag{C.13}
$$

where the last inequality is true because tan is a monotonically increasing function when $|\theta| < \pi/2$, which is satisfied in our case because we assumed f is a graph. But by the definition of the derivative and tan [85], we know $|\tan\theta| = |f'(0)|$. So, (C.13) becomes

$$
|\vec{n}_k - \hat{\vec{n}}_{k+1}| \leq 2|f'(0)| = 2|f'(0) - p'(0)|,
$$

where $p(s) := 0$ is a linear function that approximates f for $s \in [0, h_k]$. From standard interpolation theory [132], [20], we know

$$
|f'(0) - p'(0)| \leq C_0 h_k \|f''\|_{\infty,[0,h_k]},
$$

and combining with the previous inequality gives

$$|\vec{n}_k - \hat{\vec{n}}_{k+1}| \leq 2C_0 h_k \|f''\|_{\infty,[0,h_k]}, \tag{C.14}$$

To get the rest of the estimate, we use the following formula for the curvature of $\hat{\Gamma}$ in terms of the graph f [37]:

$$\kappa(s) = \frac{f''(s)}{(1 + (f'(s))^2)^{3/2}},$$

which implies that

$$\begin{aligned} |f''(s)| &\leq |1 + (f'(s))^2|^{3/2} |\kappa(s)| \leq |1 + C_f^2|^{3/2} \max_{0 \leq s \leq h_k} |\kappa(s)|, \\ &\leq |1 + C_f^2|^{3/2} \kappa_k, \end{aligned} \tag{C.15}$$

where κ_k is the maximum of the curvature. Therefore, we get the first assertion in (C.12) by plugging (C.15) into (C.14):

$$|\vec{n}_k - \hat{\vec{n}}_{k+1}| \leq C_1 h_k \kappa_k,$$

where $C_1 = 2C_0 |1 + C_f^2|^{3/2}$. The second assertion follows by the same argument. □

We now prove the following point-wise estimate lemma.

Lemma C.4.1. *Let $\hat{\Gamma}$ and Γ_h be as in Proposition C.4.1, and let S_k denote a side of Γ_h with length $h_k := |S_k|$. Suppose that $\hat{\Gamma}$ can be represented as the graph of a function on each side of the polygon Γ_h (recall Figure C.3). Furthermore, assume that Γ_h is shape regular in the following sense:*

$$b_1 h_{k-1} \leq h_k \leq b_2 h_{k-1}, \ 1 \leq k \leq N_{\mathcal{S}}, \tag{C.16}$$

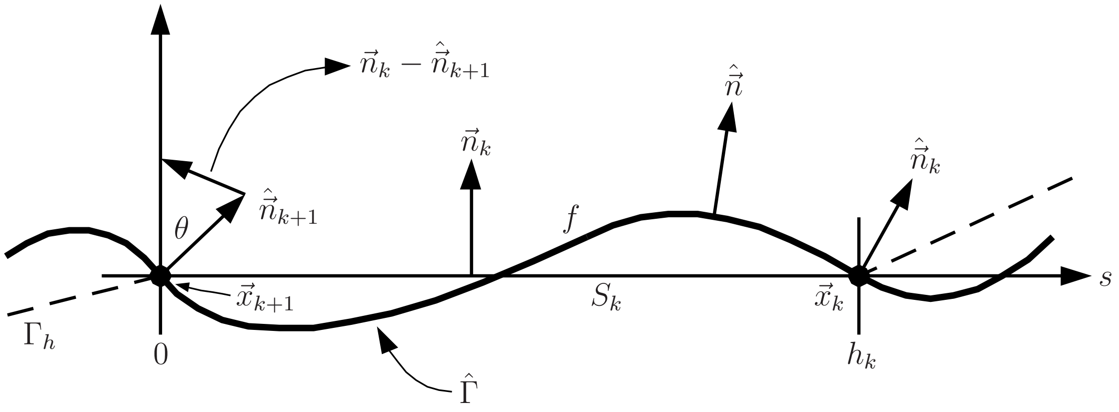


Figure C.3: Zoom-in of side S_k of the polygon Γ_h (dashed line) with the smooth (thick) curve $\hat{\Gamma}$. The side S_k of the polygon has vertices $\vec{x}_k$ and $\vec{x}_{k+1}$ with unit normal $\vec{n}_k$ and lies on the s axis with length $h_k := |S_k|$. It is assumed that Γ_h is sufficiently refined to allow for the smooth curve $\hat{\Gamma}$ to be represented (locally) as the graph of a function f on the interval $[0, h_k]$. The vertex $\vec{x}_k$ of the polygon is located at $(s = h_k, f(h_k) = 0)$, and $\vec{x}_{k+1}$ is at $(s = 0, f(0) = 0)$. The smooth curve has normal vector $\hat{\vec{n}}$, and we define $\hat{\vec{n}}_k := \hat{\vec{n}}(\vec{x}_k)$ and $\hat{\vec{n}}_{k+1} := \hat{\vec{n}}(\vec{x}_{k+1})$. The angle θ is between $\vec{n}_k$ and $\hat{\vec{n}}_{k+1}$.

where $b_1, b_2 > 0$ are uniform constants for all k (note: N_S is the total number of sides of Γ_h). Then the following point-wise estimate is true a.e. *(almost everywhere) on Γ_h*

$$|\nabla_{\Gamma_h} \vec{n}_s| \leq C_s \kappa_{\hat{\Gamma}}, \tag{C.17}$$

where $\vec{n}_s$ is the vector field defined on Γ_h by (C.3)*, $\kappa_{\hat{\Gamma}}$ is the maximum curvature of $\hat{\Gamma}$, and $C_s > 0$ is a uniform constant independent of h.*

Proof. Using (C.3), we know that

$$\vec{n}_s\Big|_{S_k} = \frac{\vec{n}_k + \vec{n}_{k-1}}{1 + \vec{n}_k \cdot \vec{n}_{k-1}} \phi_k + \frac{\vec{n}_{k+1} + \vec{n}_k}{1 + \vec{n}_{k+1} \cdot \vec{n}_k} \phi_{k+1},$$

which, by the definition of the 'surface gradient' (A.11), implies

$$\nabla_{\Gamma_h} \vec{n}_s\Big|_{S_k} = \frac{1}{h_k} \left[\frac{\vec{n}_{k+1} + \vec{n}_k}{1 + \vec{n}_{k+1} \cdot \vec{n}_k} - \frac{\vec{n}_k + \vec{n}_{k-1}}{1 + \vec{n}_k \cdot \vec{n}_{k-1}} \right] \otimes \vec{t}_k, \tag{C.18}$$

where $\vec{t}_k$ is the tangent vector on S_k. Next, we add and subtract $\hat{\vec{n}}_k$

$$\nabla_{\Gamma_h} \vec{n}_s\Big|_{S_k} = \frac{1}{h_k} \left[\left(\frac{\vec{n}_{k+1} + \vec{n}_k}{1 + \vec{n}_{k+1} \cdot \vec{n}_k} - \hat{\vec{n}}_k \right) + \left(\hat{\vec{n}}_k - \frac{\vec{n}_k + \vec{n}_{k-1}}{1 + \vec{n}_k \cdot \vec{n}_{k-1}} \right) \right] \otimes \vec{t}_k, \tag{C.19}$$

and define

$$G_1 := \frac{\vec{n}_{k+1} + \vec{n}_k}{1 + \vec{n}_{k+1} \cdot \vec{n}_k} - \hat{\vec{n}}_k, \quad G_2 := \hat{\vec{n}}_k - \frac{\vec{n}_k + \vec{n}_{k-1}}{1 + \vec{n}_k \cdot \vec{n}_{k-1}},$$

so we can rewrite (C.19) as

$$\nabla_{\Gamma_h} \vec{n}_s\Big|_{S_k} = \frac{1}{h_k} [G_1 + G_2] \otimes \vec{t}_k. \tag{C.20}$$

We proceed to manipulate G_1. Factoring out the denominator and rearranging gives

$$G_1 = \frac{1}{1 + \vec{n}_{k+1} \cdot \vec{n}_k} (\vec{n}_k - \hat{\vec{n}}_k + \vec{n}_{k+1} - \hat{\vec{n}}_k(\vec{n}_{k+1} \cdot \vec{n}_k)),$$

and adding and subtracting $\hat{\vec{n}}_{k+1}$ and $\hat{\vec{n}}_k$ gives

$$G_1 = \frac{1}{1+\vec{n}_{k+1}\cdot\vec{n}_k}[(\vec{n}_k - \hat{\vec{n}}_k) + (\vec{n}_{k+1} - \hat{\vec{n}}_{k+1}) +$$

$$+(\hat{\vec{n}}_{k+1} - \hat{\vec{n}}_k) + (\hat{\vec{n}}_k - \hat{\vec{n}}_k(\vec{n}_{k+1}\cdot\vec{n}_k))].$$

After more manipulation, this becomes

$$G_1 = \frac{1}{1+\vec{n}_{k+1}\cdot\vec{n}_k}\{(\vec{n}_k - \hat{\vec{n}}_k) + (\vec{n}_{k+1} - \hat{\vec{n}}_{k+1}) +$$

$$+(\hat{\vec{n}}_{k+1} - \hat{\vec{n}}_k) + \hat{\vec{n}}_k[\vec{n}_k\cdot(\vec{n}_k - \vec{n}_{k+1})]\},$$

where we used the fact that $\vec{n}_k\cdot\vec{n}_k = 1$. Bounding G_1 then gives

$$\begin{aligned}
|G_1| &\leq \frac{|\vec{n}_k - \hat{\vec{n}}_k| + |\vec{n}_{k+1} - \hat{\vec{n}}_{k+1}| + |\hat{\vec{n}}_{k+1} - \hat{\vec{n}}_k| + |\hat{\vec{n}}_k|\,|\vec{n}_k|\,|\vec{n}_k - \vec{n}_{k+1}|}{|1+\vec{n}_{k+1}\cdot\vec{n}_k|}, \\
&\leq \frac{|\vec{n}_k - \hat{\vec{n}}_k| + |\vec{n}_{k+1} - \hat{\vec{n}}_{k+1}| + |\hat{\vec{n}}_{k+1} - \hat{\vec{n}}_k| + |\vec{n}_k - \hat{\vec{n}}_{k+1} + \hat{\vec{n}}_{k+1} - \vec{n}_{k+1}|}{|1+\vec{n}_{k+1}\cdot\vec{n}_k|}, \\
&\leq \frac{|\vec{n}_k - \hat{\vec{n}}_k| + 2|\vec{n}_{k+1} - \hat{\vec{n}}_{k+1}| + |\hat{\vec{n}}_{k+1} - \hat{\vec{n}}_k| + |\vec{n}_k - \hat{\vec{n}}_{k+1}|}{|1+\vec{n}_{k+1}\cdot\vec{n}_k|},
\end{aligned}$$

and after using (C.12) and (C.29), this simplifies to

$$\begin{aligned}
|G_1| &\leq \frac{C_1 h_k\kappa_k + 2C_1 h_{k+1}\kappa_{k+1} + C_2 h_k\kappa_k + C_1 h_k\kappa_k}{|1+\vec{n}_{k+1}\cdot\vec{n}_k|}, \\
&\leq C_3\frac{h_k + h_{k+1}}{|1+\vec{n}_{k+1}\cdot\vec{n}_k|}\kappa_{\hat{\Gamma}},
\end{aligned}$$

where $\kappa_{\hat{\Gamma}}$ is the maximum curvature of $\hat{\Gamma}$. Next, we assume Γ_h is sufficiently refined so that $\vec{n}_{k+1}\cdot\vec{n}_k > 0$ and use the shape regularity (C.16) to get

$$|G_1| \leq C_3\frac{h_k + b_2 h_k}{|1+\vec{n}_{k+1}\cdot\vec{n}_k|}\kappa_{\hat{\Gamma}} \leq C_4 h_k\kappa_{\hat{\Gamma}}.$$

After noting that G_2 satisfies the same bound (with a different constant), we go back to (C.20) and bound to get

$$|\nabla_{\Gamma_h}\vec{n}_s|\Big|_{S_k} \leq \frac{1}{h_k}[C_4 h_k + C_5 h_k]\kappa_{\hat{\Gamma}} \leq C_s\kappa_{\hat{\Gamma}},$$

which is bounded independent of h and k. Ergo, we get the assertion (C.17) on all of Γ_h. □

By Lemma C.4.1, we have the estimate for the $H^1(\Gamma_h)$ semi-norm of $\vec{n}_s$:

$$\begin{aligned} \|\nabla_{\Gamma_h} \vec{n}_s\|_{0,\Gamma_h}^2 = \int_{\Gamma_h} |\nabla_{\Gamma_h} \vec{n}_s|^2 \leq C_s^2 \kappa_{\hat{\Gamma}}^2 \int_{\Gamma_h} 1 = C_s^2 \kappa_{\hat{\Gamma}}^2 |\Gamma_h| \leq C_s^2 \kappa_{\hat{\Gamma}}^2 |\hat{\Gamma}|, \\ \Rightarrow \|\nabla_{\Gamma_h} \vec{n}_s\|_{0,\Gamma_h} \leq C_s \kappa_{\hat{\Gamma}} |\hat{\Gamma}|^{1/2}. \end{aligned} \tag{C.21}$$

The $L^2(\Gamma_h)$ norm is estimated by first computing

$$\begin{aligned} \frac{\vec{n}_i + \vec{n}_{i-1}}{1 + \vec{n}_i \cdot \vec{n}_{i-1}} \cdot \frac{\vec{n}_i + \vec{n}_{i-1}}{1 + \vec{n}_i \cdot \vec{n}_{i-1}} = \frac{2(1 + \vec{n}_i \cdot \vec{n}_{i-1})}{(1 + \vec{n}_i \cdot \vec{n}_{i-1})^2} = \\ = \frac{2}{1 + \vec{n}_i \cdot \vec{n}_{i-1}} \leq 2, \text{ for } 1 \leq i \leq N_{\mathsf{S}}, \end{aligned} \tag{C.22}$$

as long as $\vec{n}_i \cdot \vec{n}_{i-1} > 0$ for each i, which is easily satisfied if Γ_h is a sufficiently refined approximation of $\hat{\Gamma}$. This implies that

$$\|\vec{n}_s\|_{\infty,\Gamma_h} \leq 2. \tag{C.23}$$

So, we get the following estimate:

$$\begin{aligned} \|\vec{n}_s\|_{0,\Gamma_h}^2 = \int_{\Gamma_h} \vec{n}_s \cdot \vec{n}_s \leq \|\vec{n}_s\|_{\infty,\Gamma_h}^2 \int_{\Gamma_h} 1 \leq 4|\Gamma_h| \leq 4|\hat{\Gamma}|, \\ \Rightarrow \|\vec{n}_s\|_{0,\Gamma_h} \leq 2|\hat{\Gamma}|^{1/2}. \end{aligned} \tag{C.24}$$

Therefore, the full $H^1(\Gamma_h)$ norm is bounded independently of h:

$$\|\vec{n}_s\|_{1,\Gamma_h} \leq (4 + C_s^2 \kappa_{\hat{\Gamma}}^2)^{1/2} |\hat{\Gamma}|^{1/2}, \tag{C.25}$$

so property (3) in Section C.1 is satisfied.

C.4.2 Curved Edge Case

We follow a similar exposition as in the previous section. First, we prove a similar result as in Proposition C.4.1 for curved sides.

Proposition C.4.2. *Let $\hat{\Gamma}$ be a C^2 regular closed curve with approximating* curved *polygon Γ_h, where h denotes the maximum length of all curved sides of the polygon. Suppose that Γ_h approximates $\hat{\Gamma}$ such that $\hat{\Gamma}$ can be represented as the graph of a function f near the side $S_k \subset \Gamma_h$ with $f'(0) = 0$ (see Figure C.4). Hence, the derivative is bounded (i.e. $|f'| \leq C_f$). Then, the following estimate is true:*

$$\begin{aligned} |\vec{n}_k^1 - \hat{\vec{n}}_{k+1}| &\leq C_1 h_k \kappa_k, \\ |\vec{n}_k^0 - \hat{\vec{n}}_k| &\leq C_2 h_k \kappa_k, \end{aligned} \tag{C.26}$$

where $\hat{\vec{n}}_k$ is the unit normal vector of $\hat{\Gamma}$ at the point $\vec{x}_k$, $\hat{\vec{n}}_{k+1}$ corresponds to $\vec{x}_{k+1}$, $\vec{n}_k^0$ and $\vec{n}_k^1$ is the unit normal vector of S_k at $\vec{x}_k$ and $\vec{x}_{k+1}$, respectively, h_k is the length of S_k, κ_k is the maximum curvature of the portion of $\hat{\Gamma}$ between $\vec{x}_k$ and $\vec{x}_{k+1}$, and $C_1, C_2 > 0$ are constants.

Proof. Let θ be the angle between the unit vectors $\vec{n}_k^1$ and $\hat{\vec{n}}_{k+1}$ (see Figure C.4). By the same argument as in the proof of Proposition C.4.1 (see (C.13)), we get

$$|\vec{n}_k^1 - \hat{\vec{n}}_{k+1}| \leq 2|\tan\theta|. \tag{C.27}$$

Let $p(s)$ be the quadratic function that corresponds to the side S_k, expressed in the local coordinate $s \in [0, l_k]$. Then (C.27) can be rewritten as

$$|\vec{n}_k^1 - \hat{\vec{n}}_{k+1}| \leq 2|\tan\theta| = 2|p'(0)| = 2|p'(0) - f'(0)|,$$

where the last equality is because $f'(0) = 0$. Because p is an approximation of f, by basic interpolation theory [132], [20], we have

$$|p'(0) - f'(0)| \leq C' l_k \|f''\|_{\infty,[0,l_k]},$$

and combining with the previous inequality gives

$$|\vec{n}_k^1 - \hat{\vec{n}}_{k+1}| \leq 2C' h_k \|f''\|_{\infty,[0,l_k]}, \tag{C.28}$$

where we used $l_k \leq h_k$. The rest of the estimate follows by the same argument in the proof of Proposition C.4.1:

$$|\vec{n}_k^1 - \hat{\vec{n}}_{k+1}| \leq C_1 h_k \kappa_k,$$

where κ_k is the maximum curvature of $\hat{\Gamma}$ between $\vec{x}_k$ and $\vec{x}_{k+1}$, and $C_1 > 0$ is a uniform constant with respect to k. The second assertion in (C.26) follows by the same argument but with a different local coordinate system (i.e. a different local parameterization). □

We also need a short proposition for the difference of the normal vector $\hat{\vec{n}}$ between the vertices $\vec{x}_k$ and $\vec{x}_{k+1}$.

Proposition C.4.3. *Assume the same hypothesis and definitions as in Proposition C.4.2. Then, the following estimate is true:*

$$|\hat{\vec{n}}_k - \hat{\vec{n}}_{k+1}| \leq C_1 h_k \kappa_k, \tag{C.29}$$

where κ_k is the maximum curvature of the portion of $\hat{\Gamma}$ between $\vec{x}_k$ and $\vec{x}_{k+1}$, and $C_1 > 0$ is a constant.

Proof. Let f be as in Proposition C.4.2 (in particular, f is defined on $[0, l_k]$, f' is bounded, and $f(0) = f'(0) = 0$; see Figure C.4). Then just as in Proposition C.4.2, we get

$$|\hat{\vec{n}}_k - \hat{\vec{n}}_{k+1}| \leq 2|\tan\theta| = 2|f'(l_k)| = 2|f'(l_k) - f'(0)|,$$

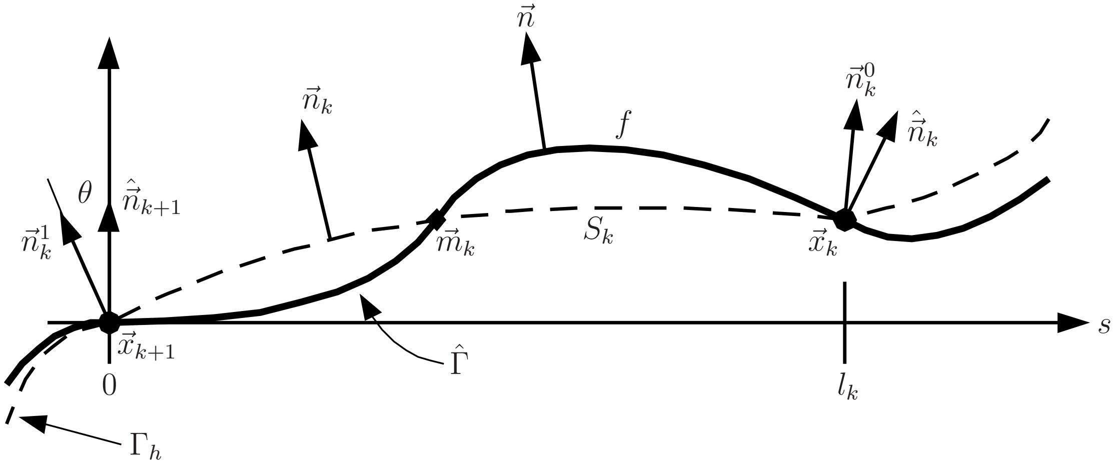


Figure C.4: Zoom-in of side S_k of the *curved* polygon Γ_h (dashed line) with the smooth (thick) curve $\hat{\Gamma}$. The side S_k of the polygon has vertices $\vec{x}_k$ and $\vec{x}_{k+1}$, with middle vertex $\vec{m}_k$ labeled as a black diamond. The unit normal vector of S_k is $\vec{n}_k$ and is *not* constant along S_k. The length of S_k is denoted $h_k := |S_k|$. Γ_h is assumed to be sufficiently refined to allow for the smooth curve $\hat{\Gamma}$ to be represented (locally) as the graph of a function f on the interval $[0, l_k]$, with $f'(0) = 0$. The vertex $\vec{x}_k$ of the polygon is located at $(s = l_k, f(l_k))$, and $\vec{x}_{k+1}$ is at $(s = 0, f(0) = 0)$. The smooth curve has normal vector $\hat{\vec{n}}$, and we define $\hat{\vec{n}}_k := \hat{\vec{n}}(\vec{x}_k)$ and $\hat{\vec{n}}_{k+1} := \hat{\vec{n}}(\vec{x}_{k+1})$. The normal vector of the curved side at the endpoints is defined as $\vec{n}_k^1 := \vec{n}_k(\vec{x}_{k+1})$ and $\vec{n}_k^0 := \vec{n}_k(\vec{x}_k)$. The angle θ is between $\vec{n}_k^1$ and $\hat{\vec{n}}_{k+1}$.

where θ is the angle between $\hat{\vec{n}}_k$ and $\hat{\vec{n}}_{k+1}$. By the fundamental theorem of calculus [83], we have

$$|\hat{\vec{n}}_k - \hat{\vec{n}}_{k+1}| \leq 2|\int_0^{l_k} f''(s)ds| \leq 2l_k\|f''\|_{\infty,[0,l_k]}.$$

The assertion (C.29) follows by similar reasoning as in the proof of Proposition C.4.1. □

We now prove a proposition and a lemma regarding the surface gradient of $\vec{n}$ and $\vec{r}$ in the definition (C.11) of $\vec{n}_s$ for the curved polygon case.

Proposition C.4.4. *Assume the same hypothesis and definitions as in Proposition C.4.2 (see Figure C.4). Then, the following estimate is true:*

$$|\nabla_{\Gamma_h}\vec{n}|\Big|_{S_k} = |\nabla_{\Gamma_h}\vec{n}_k| \leq C_n\kappa_k, \tag{C.30}$$

where $\vec{n}_k$ is the unit normal vector of Γ_h on S_k, κ_k is the maximum curvature of $\hat{\Gamma}$ between $\vec{x}_k$ and $\vec{x}_{k+1}$, and $C_n > 0$ is a uniform constant with respect to h and k.

Proof. By (A.35), we know that

$$|\nabla_{\Gamma_h}\vec{n}_k| = |\kappa_{S_k}|, \tag{C.31}$$

where κ_{S_k} is the curvature of the quadratic curved side S_k. Next, let p be the quadratic function defined in the proof of Proposition C.4.2 (i.e. p is a function of the local coordinate s). Then, by writing (C.31) in terms of the local coordinate s, we get

$$\begin{aligned}|\nabla_{\Gamma_h}\vec{n}_k| = |\kappa_{S_k}| = \left|\frac{p''}{(1+(p')^2)^{3/2}}\right| &\leq \|p''\|_{\infty,[0,l_k]} \\ &\leq \|p'' - f''\|_{\infty,[0,l_k]} + \|f''\|_{\infty,[0,l_k]},\end{aligned}$$

where f is the local parameterization of $\hat{\Gamma}$. By standard Sobolev interpolation theory [20], we get

$$|\nabla_{\Gamma_h} \vec{n}_k| \leq C_0 \|f''\|_{\infty,[0,l_k]}, \tag{C.32}$$

and the rest follows by the same argument as in the proof of Proposition C.4.1:

$$|\nabla_{\Gamma_h} \vec{n}_k| \leq C_n \kappa_k.$$

□

Lemma C.4.2. *Let $\hat{\Gamma}$ and Γ_h be as in Proposition C.4.1, and let S_k denote a side of Γ_h with length $h_k := |S_k|$. Furthermore, assume that Γ_h is shape regular in the following sense:*

$$b_1 h_{k-1} \leq h_k \leq b_2 h_{k-1}, \ 1 \leq k \leq N_{\mathcal{S}}, \tag{C.33}$$

where $b_1, b_2 > 0$ are uniform constants for all k (note: $N_{\mathcal{S}}$ is the total number of sides of Γ_h). Then the following point-wise estimate is true a.e. *on Γ_h*

$$|\nabla_{\Gamma_h} \vec{r}| \leq C_r \kappa_{\hat{\Gamma}}, \tag{C.34}$$

where $\vec{r}$ is the vector field defined on Γ_h by (C.9) *and $C_r > 0$ is a uniform constant independent of h.*

Proof. Using (C.9), we have that

$$\vec{r}\,\Big|_{S_k} = \frac{\vec{n}_k^0 + \vec{n}_{k-1}^1}{1 + \vec{n}_k^0 \cdot \vec{n}_{k-1}^1} \phi_k + \frac{\vec{n}_{k+1}^0 + \vec{n}_k^1}{1 + \vec{n}_{k+1}^0 \cdot \vec{n}_k^1} \phi_{k+1},$$

which, by the definition of the 'surface gradient' (A.11), implies

$$\nabla_{\Gamma_h} \vec{r}\,\Big|_{S_k} = \frac{1}{h_k} \left[\frac{\vec{n}_{k+1}^0 + \vec{n}_k^1}{1 + \vec{n}_{k+1}^0 \cdot \vec{n}_k^1} - \frac{\vec{n}_k^0 + \vec{n}_{k-1}^1}{1 + \vec{n}_k^0 \cdot \vec{n}_{k-1}^1} \right] \otimes \vec{t}_k, \tag{C.35}$$

where $\vec{t}_k$ is the tangent vector on S_k. Adding and subtracting some convenient terms changes (C.35) to

$$\begin{aligned}\nabla_{\Gamma_h}\vec{r}\,\big|_{S_k} = \frac{1}{h_k}\Bigg[&\frac{\vec{n}^0_{k+1}+\vec{n}^1_k}{1+\vec{n}^0_{k+1}\cdot\vec{n}^1_k} - \hat{\vec{n}}_{k+1} + \\ &+(\hat{\vec{n}}_{k+1}-\hat{\vec{n}}_k)+ \\ &+\hat{\vec{n}}_k - \frac{\vec{n}^0_k+\vec{n}^1_{k-1}}{1+\vec{n}^0_k\cdot\vec{n}^1_{k-1}}\Bigg]\otimes\vec{t}_k.\end{aligned} \tag{C.36}$$

Next, define:

$$G_1 := \frac{\vec{n}^0_{k+1}+\vec{n}^1_k}{1+\vec{n}^0_{k+1}\cdot\vec{n}^1_k} - \hat{\vec{n}}_{k+1}, \tag{C.37}$$

$$G_2 := (\hat{\vec{n}}_{k+1}-\hat{\vec{n}}_k), \tag{C.38}$$

$$G_3 := \hat{\vec{n}}_k - \frac{\vec{n}^0_k+\vec{n}^1_{k-1}}{1+\vec{n}^0_k\cdot\vec{n}^1_{k-1}}, \tag{C.39}$$

and rewrite (C.36) as

$$\nabla_{\Gamma_h}\vec{r}\,\big|_{S_k} = \frac{1}{h_k}[G_1+G_2+G_3]\otimes\vec{t}_k. \tag{C.40}$$

First, we focus on G_1. Factoring out the denominator and rearranging gives

$$G_1 = \frac{1}{1+\vec{n}^0_{k+1}\cdot\vec{n}^1_k}[\vec{n}^1_k - \hat{\vec{n}}_{k+1} + \vec{n}^0_{k+1} - (\vec{n}^0_{k+1}\cdot\vec{n}^1_k)\hat{\vec{n}}_{k+1}],$$

and adding and subtracting $\hat{\vec{n}}_{k+1}$ gives

$$G_1 = \frac{1}{1+\vec{n}^0_{k+1}\cdot\vec{n}^1_k}\{(\vec{n}^1_k - \hat{\vec{n}}_{k+1}) + (\vec{n}^0_{k+1} - \hat{\vec{n}}_{k+1}) + [1-(\vec{n}^0_{k+1}\cdot\vec{n}^1_k)]\hat{\vec{n}}_{k+1}\}.$$

Another manipulation gives

$$\begin{aligned}G_1 = \frac{1}{1+\vec{n}^0_{k+1}\cdot\vec{n}^1_k}&\{(\vec{n}^1_k - \hat{\vec{n}}_{k+1}) + (\vec{n}^0_{k+1} - \hat{\vec{n}}_{k+1}) + \\ &+[\vec{n}^1_k\cdot(\vec{n}^1_k - \vec{n}^0_{k+1})]\hat{\vec{n}}_{k+1}\},\end{aligned} \tag{C.41}$$

because $\vec{n}_k^1 \cdot \vec{n}_k^1 = 1$. Next, we bound $|G_1|$:

$$\begin{aligned}
|G_1| &\leq \frac{|\vec{n}_k^1 - \hat{\vec{n}}_{k+1}| + |\vec{n}_{k+1}^0 - \hat{\vec{n}}_{k+1}| + |\vec{n}_k^1| \, |\vec{n}_k^1 - \vec{n}_{k+1}^0| \, |\hat{\vec{n}}_{k+1}|}{|1 + \vec{n}_{k+1}^0 \cdot \vec{n}_k^1|}, \\
&= \frac{|\vec{n}_k^1 - \hat{\vec{n}}_{k+1}| + |\vec{n}_{k+1}^0 - \hat{\vec{n}}_{k+1}| + |\vec{n}_k^1 - \hat{\vec{n}}_{k+1} + \hat{\vec{n}}_{k+1} - \vec{n}_{k+1}^0|}{|1 + \vec{n}_{k+1}^0 \cdot \vec{n}_k^1|}, \\
&\leq 2 \frac{|\vec{n}_k^1 - \hat{\vec{n}}_{k+1}| + |\vec{n}_{k+1}^0 - \hat{\vec{n}}_{k+1}|}{|1 + \vec{n}_{k+1}^0 \cdot \vec{n}_k^1|},
\end{aligned}$$

and using (C.26) gives

$$|G_1| \leq 2 \frac{C_1 h_k \kappa_k + C_2 h_{k+1} \kappa_{k+1}}{|1 + \vec{n}_{k+1}^0 \cdot \vec{n}_k^1|}. \tag{C.42}$$

Assuming that Γ_h is sufficiently refined so that $\vec{n}_{k+1}^0 \cdot \vec{n}_k^1 > 0$ and taking the maximum curvature over all of $\hat{\Gamma}$ (i.e. $\kappa_{\hat{\Gamma}}$), (C.42) becomes

$$|G_1| \leq C_3 (h_k + h_{k+1}) \kappa_{\hat{\Gamma}}. \tag{C.43}$$

The term G_3 satisfies a similar bound with the index k shifted by 1:

$$|G_3| \leq C_4 (h_{k-1} + h_k) \kappa_{\hat{\Gamma}}. \tag{C.44}$$

The other term G_2 is bounded using (C.29):

$$|G_2| \leq C_5 h_k \kappa_k \leq C_5 h_k \kappa_{\hat{\Gamma}}. \tag{C.45}$$

Going back to (C.40) and bounding, we get

$$\begin{aligned}
|\nabla_{\Gamma_h} \vec{r}| \,\big|_{S_k} &\leq \frac{1}{h_k} [|G_1| + |G_2| + |G_3|], \\
&\leq \frac{C_6}{h_k} [h_{k+1} + 3h_k + h_{k-1}] \kappa_{\hat{\Gamma}},
\end{aligned}$$

and using the shape regularity (C.33) gives

$$|\nabla_{\Gamma_h} \vec{r}| \,\big|_{S_k} \leq \frac{C_6}{h_k} \left[b_2 h_k + 3h_k + \frac{1}{b_1} h_k \right] \kappa_{\hat{\Gamma}} \leq C_r \kappa_{\hat{\Gamma}},$$

which is true for all S_k. Hence, we get the assertion (C.34). □

Finally, we can prove that $|\nabla_{\Gamma_h}\vec{n}_s|$ is bounded uniformly by a constant. Because $\vec{n}_s$ is continuous and piecewise C^1, we have that

$$\|\nabla_{\Gamma_h}\vec{n}_s\|_{\infty,\Gamma_h} = \max_{1\leq k\leq N_\mathsf{S}} \|\nabla_{\Gamma_h}\vec{n}_s\|_{\infty,S_k}.$$

So, we compute $\nabla_{\Gamma_h}\vec{n}_s$ on S_k for some k. By (C.11) and the product and chain rule, we have

$$\nabla_{\Gamma_h}\vec{n}_s\big|_{S_k} = -(\vec{r}\cdot\vec{n})^{-2}[\vec{n}\cdot\nabla_{\Gamma_h}\vec{r} + \vec{r}\cdot\nabla_{\Gamma_h}\vec{n}]\otimes\vec{r} + (\vec{r}\cdot\vec{n})^{-1}\nabla_{\Gamma_h}\vec{r}\big|_{S_k}.$$

As long as the curved edges of Γ_h are not severely distorted (i.e. $\vec{r}\cdot\vec{n} > c > 0$), we get the following bound:

$$|\nabla_{\Gamma_h}\vec{n}_s|\big|_{S_k} \leq c^{-2}[|\vec{n}|\,|\nabla_{\Gamma_h}\vec{r}| + |\vec{r}|\,|\nabla_{\Gamma_h}\vec{n}|] + c^{-1}|\nabla_{\Gamma_h}\vec{r}|\big|_{S_k}.$$

Since $\vec{r}$ is bounded in $L^\infty(\Gamma_h)$, we get

$$\begin{aligned}|\nabla_{\Gamma_h}\vec{n}_s|\big|_{S_k} &\leq \tilde{C}[|\nabla_{\Gamma_h}\vec{r}| + |\nabla_{\Gamma_h}\vec{n}|]\big|_{S_k},\\ &\leq \tilde{C}[C_r\kappa_{\hat{\Gamma}} + C_n\kappa_k]\big|_{S_k} \leq C_s\kappa_{\hat{\Gamma}},\end{aligned}$$

where we used (C.34) and (C.30) (note: $C_s > 0$ is independent of h and k). Hence,

$$\|\nabla_{\Gamma_h}\vec{n}_s\|_{\infty,\Gamma_h} \leq C_s\kappa_{\hat{\Gamma}}. \tag{C.46}$$

This means the $H^1(\Gamma_h)$ semi-norm is bounded independently of h:

$$\|\nabla_{\Gamma_h}\vec{n}_s\|_{0,\Gamma_h} \leq C_s\kappa_{\hat{\Gamma}}|\Gamma_h|^{1/2} \leq 2C_s\kappa_{\hat{\Gamma}}|\hat{\Gamma}|^{1/2},$$

where we have conservatively assumed $|\Gamma_h| \leq 4|\hat{\Gamma}|$. Ergo, because the $L^2(\Gamma_h)$ norm of $\vec{n}_s$ is clearly bounded, we get the full $H^1(\Gamma_h)$ norm

$$\|\vec{n}_s\|_{1,\Gamma_h} \leq C_{\hat{\Gamma}}, \text{ for all } h, \tag{C.47}$$

where $C_{\hat{\Gamma}}$ only depends on the geometry of $\hat{\Gamma}$. Therefore, property (3) in Section C.1 is satisfied.

Appendix D

Symbol Definitions

Table D.1: Fluid Variable and Parameter Definitions (Part A)

Symbol	**Definition**
H	channel height (distance between top and bottom plates)
L_{Elec}	electrode side length
σ_{lg}	liquid-gas interface surface tension
c_{pin}	three-phase contact line pinning coefficient
P_{pin}	maximum 'pinning pressure'
ρ	fluid density
μ	fluid dynamic viscosity
K_{hys}	hysteresis constant
θ	contact angle of the droplet (through liquid)
t	time variable
$\vec{x}$	position coordinate
Ω	set of points that make up the 2-D droplet domain
Γ	$\Gamma := \partial\Omega$ (1-D boundary of droplet)
κ_{xy}	curvature of Γ
κ_z	curvature of the liquid-gas interface along channel spacing

Table D.2: Fluid Variable and Parameter Definitions (Part B)

Symbol	**Definition**
$\vec{u}$	vector velocity
(u, v)	(x,y) velocity components $\vec{u} = (u, v)$
p	droplet pressure in Ω
λ	pinning pressure on Γ
$\vec{n}$	outward normal vector
L	length scale
U_0	velocity scale
t_0	time scale
Re	Reynolds number
Ca	capillary number
α	velocity time-derivative coefficient
β	velocity coefficient
∇	gradient operator
Δ	Laplace operator

Table D.3: Variational and FEM Definitions

Symbol	**Definition**
T_Ω	set of triangles for domain Ω
S_Γ	set of sides for boundary Γ
N_S	number of sides in S_Γ
h	mesh size
$\triangle t$	time-step size
γ	augmented lagrangian coefficient
$\vec{n}$	outward normal vector
$\vec{n}_s$	continuous approximation of outward normal vector
$a(\cdot,\cdot)$, $b(\cdot,\cdot)$	bilinear forms of mixed formulation
$\chi(\cdot)$	linear form for given data
$\mathbb{V}$	velocity function space (with norm $\|\cdot\|_{\mathbb{V}}$)
$\mathbb{P}$	pressure function space (with norm $\|\cdot\|_{\mathbb{P}}$)
$\mathbb{M}$	pinning variable function space (with norm $\|\cdot\|_{\mathbb{M}}$)
Λ	convex set contained in $\mathbb{M}$ (used with pinning variable)
a.e.	almost everywhere (in the sense of measure theory)
$\|\cdot\|_{q,D}$	$H^q(D)$ norm
$\mathcal{P}_k$	set of polynomials of order k
V_k	continuous vector polynomials of order k
Q_k	continuous scalar polynomials of order k
M_k	discontinuous scalar polynomials of order k
$\mathbb{V}_h, \mathbb{P}_h, \mathbb{M}_h$	discrete, conforming approximations of $\mathbb{V}, \mathbb{P}, \mathbb{M}$
Λ_h	discrete, convex set contained in $\mathbb{M}_h$

Table D.4: Particle Control Definitions

Symbol	**Definition**
$\vec{x}$	position of a particle
$\dot{\vec{x}}$	velocity of a particle
$\nabla P_k(\vec{x}_j)$	pressure gradient vector at the j^{th} particle due to the k^{th} electrode
G	matrix of pressure gradient values at particle positions
ξ	vector of boundary condition values to set for desired particle motion
ξ_{min}, ξ_{max}	minimum and maximum boundary condition values that may be set
b	vector of desired forcing directions for the particles

Table D.5: Differential Geometry Definitions

Symbol	**Definition**
Γ	surface (1-D or 2-D)
$\vec{X}$	surface parameterization (of Γ)
$\vec{X}^{-1}$	inverse map of $\vec{X}$
$\{X_k\}_{k=1}^3$	$\vec{X} = (X_1, X_2, X_3)$ (when Γ is in a 3-D space)
s	parameterization variable for 1-D surface
I	local chart for 1-D parameterization
s_1, s_2	parameterization variables for 2-D surface
$\{U_i\}$	local charts for 2-D parameterization
ω	scalar defined on Γ
$\tilde{\omega}$	scalar in local coordinates ($\tilde{\omega} = \omega \circ \vec{X}$)
$\vec{\varphi}$	vector field defined on Γ
$\tilde{\vec{\varphi}}$	vector in local coordinates ($\tilde{\vec{\varphi}} = \vec{\varphi} \circ \vec{X}$)
$\vec{t}$	'right-handed' tangent vector
$\vec{n}$	outward normal vector
∂_s	partial derivative operator with respect to param. variable s
∇_Γ	surface (tangential) gradient (on Γ)
Δ_Γ	surface Laplacian or Laplace-Beltrami operator ($\Delta_\Gamma := \nabla_\Gamma \cdot \nabla_\Gamma$)
g	first fundamental form (matrix)
g_{ij}	(i, j) component of g
g^{ij}	(i, j) component of g^{-1}
h	second fundamental form (matrix)
h_{ij}	(i, j) component of h
δ^i_j	Kronecker delta
κ_1, κ_2	principal curvatures
κ	total curvature $\kappa = \kappa_1 + \kappa_2$
κ_G	Gaussian curvature $\kappa_G = \kappa_1 \kappa_2$

BIBLIOGRAPHY

[1] R. A. Adams and J. J. F. Fournier. *Sobolev Spaces*, volume 140 of *Pure and Applied Mathematics Series*. Elsevier, 2nd edition, 2003.

[2] A. Anderson, X. Zheng, and V. Cristini. Adaptive unstructured volume remeshing - i: The method. *Journal of Computational Physics*, 208:616, 2005.

[3] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual Review Fluid Mechanics*, 30:139, 1998.

[4] M. Armani, S. Chaudhary, R. Probst, and B. Shapiro. Using feedback control and micro-fluidics to steer individual particles. In *18th IEEE International Conference on Micro Electro Mechanical Systems.*, Miami, Florida, 2005.

[5] M. Armani, S. Chaudhary, R. Probst, and B. Shapiro. Using feedback control of micro-flows to independently steer multiple particles. *Journal of Microelectromechanical Systems*, 15(4):945–956, Aug. 2006.

[6] A. Ashkin. History of optical trapping and manipulation of small-neutral particles, atoms, and molecules. *IEEE Journal on Selected Topics in Quantum Electronics*, 6(6):841–856, 2000.

[7] T. Aubin. *Nonlinear Analysis on Manifolds. Monge-Ampere Equations.* A Series of Comprehensive Studies in Mathematics. Springer, 1982.

[8] E. Baensch. Finite element discretization of the navier-stokes equations with a free capillary surface. *Numerische Mathematik*, 88:203–235, 2001.

[9] E. Baensch, P. Morin, and R. H. Nochetto. A finite element method for surface diffusion: the parametric case. *Journal of Computational Physics*, 203:321–343, 2005.

[10] V. Bahadur and S. V. Garimella. An energy-based model for electrowetting-induced droplet actuation. *Journal of Micromechanics and Microengineering*, 16(8):1494–1503, 2006.

[11] G. K. Batchelor. *An Introduction to Fluid Dynamics.* Cambridge University Press, New York, NY, 1967.

[12] J. T. Beale, T. Y. Hou, and J. S. Lowengrub. On the well-posedness of two fluid interfacial flows with surface tension. *Singularities in fluids, plasmas and optics (Heraklion, 1992), NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, 404:11–38, 1993.

[13] A. S. Bedekar, J. W. Jenkins, and S. Sundaram. A computational model for the design of electrowetting on dielectric (ewod) systems. In *NSTI Nanotechnology Conference & Trade Show*, Anaheim, California, U.S.A., 2005.

[14] B. Berge. Électrocapillarité et mouillage de films isolants par l'eau (including an english translation). *Comptes Rendus de l'Académie des Sciences de Paris, Série II*, 317:157–163, 1993.

[15] B. Berge and J. Peseux. Variable focal lens controlled by an external voltage: An application of electrowetting. *European Physical Journal E*, 3(2):159–163, 2000.

[16] T. D. Blake. The physics of moving wetting lines. *Journal of Colloid and Interface Science*, 299:1–13, 2006.

[17] T. D. Blake, A. Clarke, and E. H. Stattersfield. An investigation of electrostatic assist in dynamic wetting. *Langmuir*, 16(6):2928–2935, 2000.

[18] D. Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, 2nd edition, 2001.

[19] K. A. Brakke. Surface evolver, version 2.26c. Technical report, Susquehanna University, September 2005.

[20] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 2nd edition, 2002.

[21] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, New York, NY, 1991.

[22] F. Brezzi, W. W. Hager, and P. A. Raviart. Error estimates for the finite element solution of variational inequalities: Part ii. mixed methods. *Numerische Mathematik*, 31:1–16, 1978.

[23] R. S. Burdon. *Surface Tension and the Spreading of Liquids*. Cambridge University Press, Cambridge, Great Britain, 1949.

[24] R. Caiden, R. Fedkiw, and C. Anderson. Numerical method for two phase flow consisting of separate compressible and incompressible regions. *Journal of Computational Physics*, 166:1–27, 2001.

[25] F. Cattaneo, K. Baldwin, S. Yang, T. Krupenkine, S. Ramachandran, and J. A. Rogers. Digitally tunable microfluidic optical fiber devices. *Journal of Microelectromechanical Systems*, 12(6):907–912, 2003.

[26] C. H. A. Cheng, D. Coutand, and S. Shkoller. Navier-stokes equations interacting with a nonlinear elastic shell. *preprint*, page 60, 2006.

[27] S. K. Cho and C.-J. Kim. Particle separation and concentration control for digital microfluidic systems. In *The 16th Annual IEEE International Conference on MEMS*, pages 686–689, Kyoto, Japan, Jan. 2003.

[28] S. K. Cho, H. Moon, J. Fowler, S.-K. Fan, and C.-J. Kim. Splitting a liquid droplet for electrowetting-based microfluidics. In *International Mechanical Engineering Congress and Exposition*, New York, NY, 2001.

[29] S. K. Cho, H. Moon, and C.-J. Kim. Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits. *Journal of Microelectromechanical Systems*, 12(1):70–80, 2003.

[30] P. Clément. Approximation by finite element functions using local regularization. *R.A.I.R.O. Analyse Numérique*, 9:77–84, 1975.

[31] V. Cristini, J. Blawzdziewicz, and M. Loewenberg. An adaptive mesh algorithm for evolving surfaces: simulation of drop breakup and coalescence. *Journal of Computational Physics*, 168:445, 2001.

[32] J. E. Curtis, B. Koss, and D. G. Grier. Dynamic holographic optical tweezers. *Optics Communications*, 207:169–175, 2002.

[33] A. A. Darhuber, J. M. Davis, S. M. Troian, and W. W. Reisner. Thermocapillary actuation of liquid flow on chemically patterned surfaces. *Physics of Fluids*, 15(5):10150–10153, 2003.

[34] A. A. Darhuber and S. M. Troian. Principles of microfluidic actuation by modulation of surface stresses. *Annual Review of Fluid Mechanics*, 37:425–455, 2005.

[35] P.-G. de Gennes, F. Brochard-Wyart, and D. Quéré. *Capillarity and Wetting Phenomena: Drops, Bubbles, Pearls, Waves*. Springer-Verlag, New York, NY, 2004.

[36] C. Decamps and J. D. Coninck. Dynamics of spontaneous spreading under electrowetting conditions. *Langmuir*, 16(26):10150–10153, 2000.

[37] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, Upper Saddle River, New Jersey, 1976.

[38] G. Doğan. *A variational shape optimization framework for image segmentation*. PhD thesis, University of Maryland at College Park, 2006.

[39] E. J. Drelich, J. S. Laskowski, and K. L. Mittal. *Apparent and Microscopic Contact Angles*. VSP BV, Zeist, The Netherlands, 2000.

[40] Q. Du, C. Liu, and X. Wang. A phase field approach in the numerical study of the elastic bending energy for vesicle membranes. *Journal of Computational Physics*, 198:450, 2004.

[41] E. B. Dussan. On the ability of drops or bubbles to stick to non-horizontal surfaces of solids .2. small drops or bubbles having contact angles of arbitrary size. *Journal of Fluid Mechanics*, 151:1–20, Feb. 1985.

[42] G. Duvaut and J. L. Lions. *Inequalities in Mechanics and Physics*. Springer, 1976.

[43] G. Dziuk. An algorithm for evolutionary surfaces. *Numerische Mathematik*, 58:603–611, 1991.

[44] D. Enright, R. P. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183:83–116, 2002.

[45] J. M. Escobar, G. Montero, R. Montenegro, and Rodríguez. An algebraic method for smoothing surface triangulations on a local parametric space. *International Journal for Numerical Methods in Engineering*, 66(4):740–760, 2006.

[46] G. Evans, J. Blackledge, and P. Yardley. *Numerical Methods for Partial Differential Equations*. Springer-Verlag, London, UK, 2000.

[47] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, Providence, Rhode Island, 1998.

[48] R. B. Fair, V. Srinivasan, H. Ren, P. Paik, V. K. Pamula, and M. G. Pollack. Electrowetting-based on-chip sample processing for integrated microfluidics. In *IEEE International Electron Devices Meeting (IEDM)*, 2003.

[49] P. Fast and M. J. Shelley. A moving overset grid method for interface dynamics applied to non-newtonian hele-shaw flow. *Journal of Computational Physics*, 195:117–142, 2004.

[50] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory eulerian approach to interfaces in multi-material flows (the ghost fluid method). *Journal of Computational Physics*, 152:457–492, 1999.

[51] D. Field. Laplacian smoothing and delaunay triangulations. *Communications in Applied Numerical Methods*, 4:709–712, 1988.

[52] J. M. Floryan and H. Rasmussen. Numerical methods for viscous flows with moving boundaries. *Applied Mechanics Review*, 42(12):323–340, 1989.

[53] N. Fortner, B. Shapiro, and A. Hightower. Modeling and passive control of channel filling for micro-fluidic networks with thousands of channels. In *American Institute of Aeronautics and Astronautics (AIAA) 33rd AIAA Fluid Dynamics Conference and Exhibit*, Orlando, Florida, June 23–26 2003.

[54] E. R. Fossum. Cmos image sensors: Electronic camera-on-a-chip. *IEEE Transactions On Electronic Devices*, 44(10):1689–1698, 1997.

[55] J. Fowler, H. Moon, and C. J. Kim. Enhancement of mixing by droplet-based microfluidics. In *Proceedings of the IEEE Conference on MEMS*, pages 97–100, Las Vegas, NV, 2002.

[56] L. Freitag. On combining laplacian and optimization-based mesh smoothing techniques. *Trends in Unstructured Mesh Generation, ASME*, 220:37–44, 1997.

[57] L. Gaul, M. Kögl, and M. Wagner. *Boundary Element Methods for Engineers and Scientists (Hardcover)*. Springer, 1st edition, April 2003.

[58] D. Gilbarg and N. S. Trudinger. *Elliptic Partial Differential Equations of Second Order (Paperback, Reprint of 1998 edition)*. Springer, 2001.

[59] V. Girault and P. A. Raviart. *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*. Springer-Verlag, Berlin, 1986.

[60] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems*. Springer-Verlag, New York, NY, 1984.

[61] R. E. Goldstein, A. I. Pesci, and M. J. Shelley. Instabilities and singularities in hele-shaw flow. *Physics of Fluids*, 10(11):2701–2723, 1998.

[62] J. Gong, S. K. Fan, and C. J. Kim. Portable digital microfluidics platform with active but disposable lab-on-chip. In *Proc. IEEE Conf MEMS*, pages 355–358, Maastricht, The Netherlands, Jan. 2004.

[63] M. Haataja, J. A. Nieminen, and T. Ala-Nissila. Molecular ordering of precursor films during spreading of tiny liquid droplets. *Physical Review E*, 52(3):R2165–R2167, Sep 1995.

[64] F. Ham and Y.-N. Young. A cartesian adaptive level set method for two-phase flows. Technical report, Center for Turbulence Research, Annual Research Briefs, 2003.

[65] R. A. Hayes and B. J. Feenstra. Video-speed electronic paper based on electrowetting. *Nature*, 425(6956):383–385, 2003.

[66] B. He and J. Lee. Dynamic wettability switching by surface roughness effect. In *The 16th Annual IEEE International Conference on MEMS*, pages 120–123, Kyoto, Japan, Jan. 2003.

[67] J. Heikenfeld and A. J. Steckl. High-transmission electrowetting light valves. *Appl. Phys. Lett.*, 86:151121, April 2005.

[68] J. Heikenfeld and A. J. Steckl. Intense switchable fluorescence in light wave coupled electrowetting devices. *Appl. Phys. Lett.*, 86(1):011105, 2005.

[69] A. Henning, U. Staudinger, K.-J. Eichhorn, K. Sahre, M. Rogalli, A. W. Neumann, and K. Grundke. Contact angle hysteresis: study by dynamic cycling contact angle measurements and variable angle spectroscopic ellipsometry on polyimide. *Langmuir*, 20:6685–6691, 2004.

[70] E. J. Hinch and F. Plouraboue. Kelvin-helmholtz instability in a hele-shaw cell: Large effect from the small region near the meniscus. *Physics of Fluids*, 17, 2005.

[71] C. W. Hirt and B. D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981.

[72] J. P. Hoogenboom, D. L. J. Vossen, C. Faivre-Moskalenko, M. Dogterom, and A. van Blaaderen. Patterning surfaces with colloidal particles using optical tweezers. *Applied Physics Letters*, 80(25):4828–4830, June 2002.

[73] K. Hoshino, S. Triteyaprasert, K. Matsumoto, and I. Shimoyama. Electrowetting-based actuation for microinjection. In *The 12th International Conference on Solid-State Sensors, Actuators and Microsystems (Transducers)*, pages 1800–1803, Boston, Massachusetts, June 2003.

[74] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. Boundary integral methods for multicomponent fluids and multiphase materials. *Journal of Computational Physics*, 169:302–362, 2001.

[75] J. Hsieh, P. Mach, F. Cattaneo, S. Yang, T. Krupenkine, K. Baldwin, and J. A. Rogers. Tunable microfluidic optical-fiber devices based on electrowetting pumps and plastic microchannels. *IEEE Photonics Technology Letters*, 15(1):81–83, 2003.

[76] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 1st edition, 2000.

[77] T. B. Jones. On the relationship of dielectrophoresis and electrowetting. *Langmuir*, 18:4437–4443, 2002.

[78] G. E. Karniadakis and A. Beskok. *Micro Flows: Fundamentals and Simulation*. Springer-Verlag, New York, NY, 2001.

[79] C.-J. Kim. Micropumping by electrowetting. In *International Mechanical Engineering Congress and Exposition*, New York, NY, 2001.

[80] D. Kinderlehrer and G. Stampacchia. *An Introduction to Variational Inequalities and their Applications*. Academic Press, 1980.

[81] A. Klingner, J. Buehrle, and F. Mugele. Capillary bridges in electric fields. *Langmuir*, 20:6770–6777, 2004.

[82] A. Klingner and F. Mugele. Electrowetting-induced morphological transistions of fluid microstructures. *Journal of Applied Physics*, 95(5):2918–2920, 2004.

[83] S. Lang. *A First Course in Calculus.* Undergraduate Texts in Mathematics. Springer, 5th edition, 1986.

[84] S. Lang. *Real and Functional Analysis.* Graduate Texts in Mathematics. Springer, 3rd edition, 1993.

[85] S. Lang. *Undergraduate Analysis.* Springer, 2nd edition, 1997.

[86] S. Lang and G. Murrow. *Geometry.* Springer, 2nd edition, 1988.

[87] P. D. Lax. *Functional Analysis.* Wiley-Interscience, 2002.

[88] H. Lee, J. Lowengrub, and J. Goodman. Modeling pinchoff and reconnection in a hele-shaw cell. ii: Nonlinear evolution. *Physics of Fluids*, 14(2):514, 2002.

[89] J. Lee, H. Moon, J. Fowler, T. Schoellhammer, and C.-J. Kim. Electrowetting and electrowetting-on-dielectric for microscale liquid handling. In *Sensors and Actuators, A-Physics (95)*, page 269, 2002.

[90] F. L. Lewis and V. L. Syrmos. *Optimal Control.* Wiley-Interscience, New York, NY, 2nd edition, 1995.

[91] D. R. Lide, editor. *Handbook of Chemistry and Physics.* CRC Press, 82nd edition, 2002.

[92] J. Lienemann, A. Greiner, and J. G. Korvink. Modeling, simulation, and optimization of electrowetting. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(2):234–247, Feb. 2006.

[93] J. L. Lions and E. Magenes. *Non-Homogeneous Boundary Value Problems*, volume 1. Springer, 1972.

[94] G. Lippmann. Relations entre les phénomènes électriques et capillaires. *Ann. Chim. Phys.*, 5:494–549, 1875.

[95] J. López, J. Hernández, P. Gómez, and F. Faura. A volume of fluid method based on multidimensional advection and spline interface reconstruction. *Journal of Computational Physics*, 195(2):718–742, 2004.

[96] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. In *ACM Trans. Graph. (SIGGRAPH Proc.)*, pages 457–462, Los Angeles, 2004.

[97] H.-W. Lu, K. Glasner, A. L. Bertozzi, and C.-J. Kim. A diffuse interface model for electrowetting droplets in a hele-shaw cell. *Journal of Fluid Mechanics, submitted July*, 2005.

[98] J. A. Marsh, S. Garoff, and E. B. Dussan. Dynamic contact angles and hydrodynamics near a moving contact line. *Physical Review Letters*, 70(18):2778–2781, 1993.

[99] L. Minnema, H. A. Barneveld, and P. D. Rinkel. An investigation into the mechanism of water treeing in polyethylene high voltage cables. *IEEE Transactions on Electrical Insulators*, 15:461–472, 1980.

[100] K. Mohseni and A. Dolatabadi. An electrowetting microvalve: Numerical simulation. *Ann NY Acad Sci*, 1077:415–425, 2006.

[101] H. Moon, A. R. Wheeler, R. L. Garrell, J. A. Loo, and C.-J. Kim. On-chip sample preparation by electrowetting-on-dielectric digital microfluidcs for matrix assisted laser desorption/ionization mass spectrometry. In *Proceedings of IEEE MEMS*, pages 859–862, Miami, Florida, Feb 2005.

[102] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, New York, NY, 1994.

[103] F. Mugele and J.-C. Baret. Electrowetting: from basics to applications. *Journal of Physics: Condensed Matter*, 17:R705–R774, 2005.

[104] N. S. Nise. *Control Systems Engineering, 2nd Ed.* The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1995.

[105] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, New York, NY, 2003.

[106] E. J. F. Padday. *Wetting, Spreading, and Adhesion*. Academic Press, London, 1978.

[107] P. Paik, V. K. Pamula, and R. B. Fair. Rapid droplet mixers for digital microfluidic systems. *Lab on a Chip*, 3:253–259, 2003.

[108] V. K. Pamula, P. Paik, J. Venkatraman, M. G. Pollack, and R. B. Fair. Microfluidic electrowetting-based droplet mixing. In *Micro Electro Mechanical Systems Conference*, Berkeley, 2001.

[109] R. L. Panton. *Incompressible Flow, 2nd Edition.* John Wiley and Sons, Inc., New York, NY, 1996.

[110] E. G. Pétré and A. Sanfeld. *Lecture Notes in Physics: Capillarity Today.* Springer-Verlag, New York, NY, 1990.

[111] M. Pollack, R. Fair, and A. Shenderov. Electrowetting-based actuation of liquid droplets for microfluidic applications. *Applied Physics Letters*, 77(11), 2000.

[112] A. Ralston and P. Rabinowitz. *A First Course in Numerical Analysis, 2nd Edition.* Dover, Mineola, NY, 2001.

[113] H. Ren and R. B. Fair. Micro/nano liter droplet formation and dispensing by capacitance metering and electrowetting actuation. *Technical Digest IEEE-NANO*, pages 369–372, 2002.

[114] M. Renardy and R. C. Rogers. *An Introduction to Partial Differential Equations*, volume 13 of *Texts in Applied Mathematics.* Springer, 1993.

[115] T. Roques-Carmes, R. A. Hayes, B. J. Feenstra, and L. J. M. Schlangen. Liquid behavior inside a reflective display pixel based on electrowetting. *Journal of Applied Physics*, 95(8):4389–4396, 2004.

[116] R. E. Rusu. An algorithm for the elastic flow of surfaces. *Interfaces and Free Boundaries*, 7(3):229–239, 2005.

[117] P. Saavedra and L. R. Scott. Variational formulation of a model free-boundary problem. *Mathematics of Computation*, 57(196):451–475, October 1991.

[118] W. Satoh, M. Loughran, and H. Suzuki. Microfluidic transport based on direct electrowetting. *Journal of Applied Physics*, 96(1):835–841, 2004.

[119] E. Schaffer and P. Z. Wong. Dynamics of contact line pinning in capillary rise and fall. *Physical Review Letters*, 80(14):3069–3072, 1998.

[120] A. Schmidt. Computation of three dimensional dendrites with finite elements. *Journal of Computational Physics*, 125:293–312, 1996.

[121] A. Schmidt and K. G. Siebert. *Design of Adaptive Finite Element Software: The Finite Element Toolbox ALBERTA*. Springer, Heidelberg, Germany, 1st edition, 2005.

[122] L. R. Scott and S. Zhang. Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Mathematics of Computation*, 54(190):483–493, 1990.

[123] S. A. Sethian. *Level Set Methods and Fast Marching Methods, 2nd Edition.* Cambridge University Press, New York, NY, 1999.

[124] B. Shapiro, H. Moon, R. Garrell, and C.-J. Kim. Equilibrium behavior of sessile drops under surface tension, applied external fields, and material variations. *Journal of Applied Physics*, 93, 2003.

[125] M. J. Shelley, R. E. Goldstein, and A. I. Pesci. Topological transitions in hele-shaw flow. *Singularities in fluids, plasmas and optics (Heraklion, 1992), NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, 404:167–188, 1993.

[126] J. R. Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In M. C. Lin and D. Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.

[127] J. R. Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures. Paper Found at http://www.cs.cmu.edu/ jrs/jrspapers.html, 2002.

[128] K. G. Siebert. Iso-parametric elements (curved triangles) are required when computing with quadratic elements to maintain convergence order and accuracy. Private Communication.

[129] L. Slimane, A. Bendali, and P. Laborde. Mixed formulations for a class of variational inequalities. *Mathematical Modelling and Numerical Analysis*, 38(1):177 – 201, 2004.

[130] G. D. Smith. *Numerical Solution of Partial Differential Equations, Third Edition*. Oxford University Press, Oxford, UK, 1985.

[131] V. Srinivasan, V. K. Pamula, M. G. Pollack, and R. B. Fair. A digital microfluidic biosensor for multianalyte detection. In *IEEE 16th Annual International Conference on Micro Electro Mechanical Systems*, pages 327–330, 2003.

[132] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*, volume 12 of *Texts in Applied Mathematics*. Springer, 3rd edition, 2002.

[133] G. Strang. *Linear Algebra and Its Applications*. Brooks Cole, New York, NY, 3rd edition, 1988.

[134] Y. Sun and C. Beckermann. Diffuse interface modeling of two-phase flows based on averaging: mass and momentum equations. *Physica D*, 198:281, 2004.

[135] G. Supeene, C. R. Koch, and S. Bhattacharjee. Numerically modeled dynamic response of perfect and leaky dielectric droplets in an electric field. *icmens*, 0:99, 2005.

[136] M. Sussman and E. G. Puckett. A coupled level set and volume of fluid method for computing 3-d and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162:301, 2000.

[137] P. A. Thompson and M. O. Robbins. Simulations of contact-line motion: Slip and the dynamic contact angle. *Physical Review Letters*, 63(7):766–769, Aug 1989.

[138] M. Vallet, M. Vallade, and B. Berge. Limiting phenomena for the spreading of water on polymer films by electrowetting. *The European Physical Journal B*, 11:583–591, 1999.

[139] H. J. J. Verheijen and W. J. Prins. Reversible electrowetting and trapping of charge: Model and experiments. *Langmuir*, 15:6616–6620, 1999.

[140] S. W. Walker. A hybrid variational-level set approach to handle topological changes. Master's thesis, University of Maryland at College Park, May 2007.

[141] S. W. Walker and B. Shapiro. A control method for steering individual particles inside liquid droplets actuated by electrowetting. *Lab on a Chip*, 5:1404–1407, October 2005.

[142] S. W. Walker and B. Shapiro. Modeling the fluid dynamics of electrowetting on dielectric (ewod). *Journal of Microelectromechanical Systems*, 15(4):986–1000, August 2006.

[143] K.-L. Wang and T. B. Jones. Electrowetting dynamics of microfluidic actuation. *Langmuir*, 21(9):4211–4217, 2005.

[144] A. R. Wheeler, H. Moon, C.-J. Kim, J. A. Loo, and R. L. Garrell. Electrowetting-based microfluidics for analysis of peptides and proteins by

matrix-assisted laser desorption/ionization mass spectrometry. *Analytical Chemistry*, 76:4833–4838, 2004.

[145] T. J. Willmore. Total mean curvature squared of surfaces. *Geometry and topology of submanifolds, VIII (Brussels, 1995/Nordfjordeid, 1995),*, pages 383–391, 1996.

[146] E. Wolfram and R. Faust. *Wetting, Spreading and Adhesion*, chapter Liquid Drops on a Tilted Plate, Contact Angle Hysteresis and the Young Contact Angle. Academic Press, Leicestershire, England, 1978.

[147] P. Yue, J. J. Feng, C. Liu, and J. Shen. A diffuse-interface method for simulating two-phase flows of complex fluids. *Journal of Fluid Mechanics*, 515:293, 2004.

[148] E. C. Zachmanoglou and D. W. Thoe. *Introduction to Partial Differential Equations with Applications (Paperback)*. Dover Publications, 1987.

[149] X. Zheng, J. Lowengrub, A. Anderson, and V. Cristini. Adaptive unstructured volume remeshing - ii: Application to two- and three-dimensional level set simulations of multiphase flow. *Journal of Computational Physics*, 208:626, 2005.