Signorini conditions for inviscid fluids

by

Yu Gu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Computer Science

Waterloo, Ontario, Canada, 2021

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In this thesis, we present a new type of boundary condition for the simulation of inviscid fluids – the Signorini boundary condition. The new condition models the non-sticky contact of a fluid with other fluids or solids. Euler equations with Signorini boundary conditions are analyzed using variational inequalities. We derived the weak form of the PDEs, as well as an equivalent optimization based formulation. We proposed a finite element method to numerically solve the Signorini problems. Our method is based on a staggered grid and a level set representation of the fluid surfaces, which may be plugged into an existing fluid solver. We implemented our algorithm and tested it with some 2D fluid simulations. Our results show that the Signorini boundary condition successfully models some interesting contact behavior of fluids, such as the hydrophobic contact and the non-coalescence phenomenon.

Acknowledgements

I would like to thank my supervisor professor Christopher Batty. During my study at University of Waterloo, I was able to freely explore any idea that interests me and always get his support and helpful guidance. He was also very supportive when I made new career choices. The time at Waterloo has been enjoyable and rewarding. All of this would not have been possible without him.

I would like to also thank professor Justin Wan and professor Yuying Li for taking the time to read my proposal and thesis, and to be a committee member.

Table of Contents

List of Figures

List of Tables

Chapter 1

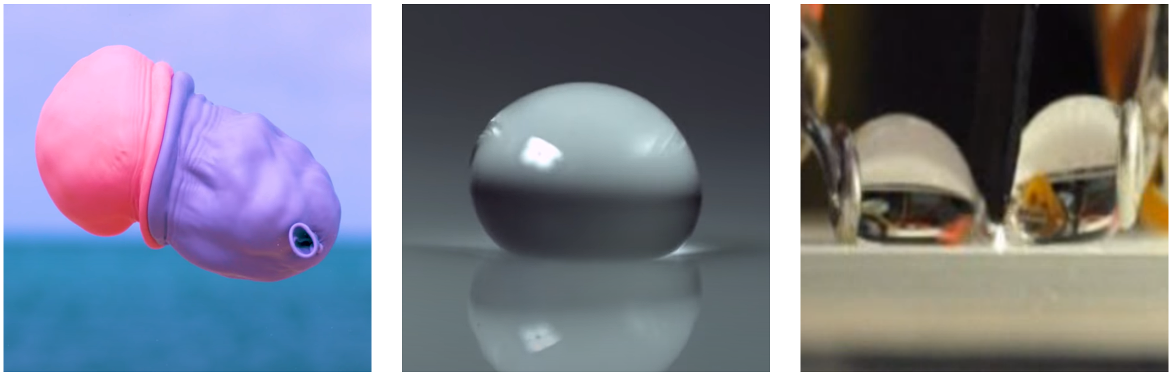
Introduction

Simulating fluids realistically has been a popular topic of computer graphics since the 1990s. Fluid mechanics, as a well studied subject, is able to explain a wide range of common fluid related phenomena. Computational fluid dynamics (CFD), based on fluid mechanical principles, provides qualitative approaches for analyzing and simulating fluids. Computer graphics researchers have taken advantage of these theories to produce high quality animation of fluids. In the past few decades, basic fluid animation techniques have been intensively studied, and researchers have begun to work on simulating many more exotic phenomena, such as ferrofluid [27] and bubble rings [45].

In this thesis, we introduce a new boundary condition for simulating inviscid fluids: the Signorini condition. In continuum mechanics, the Signorini condition models the unilateral contact between objects that disallows penetration and attraction. We apply the condition to inviscid fluids to model certain behavior of fluids. Figure 1.1 illustrates some potential applications of the new condition:

Contact of fluids through a membrane. Consider a frictionless balloon filled with water, where the mass of the balloon is much smaller than the water it contains. The liquid inside the balloon follows the Euler equations, and the balloon creates strong tension on the surface. When the balloon contacts other solid objects (e.g., another balloon), the observed contact behavior is that of two solids. Therefore it make senses to model the overall motion by Euler equations with a solid-like boundary condition. In practice, we do not consider or implement this physical system, but it provides a useful and intuitive analogy for the phenomena to be considered.

Non-coalescing fluids. This phenomenon is usually expected when two different types of *immiscible* fluid, such as water and oil, collide with each other. Each fluid would



(a) Two water-filled balloons collide.

(b) A droplet lands on a water surface and bounces off.

(c) A droplet is cut by a hydrophobic knife.

Figure 1.1: Various fluid contact phenomena.

retain its clear boundary with the other fluid, and under forces (e.g., surface tension), the two fluids may separate. The possibility of non-coalescence between two fluid bodies of the *same* type may appear counterintuitive at first, but it occurs very often even for common fluids like water. It is observed in dripping coffee, falling raindrops, floating oil droplets, lava lamps, etc.

The bouncing water droplet is a good example of non-coalescence of fluids. When a water droplet (with diameter $\approx$ 5mm) falls onto a water pool, often it will not merge with the water pool immediately. Instead, it collides against the surface of the water pool and then bounces off, as if they were two immiscible fluids.

Hydrophobic surfaces. Hydrophobic surfaces, like the surface of a lotus leaf, are very difficult to wet. The mechanism of hydrophobicity is complicated. One possible cause of hydrophobicity is that the microstructure on the surface prevents a substantial contact between the solid and the fluid.

A physically truthful simulation of the above phenomena requires resolving objects of contrasting scales at the same time. For example, the thickness of a balloon is around 10^{-3} of its diameter; for the bouncing droplets, one of the dominant factors behind these phenomena is the "trapped fluid". In the bouncing droplet example, when the droplet approaches the underlying surface, it would squeeze the ambient air and finally trap a little bit of air in between the two fluid bodies. The two fluid bodies are not directly interacting with each other, but through a thin layer of air, the thickness of which is estimated to be $O(\mu\text{m})$ [55, 42]. That is also around 10^{-3} of the diameter of the droplet. The contrasting scales make the simulation very difficult. On the other hand, if we do not care much about the details of the balloon or the trapped air, we may make some reasonable assumptions and build models that approximate reality. Using simplified models to encapsulate low-level details is a common practice in mechanical simulation. For example, friction is a result of complicated microscopic interactions, but it is often well approximated by Coulomb's law. In this thesis, we will use Signorini conditions as a simplified model of the contact behavior, and propose numerical methods for simulating these interesting phenomena.

Chapter 2

Preliminaries

In this chapter, we will summarize the necessary preliminary knowledge for our work, and review some recent advances in related fields.

We first clarify our conventions for notations. In mathematical formulae, we use italic letters (e.g., a) to denote scalar constants or fields, bold letters (e.g., $\mathbf{a}$) to denote vector or vector fields and bold capital letters (e.g., $\mathbf{A}$) for matrices. We use $\equiv$ to indicate either a definition or an identity which holds regardless of the choice of an involving variable. Number subscripts (e.g., a_1, a_i) are used to identify variables for different fluid phases, and superscripts (e.g., $\mathbf{a}^n$) are used to indicate the number of time steps.

2.1 The basics of fluid simulation

In this thesis, we are concerned with the simulation of incompressible inviscid fluids, which are characterized by incompressible Euler equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{1}{\rho}(-\nabla p + \mathbf{f}), \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{2.2}$$

Let $\Omega \in \mathbb{R}^3$ be an open set representing the space occupied by the fluid. Equations (2.1) and (2.2) are the equations that hold in Ω. To solve these equations, boundary conditions must be specified on the boundary $\Gamma = \partial\Omega$. We use Γ with a subscript (e.g., Γ_D, Γ_N) to denotes a subset of Γ. Two most commonly used types of boundary conditions are:

- Dirichlet boundary condition, applied in areas where the pressure is known:
$$p = g, \quad \text{on } \Gamma_D. \tag{2.3}$$

- Neumann boundary condition, applied in areas where velocity in the direction of the surface normal $\mathbf{n}$ is known:
$$\mathbf{u} \cdot \mathbf{n} = u_n, \quad \text{on } \Gamma_N. \tag{2.4}$$

The advection-projection scheme is a popular split-step method to solve these PDEs. Bridson has a comprehensive discussion on this topic in his book [9]. Here is an outline of a single iteration of the method:

1. Advect the current velocity field in itself using semi-Lagrangian advection [52]:
$$\mathbf{u}'(\mathbf{x}) = \mathbf{u}^n(\mathbf{x} - \mathbf{u}^n(\mathbf{x})\Delta t). \tag{2.5}$$

2. Apply external forces to $\mathbf{u}'$ using Forward Euler w.r.t. t:
$$\mathbf{u}^* = \mathbf{u}' + \rho^{-1}\mathbf{f}\Delta t. \tag{2.6}$$

3. $\mathbf{u}^*$ is in general not divergence-free and does not satisfy the boundary conditions. We need to solve for a pressure field p that makes $\mathbf{u}^{n+1}$ divergence-free and satisfy the boundary conditions, where
$$\mathbf{u}^{n+1} = \mathbf{u}^* - \rho^{-1}\nabla p \Delta t. \tag{2.7}$$

4. Compute $\mathbf{u}^{n+1}$ with equation (2.7) and start a new cycle.

In the above procedure, steps 1, 2 and 4 are conceptually simple and computationally inexpensive, while step 3, also called the "projection" step, encapsulates most of the complexity. In the case of liquid simulation, the simulation domain Ω is constantly evolving. We need surface tracking mechanisms, which are often based on the level set method [2], in addition to these steps.

When multiple immiscible fluids are present, equations (2.1) and (2.2) still hold inside each of the fluids, but we will need a new equation to model how two fluids interact at their interface of contact Γ_I. For inviscid fluids, the free-slip condition is the most often used:
$$\mathbf{u}_1 \cdot \mathbf{n_1} + \mathbf{u}_2 \cdot \mathbf{n}_2 = 0, \text{ on } \Gamma_I. \tag{2.8}$$

This condition ensures that the volume is not gained or lost, while allowing the fluids to freely slide against each other. Surface tracking becomes more complicated in multi-phase liquid simulation, as we need to maintain a consistent representation for all liquids. Losasso et al. [35] addressed this challenge using multiple level sets and proposed modifications to the advection-projection scheme accordingly to support multi-phase simulation.

2.2 Related studies on numerical methods for Poisson's equation

Setting $\nabla \cdot \mathbf{u}^{n+1} = 0$, we get the following equation:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*. \tag{2.9}$$

It is an instance of the Poisson's equation, which is one of the simplest forms of elliptic PDEs. It has important application in many other fields, and it has been extensively studied. Generally, the equation has no analytical solution. The numerical algorithms to solve the PDE vary significantly depending on a number of factors, such as the types of boundary conditions, the geometry properties of the domain and the method of discretization.

The Marker-and-Cell (MAC) method [25] was invented to solve PDEs arising from incompressible flows in the 1960s. It is a finite difference method based on a staggered grid and it is widely adopted in the computer graphics community. Fedkiw et al. [18] proposed the Ghost Fluid Method (GFM) to better address free surface (Dirichlet) conditions. GFM has become popular since it adds no cost to the MAC method but significantly improves the quality of the simulation. For Neumann boundary conditions, Batty et al. proposed a minimization approach [6], which was later interpreted in a finite volume method context [43]. Both GFM and Batty's finite volume method exhibit second order accuracy in pressure [60, 43]. However, GFM is only for Dirichlet boundary conditions and Batty's finite volume method is only for Neumann boundary conditions, and they are not naturally compatible with each other when non-axis-aligned boundaries are present. Helgadottir et al. combined the two methods to give a second order accurate algorithm for problems that have mixed Dirichlet, Neumann and Robin boundary conditions [26].

Instead of relying entirely on a regular Cartesian grid, some other methods use non-Cartesian meshes to better capture irregular boundaries. Feldman et al. [19] combined regular grids and unstructured tetrahedral meshes for the simulation of gas inside an irregular domain. Chentanez et al. [13] extended the idea to support dynamic boundaries and free-surface liquid. Brochu et al. [10] discretize the simulation domain using Voronoi meshes, which captures more thin features than a plain grid. Guittet et al. [23] proposed the Voronoi interface method, which uses a hybrid mesh consisting of a Voronoi mesh near the boundary and a Cartesian grid for the interior. The Voronoi mesh is generated such that boundary Voronoi cell faces are aligned with the surface, thus allowing accurate treatment of Neumann conditions. The method of Brochu et al. correctly resolves thin gaps formed by two surfaces next to each other , but it is designed for single phase simulation, meaning that the method either treats both surfaces as free surfaces, or merges the two volumes so that the surfaces become the interior. The Voronoi interface method is for multiphase problems with submerged interfaces, which does not allow different phases to separate. For our purpose, we need a method that combines the merits of both. However, robustly generating Voronoi meshes for multiphase simulation is a non-trivial task. There has been dedicated research on such problems [1], but the code is proprietary and not readily available.

There are also methods based on unstructured meshes. Klingner et al. [29] used unstructured meshes to simulate gases. Some used finite element method (FEM) based approaches for simulating gases and liquids [37, 15]. The difficulty of applying FEM to fluid simulation mostly lies in geometry processing. Unstructured meshes enable comfortable treatment of both Dirichlet and Neumann boundary conditions, however, as pointed out by Clausen et al., obtaining high quality tetrahedral meshes and handling topological changes are tricky and computationally expensive [15].

All the methods mentioned above, despite having different mesh structures and discretization schemes, support problems involving only Dirichlet or Neumann boundary conditions. Studies on PDEs with Signorini boundary conditions will be reviewed in section 2.4.

2.3 Linear complementarity problems

A linear complementarity problem (LCP) refers to the following algebraic problem: given a square matrix $\mathbf{M}$ and a vector $\mathbf{q}$, we seek a pair of vectors $\mathbf{w}, \mathbf{z}$, s.t.

$$\mathbf{w} = \mathbf{M}\mathbf{z} + \mathbf{q}, \tag{2.10a}$$

$$\mathbf{z}, \mathbf{w} \in \mathbb{R}^{+^n}, \tag{2.10b}$$

$$\mathbf{z}^T\mathbf{w} = 0. \tag{2.10c}$$

The last constraint is called the *complementarity condition*. Note that equations (2.10b) and (2.10c) are equivalent to component-wise constraints

$$w_i, z_i \geq 0 \quad \text{and} \quad w_i z_i = 0, \quad i = 1, \ldots, n. \tag{2.11}$$

The following properties hold true if $\mathbf{M}$ is positive definite:

1. The LCP instance has a unique solution (vector pair $\mathbf{w}, \mathbf{z}$).
2. The solution $\mathbf{z}$ minimizes $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{M}\mathbf{x} + \mathbf{q}^T\mathbf{x}$ subject to $\mathbf{x} \in \mathbb{R}^{+^n}$.
3. The LCP can be equivalently formulated as: find $\mathbf{z}$, s.t. $\forall \mathbf{v} \in \mathbb{R}^{+^n}$:

$$(\mathbf{M}\mathbf{z} + \mathbf{q})^T\mathbf{z} = 0, \tag{2.12a}$$

$$(\mathbf{M}\mathbf{z} + \mathbf{q})^T\mathbf{v} \geq 0. \tag{2.12b}$$

The first property suggests the importance of having a positive definite linear system in our discretization. In many other discretization schemes positive definite systems are preferred because they may be solved efficiently with methods like the conjugate gradient method. For our problem, it is also a guarantee for solvability. The second and third properties are useful for making connections between the variational forms of the PDEs and their discretized counterparts, which we will further discuss in chapter 3. We refer readers to the textbook by Cottle [16] for basic theory of LCP.

A variant of LCP, called mixed LCP (MLCP), does not impose conditions (2.11) on all components, but a subset of all components $C \subset \{1, \ldots n\}$:

$$w_i, z_i \geq 0 \quad \text{and} \quad w_i z_i = 0, \quad \forall i \in C; \tag{2.13}$$

for other components, MLCP only requires

$$w_i = 0, \quad z_i \in \mathbb{R}, \quad \forall i \in \bar{C}. \tag{2.14}$$

When C is empty, the MLCP is reduced to the linear system $\mathbf{M}\mathbf{z} + \mathbf{q} = \mathbf{0}$. If $\mathbf{M}$ is positive definite, the MLCP is equivalent to the following quadratic minimization problem:

$$\min_{x_i \geq 0 | i \in C} \frac{1}{2}\mathbf{x}^T\mathbf{M}\mathbf{x} + \mathbf{q}^T\mathbf{x}. \tag{2.15}$$

2.4 Continuous unilateral contact

In contact mechanics, *unilateral contact* or *unilateral constraint* refers to a mechanical constraint that forbids both *penetration* and *attraction* between two rigid/deformable bodies. It effectively models how solid objects in our daily life behave when they are in contact. The Signorini condition was proposed to model such contact. The condition was proposed by Antonio Signorini, an Italian mathematician, in an elastostatic problem in linear elasticity [51]. Since the 1960s, elasticity problems with Signorini conditions have been studied extensively. A new mathematical tool, the *variational inequality*, was born to examine the theoretical aspects of PDEs with Signorini conditions [20]. In the engineering community, numerical methods were devised to directly tackle problems with Signorini conditions [28]. Unlike in computer graphics, they focused more on obtaining accurate solutions on a limited number of simple geometries.

In computer graphics, contact problems have been systematically studied for simulation of rigid bodies since the late 1980s [5]. For deformable bodies, most research in computer graphics treats contact between different bodies as an instantaneous collision. A post-impact status is calculated for the involved bodies after integrating over one time step. This formulation is simple and convenient, but intrinsically heuristic. Since the 2000s, researchers have been putting an increasing effort into accurately modelling contact using Signorini conditions [47, 3, 58].

Despite its significance in the field of contact dynamics, the Signorini condition is rarely mentioned in the CFD community. There are studies on numerically solving elliptic PDEs with Signorini boundary conditions (e.g., [62]). In computer graphics, Narain et al. [41] proposed the unilateral incompressibility constraint (UIC), which is a pressure-density complementarity condition, for the simulation of granular materials. Gerszewski and Bargteil [22] introduced UIC into fluid simulation to allow fluids to break up, creating splashy fluid motions. Despite having similar LCP formulations, these studies do not model a contact constraint. Batty et al. proposed a complementarity condition for eliminating "wall-sticking" artifacts [6]. Follow-up studies [13, 4, 30] proposed various efficient LCP solvers to tackle the scalability issue in the method of Batty et al. [6]. These studies are the closest to a serious treatment of Signorini condition in the CFD field. The potential of Signorini conditions in modelling more fluid dynamics phenomena remains largely unexplored.

Chapter 3

Weak form of Signorini problems

As discussed in section 2.1, a key part of the fluid simulation algorithm is the projection step, which aims to solve the following equations:

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p\right) = \frac{1}{\Delta t}\nabla \cdot \mathbf{u}^* \qquad \text{in } \Omega, \tag{3.1a}$$

$$p = 0 \qquad \text{on } \Gamma_D, \tag{3.1b}$$

$$\left(\mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p\right) \cdot \mathbf{n} = u_n \qquad \text{on } \Gamma_N. \tag{3.1c}$$

In the above equations, Ω denotes the interior of the fluid, Γ_D denotes the free surface, and Γ_N denotes the fluid-solid interface, where the normal component of the solid velocity is prescribed by u_n. In multiphase fluid simulations, we also have a free-slip condition for fluid-fluid contact:

$$\mathbf{u}_1 \cdot \mathbf{n_1} + \mathbf{u}_2 \cdot \mathbf{n}_2 = 0, \text{ on } \Gamma_I, \tag{3.2}$$

where $\mathbf{n}_i$ is the normal vector pointing away from fluid Ω_i and Γ_I is the boundary shared by two phases.

In this chapter, we will introduce a new type of boundary condition in addition to (3.1b) and (3.1c): the Signorini boundary condition. We discuss its generalization as a replacement of the free-slip condition (3.2) in multiphase fluid simulation.

3.1 Signorini boundary conditions

3.1.1 Motivation

At fluid-solid boundaries, condition (3.1c) only allows tangential slip. The velocity of the fluid and the solid must agree in the normal direction. Although the Neumann condition is physically sound, it may cause certain artifacts, e.g., inviscid fluid may appear to stick onto solid surfaces. This is expected on small scale, but usually the simulation is done in larger scale where this becomes undesirable from a visual perspective. Batty et al. suggested that imposing non-negativity constraints on the discrete pressure samples at the boundary may help resolve this artifact [6]. The linear system resulting from discretizing equation (3.1a) is then replaced by a mixed linear complementarity problem. Later, Chentanez and Müller-Fischer [14] proposed a multigrid solver for the MLCP, which was further improved by Lai et al. [30].

However, in all of the work above, there is no continuous characterization (in the form of PDEs) for the physics we aim to model, making these methods difficult to generalize to multiphase scenarios and other numerical methods. We will fill this gap by finding the correct continuous PDE characterization, and perform a variational analysis on the proposed PDEs. This allows us to formulate the problem as an optimization problem, and employ existing results from the finite element method.

We now introduce the Signorini boundary condition. For the simplicity of subsequent analysis, we assume that the Neumann boundary condition (equation (3.1c)) is replaced everywhere by the Signorini condition ($\Gamma_S = \Gamma_N$):

$$p \geq 0, \qquad \text{and} \tag{3.3a}$$

$$u_n - \left(\mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p\right) \cdot \mathbf{n} \geq 0, \qquad \text{and} \tag{3.3b}$$

$$p\left(u_n - \left(\mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p\right) \cdot \mathbf{n}\right) = 0 \qquad \text{on } \Gamma_S. \tag{3.3c}$$

When the right-hand side of (3.3a) is zero, we say the Signorini condition is *homogeneous*. Non-homogeneous conditions will be discussed in section 3.3. If a Neumann boundary condition is present alongside the Signorini boundary condition, it will go through the same transformations as condition (3.3b) and be absorbed into the linear form in the resulting weak form. This is also why Neumann boundary conditions are also called natural boundary conditions in FEM.

Note that $\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla p/\rho$. If we define a new notation $\perp^{\geq 0}$ as

$$a \perp^{\geq 0} b \iff (a, b \geq 0) \text{ and } (ab = 0), \tag{3.4}$$

the relations in (3.3) may be written compactly as

$$p \perp^{\geq 0} (u_n - \mathbf{u}^{n+1} \cdot \mathbf{n}) \quad \text{ on } \Gamma_S. \tag{3.5}$$

The intuition behind Signorini boundary conditions (3.3) is that the pressure of the fluid at the boundary should be no less than an ambient value, which prevents attraction by “negative pressure”. Moreover, it may exceed the ambient value only when the fluid is not leaving the solid. This is exactly the implication of “no penetration” and “no attraction” in the fluid dynamics context.

3.1.2 The weak form of homogeneous Signorini problem

In this section we analyze the elliptic equation (3.1a) with Dirichlet boundary condition (3.1b) and Signorini boundary conditions (3.3). Physically, these equations describe the behavior of a single liquid phase in contact with a solid.

Before diving into derivations, we would like to point out first that our analysis is *not* meant to be rigorous in the mathematical sense. We assume that all functions we are interested in are as smooth as desired, that all domains and their boundaries are regular, and that all weak solutions are also strong solutions. It is not in our interest nor within our expertise to undertake such formalities, but we do believe that our derivations provide useful insight into the problem, and by fixing the details along the steps we take, a solid foundation for our work may be established.

Let us start by defining the function space $K(\Omega)$ as

$$K(\Omega) = \{(f \geq 0 \text{ on } \Gamma_S) \text{ and } (f = 0 \text{ on } \Gamma_D) | f : \bar{\Omega} \to \mathbb{R}\}. \tag{3.6}$$

It is clear that the solution $p \in K(\Omega)$. Like in many other functional analyses (or finite element analyses), we first multiply both sides of equation (3.1a) by a trial function $v \in K(\Omega)$ and integrate the equation over Ω:

$$\int_\Omega \nabla \cdot \left(\frac{1}{\rho} \nabla p\right) v dx \equiv \int_\Omega \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* v dx. \tag{3.7}$$

Integrating by parts, we get

$$\int_{\partial\Omega} \frac{v}{\rho}\nabla p \cdot d\mathbf{s} - \int_{\Omega} \frac{1}{\rho}\nabla p \cdot \nabla v dx \equiv \frac{1}{\Delta t}\left(\int_{\partial\Omega} \mathbf{u}^* v \cdot d\mathbf{s} - \int_{\Omega} \mathbf{u}^* \cdot \nabla v dx\right). \tag{3.8}$$

Rearranging the terms:

$$\int_{\Omega} \rho^{-1}\nabla p \cdot \nabla v dx + \frac{1}{\Delta t}\int_{\Gamma_S} v\left(\mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p\right) \cdot d\mathbf{s} - \frac{1}{\Delta t}\int_{\Omega} \mathbf{u}^* \cdot \nabla v dx \equiv 0. \tag{3.9}$$

Since $v \geq 0$ on Γ_S, by inequality (3.3b) we have

$$\int_{\Gamma_S} v\left(\mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p\right) \cdot d\mathbf{s} \leq \int_{\Gamma_S} v u_n ds. \tag{3.10}$$

Combining (3.9) and (3.10), and letting

$$a(u, v) = \int_{\Omega} \rho^{-1}\nabla u \cdot \nabla v dx \tag{3.11}$$

and

$$b(v) = \frac{1}{\Delta t}\int_{\Gamma_S} v u_n ds - \frac{1}{\Delta t}\int_{\Omega} \mathbf{u}^* \cdot \nabla v dx, \tag{3.12}$$

we get:

$$a(p, v) + b(v) \geq 0, \quad \forall v \in K. \tag{3.13}$$

Next we will compute $a(p, p) + b(p)$. Note that everywhere on Γ_S, equality holds at least in one of the inequalities (3.3a) or (3.3b). Therefore we can split the the following integral:

$$\begin{aligned}\int_{\Gamma_S} p\left(\mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p\right) \cdot d\mathbf{s} &= \int_{\Gamma_S \cap \{x|p=0\}} 0 ds + \int_{\Gamma_S \cap \{x|p>0\}} p\left(\mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p\right) \cdot d\mathbf{s} \\ &= \int_{\Gamma_S \cap \{x|p=0\}} p u_n ds + \int_{\Gamma_S \cap \{x|p>0\}} p u_n ds \\ &= \int_{\Gamma_S} p u_n ds.\end{aligned} \tag{3.14}$$

Then by identity (3.9),

$$a(p, p) + b(p) = \int_{\Omega} \rho^{-1}\nabla p \cdot \nabla p dx + \frac{1}{\Delta t}\int_{\Gamma_S} p\left(\mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p\right) \cdot d\mathbf{s} - \frac{1}{\Delta t}\int_{\Omega} \mathbf{u}^* \cdot \nabla p dx = 0. \tag{3.15}$$

We are now ready to write down the weak form of the problem.

Definition 1 *(Weak form of PDEs with Signorini boundary conditions). For PDE (3.1a) with boundary conditions (3.1b) and (3.3), its weak form is defined as:*

Let K be defined as in (3.6), a, b be defined as in equations (3.11) and (3.12). We seek $p \in K(\Omega)$, s.t. $\forall v \in K(\Omega)$

$$a(p,p) + b(p) = 0, \tag{3.16a}$$

$$a(p,v) + b(v) \geq 0. \tag{3.16b}$$

Equations (3.16a),(3.16b) resemble the LCP formulation (2.12a),(2.12b). In fact, the latter may be seen as variational inequalities in the finite dimensional Euclidean space $\mathbb{R}^n$, in which $\{z \in \mathbb{R}^n | z_i \geq 0\}$ is an infinite cone. Likewise, $K(\Omega)$ is an infinite cone in the Sobolev space $H(\Omega)$. In (2.12a) and (2.12b), $\mathbf{M}$ is symmetric positive definite, and so is the bilinear form a here. There has been research on variational inequalities in the form of (3.16) in Banach spaces, which suggest a unique solution does exist for the problem [32].

3.2 Signorini jump conditions

In this section, we consider a multiphase scenario, and extend the Signorini boundary conditions to model the contact of two fluid bodies. We first write down the known equations for a two-phase projection. For $i = 1, 2$:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}_i^* \qquad \text{in } \Omega_i, \tag{3.17a}$$

$$p = 0 \qquad \text{on } \Gamma_{D,i} \tag{3.17b}$$

$$p \perp^{\geq 0} (u_{n,i} - \mathbf{u}_i^{n+1} \cdot \mathbf{n}_i) \qquad \text{on } \Gamma_{S,i}. \tag{3.17c}$$

$\Omega_1, \Omega_2, \Gamma_{D,1}, \Gamma_{D,2}, \Gamma_{S,1}, \Gamma_{S,2}, \Gamma_I$ are as illustrated in Figure 3.1. On Γ_I, we would like a different condition than the free-slip condition, which reads:

$$\mathbf{u}_1^{n+1} \cdot \mathbf{n}_1 + \mathbf{u}_2^{n+1} \cdot \mathbf{n}_2 = 0 \text{ on } \Gamma_I. \tag{3.18}$$

When there is surface tension, pressure is discontinuous across Γ_I. Therefore, we further need

$$p_1^- - p_2^- = J \text{ on } \Gamma_I, \tag{3.19}$$

where p_i^- is the one-sided limit of p taken from the side of Ω_i and J is value of the pressure jump. In this section, we assume $J = 0$, i.e., zero surface tension. The "no penetration, no attraction" idea in this context translates to

$$p \perp^{\geq 0} -(\mathbf{u}_1^{n+1} \cdot \mathbf{n}_1 + \mathbf{u}_2^{n+1} \cdot \mathbf{n}_2) \quad \text{on } \Gamma_I. \tag{3.20}$$

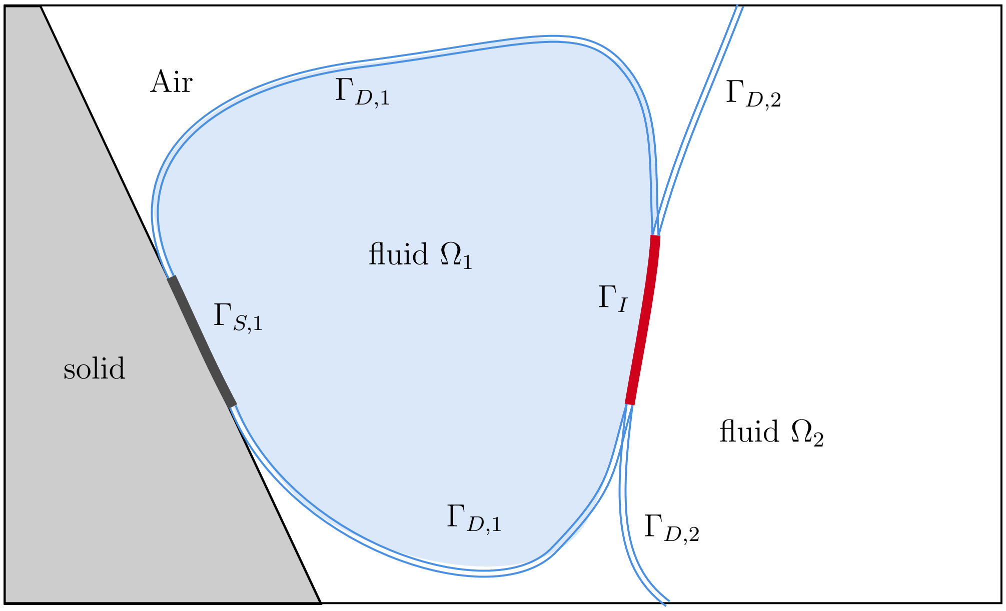


Figure 3.1: Two fluid bodies in contact.

Equations (3.17a)–(3.17c),(3.20) together give the complete PDE formulation, which we will refer to as the *Signorini Interface Problem (SIP)*. Following similar steps as in the last section, we may get the weak form of the SIP. A solution p to an SIP is a pair of continuous fields (p_1, p_2). We will omit the subscript when it is clear from the context which field is being referred to, e.g., $p = p_1$ in Ω_1.

Definition 2 *(Weak form of SIP) Let a and b be defined as*

$$a(u,v) = \sum_{i=1,2} \int_{\Omega_i} \rho^{-1} \nabla u_i \cdot \nabla v_i dx, \tag{3.21a}$$

$$b(v) = \sum_{i=1,2} \frac{1}{\Delta t} \int_{\Gamma_{S,i}} v_i u_{n,i} ds - \frac{1}{\Delta t} \int_{\Omega_i} \mathbf{u}_i^* \cdot \nabla v_i dx \tag{3.21b}$$

respectively, and

$$K_I = \{f = (f_1, f_2) \in K(\Omega_1) \times K(\Omega_2) | f_1 = f_2 \text{ on } \Gamma_I\}. \tag{3.22}$$

The weak form of PDEs (3.17a)–(3.17c) and (3.20) seeks to find $p \in K_I$ s.t. $\forall v \in K_I$:

$$a(p,p) + b(p) = 0, \tag{3.23a}$$

$$a(p,v) + b(v) \geq 0. \tag{3.23b}$$

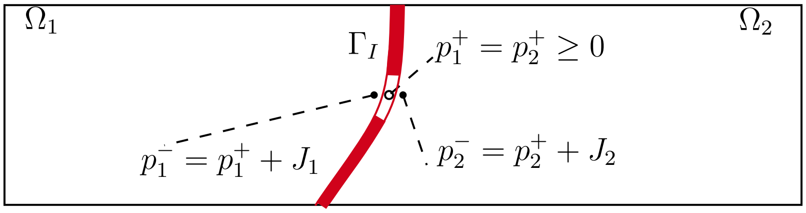


Figure 3.2: Pressure jump across the fluid-fluid surface.

3.3 Non-homogeneous Signorini conditions

In this section, we will look into non-homogeneous boundary/jump conditions, which is necessary when we take surface tension into account. Surface tension will incur a pressure jump $J = \sigma\kappa$ across the surface, where σ is the coefficient of surface tension and κ is the local mean curvature of the surface. That means if p^+ is the pressure on the outside the fluid surface, and p^- is the pressure on the inside of the fluid surface, their difference $p^- - p^+ = J$ (see figure 3.2).

Now let us revisit the Dirichlet conditions (3.1b), (3.17b), the Signorini boundary conditions (3.3), (3.17c) and the Signorini jump condition (3.20). The pressure p in these conditions represents p^+ though it is not specified. We may think of the fluids as being wrapped in balloons of zero thickness, and p^+ is the pressure on the outside of the balloon and p^- is the pressure on the inside of the balloon at the same point. p^+ is the pressure we should compare with the ambient value. On the fluid-fluid interface Γ_I, p_1^+ and p_2^+ should be greater than or equal to 0 to prevent attraction. If there are no forces between the fluids, we have $p_1^+ = p_2^+ = 0$. If there are forces, $p_1^+ = p_2^+ \geq 0$ by Newton's third law. In both cases, we have $p_1^+ = p_2^+$.

With this in mind, we may write down the non-homogeneous equations:

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p_i\right) = \frac{1}{\Delta t}\nabla \cdot \mathbf{u}_i^* \qquad \text{in } \Omega_i, \tag{3.24a}$$

$$p_i^+ = 0 \qquad \text{on } \Gamma_{D,i} \tag{3.24b}$$

$$p_i^+ \perp^{\geq 0} (u_{n,i} - \mathbf{u}_i^{n+1} \cdot \mathbf{n}_i) \qquad \text{on } \Gamma_{S,i}, \tag{3.24c}$$

$$p_i^+ \perp^{\geq 0} -(\mathbf{u}_1^{n+1} \cdot \mathbf{n}_1 + \mathbf{u}_2^{n+1} \cdot \mathbf{n}_2) \qquad \text{on } \Gamma_I, \tag{3.24d}$$

$$p^+ \equiv p_i^+ = p_i - J_i \qquad \text{on } \partial\Omega_i. \tag{3.24e}$$

In these equations, we may write $p^+ \equiv p_i^+$ without concerns of ambiguity, because p_1^+ and p_2^+ agree on the common interface Γ_I, and only one of them is defined on other parts of the boundary. $p_i^- = p_i$, since p_i is continuous in $\bar{\Omega}$.

The technique used in previous sections is no longer applicable, because the set L of all admissible $p = (p_1, p_2)$ is no longer a "cone" whose apex is at the origin. In other words, $\{\lambda f | \lambda \geq 0, f \in K_I\} = K_I$ (cf. (3.22)), but $\{\lambda f | \lambda \geq 0, f \in L\} \neq L$, which breaks the fundamental assumption in the analysis of variational inequalities. To apply our earlier results, we need to separate the non-homogeneous part from the problem.

To simplify our discussion, we first define a few linear operators:

$$\mathcal{L}(u) = \nabla \cdot \left(\frac{1}{\rho} \nabla u\right) \qquad \text{in } \Omega_1 \cup \Omega_2, \tag{3.25}$$

$$\mathcal{N}_i(u) = \left(\mathbf{u}_i^* - \frac{\Delta t}{\rho} \nabla u\right) \cdot \mathbf{n}_i \qquad \text{on } \partial\Omega_i, \tag{3.26}$$

$$\mathcal{V}(u) = \mathcal{N}_1(u_1) + \mathcal{N}_2(u_2) \qquad \text{on } \Gamma_I. \tag{3.27}$$

$\mathcal{L}$ is the operator on the left-hand side of equation (3.24a). $\mathcal{N}_i(u)$ gives the normal velocity of fluid i on its boundaries after updating $\mathbf{u}_i$ with pressure field u. $\mathcal{V}(u)$ gives the relative normal velocity, at which the two fluid bodies are approaching each other, on their common boundary Γ_I. To prevent penetration, $\mathcal{V}(u)$ should be zero or negative.

Next we write down three PDE systems, solving for $p, p^{\mathcal{D}}$ and $p^{\mathcal{S}}$ respectively:

$p = p^{\mathcal{D}} + p^{\mathcal{S}}$	$p^{\mathcal{D}}$	$p^{\mathcal{S}}$	in/on
$\mathcal{L}(p_i) = \dfrac{1}{\Delta t} \nabla \cdot \mathbf{u}_i^*$	$\mathcal{L}(p_i^{\mathcal{D}}) = 0$	$\mathcal{L}(p_i^{\mathcal{S}}) = \dfrac{1}{\Delta t} \nabla \cdot \mathbf{u}_i^*$	Ω_i
$p_i = J_i$	$p_i^{\mathcal{D}} = J_i$	$p_i^{\mathcal{S}} = 0$	$\Gamma_{D,i}$
$p_i \geq J_i$	$p_i^{\mathcal{D}} = J_i$	$p_i^{\mathcal{S}} \geq 0$	$\Gamma_{S,i}$
$u_{n,i} \geq \mathcal{N}_i(p_i)$		$u_{n,i} \geq \mathcal{N}_i(p_i^{\mathcal{D}} + p_i^{\mathcal{S}})$	$\Gamma_{S,i}$
$(p_i - J_i) \perp (u_{n,i} - \mathcal{N}_i(p_i))$		$p_i^{\mathcal{S}} \perp u_{n,i} - \mathcal{N}_i(p_i^{\mathcal{D}} + p_i^{\mathcal{S}})$	$\Gamma_{S,i}$
$p_i - J_i \geq 0$	$p_i^{\mathcal{D}} = J_i$	$p^{\mathcal{S}} \geq 0$	Γ_I
$0 \geq \mathcal{V}(p)$		$0 \geq \mathcal{V}(p^{\mathcal{D}} + p^{\mathcal{S}})$	Γ_I
$(p_i - J_i) \perp \mathcal{V}(p)$		$p^{\mathcal{S}} \perp \mathcal{V}(p^{\mathcal{D}} + p^{\mathcal{S}})$	Γ_I

Each column is a PDE system. The left column rewrites equations (3.24a)–(3.24e); the middle column represents two independent Poisson's equations with pure Dirichlet boundary conditions; the right column is an instance of homogeneous Signorini interface problem (cf. Definition 2), whose inputs are dependent on $p^{\mathcal{D}}$.

Equations for $p^{\mathcal{D}}$ and $p^{\mathcal{S}}$ have been shown solvable. By adding the corresponding equations, we can easily verify that $p^{\mathcal{D}}+p^{\mathcal{S}}$ is a solution for the non-homogeneous problem. This decomposition is a constructive proof that a solution exists for (3.24a)–(3.24e). We will use this decomposition to derive an optimization formulation for the problem, which eventually will not rely on this construction but instead solve the original PDEs directly.

3.3.1 An optimization perspective

In section 2.3, we mentioned that LCPs may be formulated as optimization problems. Signorini problems may be formulated as optimization problems, too. In the homogeneous Signorini boundary problem (cf. Definition 1) and homogeneous SIP (cf. Definition 2), the solution p minimizes

$$F(p) = \frac{1}{2}a(p,p) + b(p) \tag{3.28}$$

in K and K_I respectively.

For the non-homogeneous problem, we may write the solutions of its two subproblems as $p^{\mathcal{D}} = (p_1^{\mathcal{D}}, p_2^{\mathcal{D}})$ and $p^{\mathcal{S}} = (p_1^{\mathcal{S}}, p_2^{\mathcal{S}})$. $\forall u, v \in K_I$, let

$$a_i(u,v) = \int_{\Omega_i} \frac{1}{\rho} \nabla u_i \cdot \nabla v_i dx \tag{3.29}$$

and

$$b_i(v) = \frac{1}{\Delta t}\left(\int_{\Gamma_{S,i}} v_i u_{n,i} ds - \int_{\Omega_i} \mathbf{u}_i^* \cdot \nabla v_i dx\right), \tag{3.30}$$

then $p^{\mathcal{S}}$ minimizes the following functional s.t. $u \in K_I$:

$$\begin{aligned}
F^{\mathcal{S}}(u) &= \frac{1}{2}\sum_{i=1,2} a_i(u,u) + \sum_{i=1,2} b_i(u) + \sum_{i=1,2}\int_{\Gamma_{N,i}\cup\Gamma_{S,i}} \frac{u}{\rho}\nabla p_i^{\mathcal{D}} \cdot d\mathbf{s} \qquad \text{(eq. (3.28))} \\
&= \frac{1}{2}\sum_{i=1,2} a_i(u,u) + \sum_{i=1,2} b_i(u) + \sum_{i=1,2}\int_{\partial\Omega_i} \frac{u}{\rho}\nabla p_i^{\mathcal{D}} \cdot d\mathbf{s} \qquad (u = 0 \text{ on } \Gamma_{D,i}) \\
&= \frac{1}{2}\sum_{i=1,2} a_i(u,u) + \sum_{i=1,2} b_i(u) + \sum_{i=1,2}\left(\int_{\Omega_i} \frac{1}{\rho}\nabla p_i^{\mathcal{D}} \cdot \nabla u dx + \int_{\Omega_i} \nabla \cdot \left(\frac{1}{\rho}\nabla p^{\mathcal{D}}\right) u dx\right) \\
&= \frac{1}{2}\sum_{i=1,2} a_i(u,u) + \sum_{i=1,2} b_i(u) + \sum_{i=1,2} a_i(p_i^{\mathcal{D}}, u) + 0 \qquad (\text{since } \mathcal{L}(p^{\mathcal{D}}) = 0) \\
&= \sum_{i=1,2}\left(\frac{1}{2}a_i(u,u) + b_i(u) + a_i(p_i^{\mathcal{D}}, u)\right).
\end{aligned} \tag{3.31}$$

Since $p^{\mathcal{D}}$ is a constant field, we have

$$\begin{aligned}
p^{\mathcal{S}} &= \arg\min_{u\in K_I} F(u) \\
&= \arg\min_{u\in K_I} \sum_{i=1,2}\left(\frac{1}{2}a_i(u,u) + b_i(u) + a_i(p_i^{\mathcal{D}}, u) + \frac{1}{2}a_i(p_i^{\mathcal{D}}, p_i^{\mathcal{D}}) + b(p_i^{\mathcal{D}})\right) \\
&= \arg\min_{u\in K_I} \sum_{i=1,2}\left(\frac{1}{2}a_i(u + p_i^{\mathcal{D}}, u + p_i^{\mathcal{D}}) + b_i(u + p_i^{\mathcal{D}})\right) \\
&= \arg\min_{u-p^{\mathcal{D}}\in K_I} \sum_{i=1,2}\left(\frac{1}{2}a_i(u,u) + b_i(u)\right) - p^{\mathcal{D}}.
\end{aligned} \tag{3.32}$$

Therefore, the solution $p = p^{\mathcal{D}} + p^{\mathcal{S}}$ of the non-homogeneous SIP (3.24a)–(3.24e) is also the solution of the minimization problem:

$$\min_{u|u-p^{\mathcal{D}}\in K_I} \sum_{i=1,2}\left(\frac{1}{2}a_i(u,u) + b_i(u)\right). \tag{3.33}$$

The domain of minimization in (3.33) may be rewritten as

$$\begin{aligned}
\{u|u - p^{\mathcal{D}} \in K_I\} &= \{u = (u_1, u_2)|\forall i(u_i = J_i \text{ on } \Gamma_{D,i}) \text{ and } (u_i \geq J_i \text{ on } \Gamma_{N,i} \cup \Gamma_I)\} \\
&= \{u|u - \bar{J} \in K_I\},
\end{aligned} \tag{3.34}$$

where $\bar{J} = (\bar{J}_1, \bar{J}_2)$ and $\bar{J}_i$ is an arbitrary continuous function in $\bar{\Omega}_i$ s.t. $\bar{J}_i = J_i$ on $\partial\Omega_i$. This suggests we do not need to explicitly calculate $p^{\mathcal{D}}$ to get p. We can continue to simplify (3.33):

$$\begin{aligned}
p &= \operatorname*{arg\,min}_{u|u-p^{\mathcal{D}}\in K_I} \sum_{i=1,2} \left(\frac{1}{2}a_i(u,u) + b_i(u)\right) \\
&= \operatorname*{arg\,min}_{u|u-\bar{J}\in K_I} \sum_{i=1,2} \left(\frac{1}{2}a_i(u,u) + b_i(u)\right) \\
&= \operatorname*{arg\,min}_{u\in K_I} \sum_{i=1,2} \left(\frac{1}{2}a_i(u+\bar{J}_i, u+\bar{J}_i) + b_i(u+\bar{J}_i)\right) + \bar{J} \\
&= \operatorname*{arg\,min}_{u\in K_I} \sum_{i=1,2} \left(\frac{1}{2}a_i(u,u) + b_i(u) + a_i(\bar{J}_i, u)\right) + \bar{J}.
\end{aligned} \tag{3.35}$$

Now the domain of minimization becomes K_I, and $a_i(\bar{J}_i, u)$ is just another term linear in u. (3.35) may be reduced to variational inequalities and be solved the same way (3.28) is solved.

Our FEM discretization, developed in the next chapter, will be based on minimization (3.35).

Chapter 4

A finite element method for 2D Signorini problems

In this chapter, we will develop a first order finite element method for the 2D non-homogeneous Signorini interface problem based on the variational formulation (3.35). Then in section 4.3, we will give the complete description of how we incorporate the finite element solve into the fluid simulation procedure.

4.1 The computational mesh

Our discretization is based on a staggered grid. The state of a fluid body at time step t^n consists of two parts: the volume it occupies Ω_i^n and the velocity field $\mathbf{u}_i^n$ defined in Ω_i^n. We use level sets for surface tracking. A level set encodes a closed surface implicitly as the zero-contour of a signed distance field ϕ. At any location $\mathbf{x}$, the sign of $\phi(\mathbf{x})$ indicates whether $\mathbf{x}$ is inside the surface and $|\phi(\mathbf{x})|$ is the distance to the nearest point on the surface. Ω_i^n is represented by a level set sampled at integer grid positions, and the discretized velocity is sampled at staggered positions (along the edges of the grid) in a component-wise manner. For different fluid phases, the level sets and discretized velocity fields are defined separately, which may overlap and hold different values in the overlapping area. If one connected fluid body breaks apart duration the simulation, and we would like to impose Signorini conditions if those two chunks contact later, we divide the broken fluid body into multiple phases, each with a separate level set and a separate velocity field. Maintaining separate level sets and velocity fields is necessary for imposing Signorini conditions, otherwise we cannot correctly locate the interface or have distinct velocities assigned to different phases.

We will use linear triangle elements and bilinear square elements in our FEM discretization. We first calculate the position of the nodes from the level set presentation (see figure 4.1). Let V be the set of all nodes. It consists of the following types:

- G_i: grid nodes inside the fluid body Ω_i^h (blue nodes in figure 4.1).
- Boundary nodes. A boundary node is added between neighboring grid nodes if the signed distance values have opposite signs. Boundary nodes are further divided into three types:
  - D_i: boundary nodes on the fluid-air interface $\Gamma_{D,i}$ (white);
  - S_i: boundary nodes on the fluid-solid interface $\Gamma_{S,i}$ (grey);
  - I: merged boundary nodes (see the discussion below) on the fluid-fluid interface Γ_I (red).

Within each cell ($[i, i+1] \times [j, j+1]$), we connect the nodes to form cut-cell shapes and then divide these cut-cell shapes into triangles and squares by the method shown in figure 4.2. We will solve for pressure on all nodes with FEM on these triangle and square elements. In the FEM solve, each node is assigned one degree of freedom for pressure, except for the fluid-fluid interface nodes (red nodes in figure 4.1). For fluid-fluid interface nodes, we will assign one degree of freedom for each *pair* of interface nodes between the same pair of grid nodes, since the pressure on the two nodes are coupled as discussed in section 3.3. The boundary nodes sometimes do not lie exactly on the boundary or do not agree at the interface, due to either a physically present gap or a numerical error in the geometrical representation. We will address this inexactness in section 4.2.5.

We are now ready to assemble the FEM matrices and vectors.

4.2 The assembly process

In this section we will find the discretized definitions for terms in minimization problem (3.35), namely, K_I, a_i, b_i and $\bar{J}_i$.

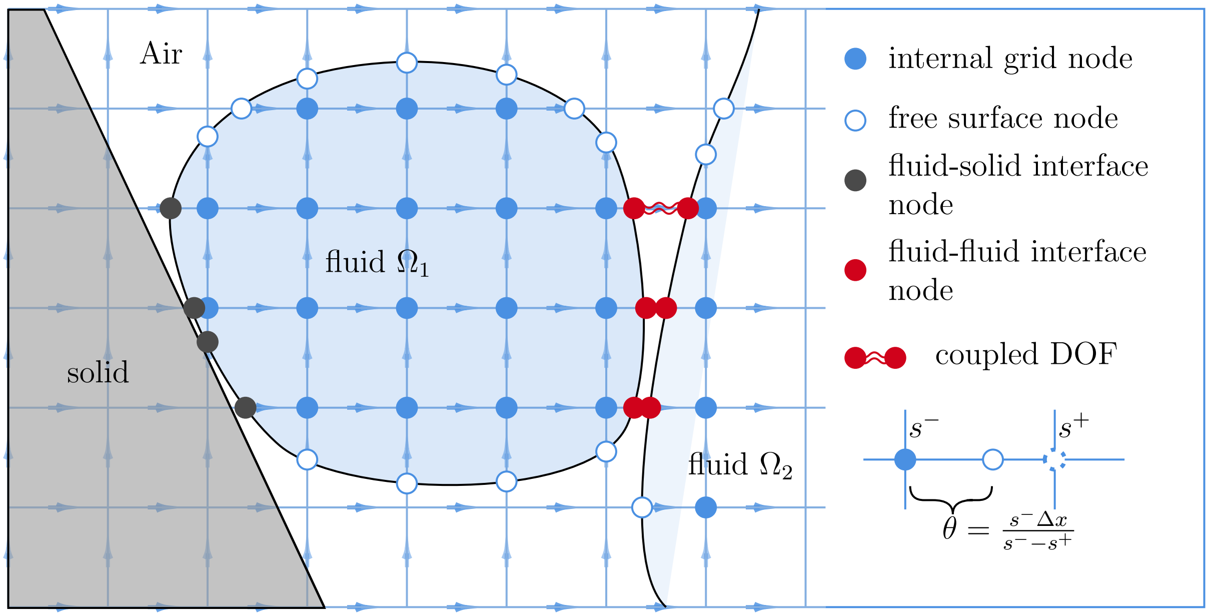
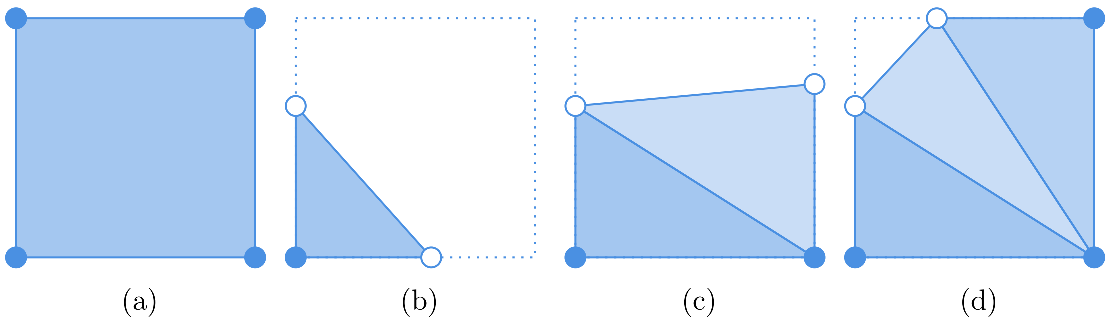


Figure 4.1: Nodes of a cut-cell mesh. The position of the boundary nodes are given by the bottom-right equation, where s is the signed distance value. Velocity samples are on the edges of the grid. Each fluid body keeps a different velocity field, but it is valid only in the region of the corresponding fluid body. Velocity outside the fluid needs to be extrapolated from valid samples.



(a)

(b)

(c)

(d)

Figure 4.2: Subdivided cut-cell shapes. (a): a full submerged cell becomes a bilinear square element. (b): a triangle cut-cell becomes one linear triangle element. (c): a trapezoid cut-cell is subdivided into two triangle elements along its shorter diagonal. (d): a pentagon cut-cell is subdivided into three triangle elements by connecting two boundary nodes to their opposite grid node.

4.2.1 discretized function space

Based on the types of elements we use for the discretization, the set of basis functions of the discretized function space K_I^h are the "hat functions"

$$\phi_v(x) = \sum_{E \text{ incident on } v} \phi_{E,v}(x), \qquad v \in V, \tag{4.1}$$

where $\phi_{E,v}$ is the linear (if E is a triangle) or bilinear (if E is a square) shape function that is non-zero only on element E and

$$\phi_{E,v}(x) = \begin{cases} 1 & \text{at node } v, \\ 0 & \text{at nodes } v' \text{ of } E \text{ s.t. } v' \neq v. \end{cases} \tag{4.2}$$

Let $\Phi = (\phi_v | v \in V)^T$. K_I^h consists of linear combinations of functions in Φ:

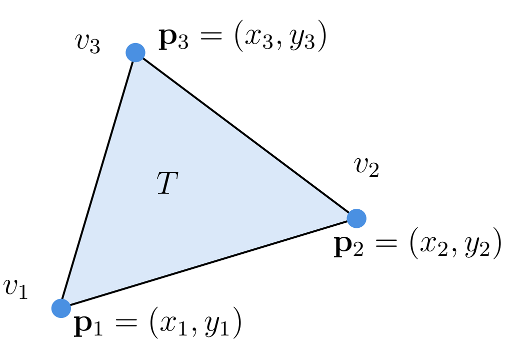
$$\begin{aligned} K_I^h = \{ \ \mathbf{x}^T\Phi \quad & |\mathbf{x} \in \mathbb{R}^{|V|}; \\ & \mathbf{x}_v = 0 \text{ if } (v \in D_1 \cup D_2); \\ & \mathbf{x}_v \geq 0 \text{ if } (v \in S_1 \cup S_2 \cup I) \quad \}, \end{aligned} \tag{4.3}$$

which are globally continuous, element-wise linear or bilinear functions. Equation (4.3) establishes the relation between a field (function) $u^h \in K_I^h$ and a vector $\mathbf{x} \in \mathbb{R}^{|V|}$.

4.2.2 Bilinear forms

For $u^h = \mathbf{x}^T\Phi, v^h = \mathbf{y}^T\Phi \in K_I^h$,

$$\begin{aligned} a_i(u^h, v^h) &= \int_{\Omega_i^h} \rho^{-1} \nabla u^h \cdot \nabla v^h dx \\ &= \sum_{E \subset \Omega_i^h} \int_E \rho^{-1} \nabla \left(\sum_{v \in E} x_v \phi_{E,v} \right) \cdot \nabla \left(\sum_{v' \in E} y_{v'} \phi_{E,v'} \right) dx \\ &= \sum_{E \subset \Omega_i^h} \sum_{v,v' \in E} x_v y_{v'} \int_E \rho^{-1} \nabla \phi_{E,v} \cdot \nabla \phi_{E,v'} dx \\ &= \sum_{E \subset \Omega_i^h} \sum_{v,v' \in E} x_v y_{v'} \mathbf{A}_{E:v,v'} \\ &= \sum_{E \subset \Omega_i^h} \mathbf{x}^T \mathbf{A}_E \mathbf{y} \\ &= \mathbf{x}^T \mathbf{A}_i \mathbf{y}, \end{aligned} \tag{4.4}$$


v_3
$\mathbf{p}_3 = (x_3, y_3)$
T
v_2
$\mathbf{p}_2 = (x_2, y_2)$
v_1
$\mathbf{p}_1 = (x_1, y_1)$

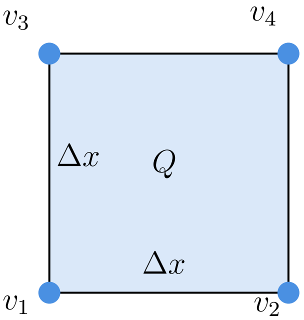


Figure 4.3: A linear triangle element T and a bilinear square element Q.

where $\mathbf{A}_E$ is the local stiffness matrix of element E whose non-zero entries are defined by

$$\mathbf{A}_{E:v,v'} = \int_E \rho^{-1} \nabla \phi_{E,v} \cdot \nabla \phi_{E,v'} dx \tag{4.5}$$

and

$$\mathbf{A}_i = \sum_{E \subset \Omega_i^h} \mathbf{A}_E \tag{4.6}$$

is the global stiffness matrix for fluid body Ω_i.

Linear triangle elements and bilinear square elements are the simplest elements in FEM. A detailed derivation of local stiffness matrices for them may be found in many FEM textbooks (e.g., [63]). Here we directly note down the matrices. Assuming that ρ is constant on each individual element, the local stiffness matrices for a linear triangle T and a bilinear square element as in figure 4.3 are given by

$$\mathbf{A}_T = \frac{1}{4\rho \times \text{Area}(T)} \begin{pmatrix} \|\mathbf{r}_{23}\|^2 & -\mathbf{r}_{13} \cdot \mathbf{r}_{23} & -\mathbf{r}_{32} \cdot \mathbf{r}_{12} \\ -\mathbf{r}_{13} \cdot \mathbf{r}_{23} & \|\mathbf{r}_{13}\|^2 & -\mathbf{r}_{21} \cdot \mathbf{r}_{31} \\ -\mathbf{r}_{12} \cdot \mathbf{r}_{32} & -\mathbf{r}_{21} \cdot \mathbf{r}_{31} & \|\mathbf{r}_{12}\|^2 \end{pmatrix} \tag{4.7}$$

and

$$\mathbf{A}_Q = \frac{1}{6\rho} \begin{pmatrix} 4 & -1 & -1 & 2 \\ -1 & 4 & -2 & -1 \\ -1 & -2 & 4 & -1 \\ -2 & -1 & -1 & 4 \end{pmatrix}, \tag{4.8}$$

where $\mathbf{r}_{ij} = \mathbf{p}_j - \mathbf{p}_i$ is the relative position of node v_i and v_j. With (4.7) and (4.8), we may construct the discretized bilinear forms (or stiffness matrices) $\mathbf{A}_i$.

4.2.3 Linear forms

Next we will express the discretization of two linear terms in (3.21b) as vector products. $\forall v^h = \mathbf{x}^T\Phi \in K_I^h$, we try to write $b(v^h)$ as the sum of two vector products. For the first term:

$$\sum_{i=1,2} \frac{1}{\Delta t} \int_{\Gamma_{S,i}} v^h u_{n,i} ds = \sum_{v\in V} x_v \left(\int_{\Gamma_{S,1}\cup\Gamma_{S,2}} \frac{1}{\Delta t} \phi_v u_n ds \right) \equiv \mathbf{b}_1^T\mathbf{x}, \tag{4.9}$$

the entries of $\mathbf{b}_1$ are

$$\begin{aligned} b_{1,v} &= \int_{\Gamma_{S,1}\cup\Gamma_{S,2}} \frac{1}{\Delta t}\phi_v u_n ds \\ &= \begin{cases} \displaystyle\sum_{\substack{\text{edge } e \text{ incident} \\ \text{on } v}} \int_e \frac{1}{\Delta t}\phi_v u_n ds & \text{if } v \in S_1 \cup S_2, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{4.10}$$

Similarly, for the second term:

$$-\sum_{i=1,2} \frac{1}{\Delta t} \int_{\Omega_i} \mathbf{u}_i^* \cdot \nabla v^h dx = \sum_{v\in V} x_v \left(\int_{\Omega_1\cup\Omega_2} \frac{-1}{\Delta t}\mathbf{u}_i^* \cdot \nabla\phi_v dx \right) \equiv \mathbf{b}_2^T\mathbf{x}, \tag{4.11}$$

the entries of $\mathbf{b}_2$ are

$$b_{2,v} = \int_{\Omega_1\cup\Omega_2} \frac{-1}{\Delta t}\mathbf{u}_i^* \cdot \nabla\phi_v dx = \sum_{\substack{\text{Element } E \text{ in-} \\ \text{cident on } v}} -\frac{1}{\Delta t}\int_E \mathbf{u}_i^* \cdot \nabla\phi_{E,v} dx. \tag{4.12}$$

The integral in (4.10) or (4.12) may be approximated using one point quadrature (or higher order if desired) on each edge e (in (4.10)) or element E (in (4.12)). Together, we may write

$$b(v^h) = (\mathbf{b}_1 + \mathbf{b}_2)^T\mathbf{x}. \tag{4.13}$$

4.2.4 Pressure jump

The discretized pressure jump $\mathbf{j}_i$ is given by

$$j_{i,v} = \begin{cases} \sigma_i\kappa_{i,v} & \text{if } v \in D_i \cup S_i \cup I, \\ 0 & \text{otherwise,} \end{cases} \tag{4.14}$$

where σ_i is the coefficient of surface tension for fluid body i and κ_i is the local mean curvature of $\partial\Omega_i$ at v. The mean curvature κ is estimated using the "height function" technique. Readers are referred to the paper of Sussman and Ohta [54] for details of the technique.

4.2.5 Colliding velocity tolerance

In the last section, we have discretized every concept in (3.35). In this section we will introduce another linear corrective term to counteract the numerical error caused by both spatial and temporal discretization.

According to the discussion in section 4.1, if two surfaces pass through the same pair of neighboring grid nodes, we consider them as being in contact. The "no penetration" conditions (3.24c) and (3.24d) will prevent the two surfaces from getting closer, even though they could be as far as $1\Delta x$ apart. This could result in persisting gaps at fluid-solid or fluid-fluid interfaces, degrade the quality of surfaces and have significant impact on curvature computation. To tackle this problem, we propose a mechanism that allows surfaces to get closer until they close the gap.

If the gap between the two surfaces is sufficiently thin, we can model the gap as a thickness field $\delta : \Gamma_S \cup \Gamma_I \to \mathbb{R}$ on the surface. We propose that instead of enforcing (3.24c) and (3.24d), which stops the surfaces from moving closer at t^{n+1}, we allow the two surfaces to approach at a velocity u_δ that will not cause penetration within the time frame $[t^{n+1}, t^{n+1} + \Delta t]$. Conditions (3.24c) and (3.24d) become

$$p_i^+ \perp^{\geq 0} (u_{n,i} + \frac{\delta}{\Delta t} - \mathbf{u}_i^{n+1} \cdot \mathbf{n}_i) \qquad \text{on } \Gamma_{S,i}, \tag{4.15a}$$

$$p_i^+ \perp^{\geq 0} (\frac{\delta}{\Delta t} - (\mathbf{u}_1^{n+1} \cdot \mathbf{n}_1 + \mathbf{u}_2^{n+1} \cdot \mathbf{n}_2)) \qquad \text{on } \Gamma_I. \tag{4.15b}$$

This will result in modifications to linear form b:

$$\begin{aligned} b'(v) &= \sum_{i=1,2} \frac{1}{\Delta t} \left(\int_{\Gamma_{S,i}} v \left(u_{n,i} + \frac{\delta}{\Delta t} \right) ds - \int_{\Omega_i} \mathbf{u}_i^* \cdot \nabla v dx \right) + \frac{1}{\Delta t^2} \int_{\Gamma_I} v \delta ds. \\ &= b(v) + \frac{1}{\Delta t^2} \int_{\Gamma_{S,1} \cup \Gamma_{S,2} \cup \Gamma_I} v \delta ds. \end{aligned} \tag{4.16}$$

In the discretized setting, we need to estimate the extra integral for each basis function. In figure 4.4, we have shown a cell with two fluid phases. The integral over the shown

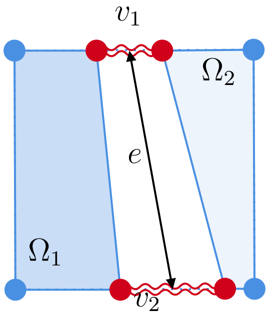


Figure 4.4: A cell containing two fluid phases. We may use the middle line e as an approximation of the shared boundary Γ_I, since the two surfaces do not exactly agree in this cell.

fragment of Γ_I may be estimated as

$$\int_e \phi_v \delta ds \approx \overline{\phi_v} \int_e \delta ds \approx \frac{1}{2} \times (\text{Area of the gap}). \tag{4.17}$$

In (4.17), we have assumed that ϕ_v and δ are independent variables. When the area of the gap approaches 0, the extra integral vanishes, too. Therefore, let $\mathbf{b}_3 \in \mathbb{R}^{|V|}$ and

$$\begin{aligned} b_{3,v} &= \int_{\Gamma_{S,1} \cup \Gamma_{S,2} \cup \Gamma_I} \phi_v \delta ds \\ &= \begin{cases} \frac{1}{2\Delta t^2} \times \begin{pmatrix} \text{Sum of gap area in} \\ \text{neighboring cells} \end{pmatrix} & \text{if } v \in S_1 \cup S_2 \cup I, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{4.18}$$

Then $\forall v^h = \mathbf{x}^T \Phi \in K_I^h$,

$$b'(v^h) = b(v^h) + \mathbf{b}_3^T \mathbf{x}. \tag{4.19}$$

4.2.6 Volume compensation

Standard level sets and semi-Lagrangian advection cause noticeable volume loss during the simulation. We may optionally impose a positive divergence inside the fluids to compensate for the volume loss. This is done by adding a linear term $\mathbf{b}_4^T \mathbf{x}$, where

$$b_{4,v} = \begin{cases} \frac{(V_i^n - V_i^0)}{V_i^n N \Delta t^2} & \text{if } v \in G_i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.20}$$

Intuitively, we distribute the task of sneaking in the lost volume $-(V_i^n - V_i^0)$ evenly across the domain Ω_i^n, which currently has volume V^n, and evenly over a planned time of $N\Delta t$. The remaining Δt is derived from equation (3.1a). Setting $N = 100$ works well in our experiments.

4.3 The algorithm

Putting everything together, we obtain the discretized minimization problem:

$$\min_{\mathbf{x}^T\Phi\in K_I^h} \frac{1}{2}\mathbf{x}^T\left(\sum_{i=1,2}\mathbf{A}_i\right)\mathbf{x} + \left(\mathbf{b}_1^T + \mathbf{b}_2^T + \mathbf{b}_3^T + \mathbf{b}_4^T + \sum_{i=1,2}\mathbf{j}_i^T\mathbf{A}_i\right)\mathbf{x}. \tag{4.21}$$

The constraint $\mathbf{x}^T\Phi \in K_I^h$ is a mixture of equalities and inequalities. Minimization (4.21) may be solved with optimization solvers like Gurobi [24]. The solution of (4.21) concludes the projection step. In the rest of this chapter, we will review the complete simulation procedure and make some additional notes on the implementation details.

Algorithm 1: A single time step of the simulation

```
Input  : discretized velocity fields $\mathbf{u}_i^n$, level sets $s_i$, time increment $\Delta t$.
Output: updated fluid states $\mathbf{u}_i^{n+1}$, $s_i^{n+1}$.
1 Advect $s_i$ in $\mathbf{u}_i^n$, using semi-Lagrangian advection;
2 Advect $\mathbf{u}_i^n$ in $\mathbf{u}_i^n$ to get $\mathbf{u}_i^*$, using semi-Lagrangian advection;
3 Apply external force to $\mathbf{u}_i^*$: $\mathbf{u}_i^* = \mathbf{u}_i^* + \mathbf{f}\Delta t/\rho_i$;
4 Resolve conflicts in advected level sets $s_1$ and $s_2$ using the method in [35]; store
    the conflict-free level sets in $s_i^{n+1}$;
5 Perturb $s_i$ s.t. at every grid node $x$, $|s_i(x)| \geq 0.01\Delta x$; store the perturbed level
    sets in $\tilde{s}_i$;
6 Pressure $\mathbf{p}$ = Project($\mathbf{u}_i^*, \tilde{s}_i, \Delta t$, motion of the solid boundary);
7 Update $\mathbf{u}_i^*$ with $\mathbf{p}_i$; store the new velocity fields in $\mathbf{u}_i^{n+1}$;
8 Extrapolate $\mathbf{u}_i^{n+1}$ to the outside of $\Omega_i$;
9 Output $\mathbf{u}_i^{n+1}$ and $s_i^{n+1}$;
```

Line 5: this may be done by setting $s = \operatorname{sgn}(s)\max(|s|, 0.01)$. It improves the numerical stability of the method, similar to the clamping of θ (see figure 4.1) in the ghost fluid method. The perturbed geometry is used for projection (line 6) and velocity update (line 7), but is not passed on to the next time step.

Line 7: the final pressure field of fluid body i is given by $\mathbf{p}_i = \mathbf{p} + \mathbf{j}_i$. $\mathbf{u}_i^*$ is only updated with values in $\mathbf{p}_i$, and the position of nodes are calculated according to $\tilde{s}_i$ (to be consistent with the projection). Figure 4.5 shows three possible configurations and we will demonstrate how the velocity samples are updated in each of them. Let p_L, p_I, p_R be value of $\mathbf{p}$ at nodes v_L, v_I, v_R as in the figures:

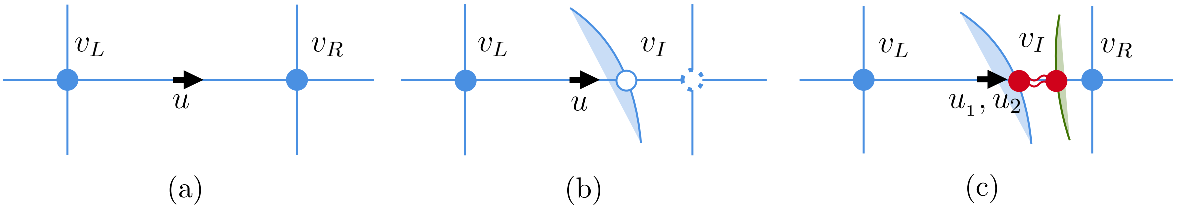


Figure 4.5: 3 different velocity update configurations: (a) the interior of a fluid phase; (b) the boundary of a single fluid phase; (c) the interface between two fluid phases.

- (a) In the interior region of a fluid, central difference is used:

$$u^{n+1} = u^* - \frac{(p_R - p_L)\Delta t}{\rho \Delta x}. \tag{4.22}$$

- (b) At a fluid-solid or fluid-air interface, the ghost fluid method is used:

$$\theta = \frac{\tilde{s}_L \Delta x}{\tilde{s}_L - \tilde{s}_R}, \tag{4.23}$$

$$u^{n+1} = u^* - \frac{(p_I - p_L)\Delta t}{\rho \theta}. \tag{4.24}$$

- (c) At a fluid-fluid interface, the two velocity samples at the same location for different fluid phases are updated independently using the ghost fluid method:

$$\theta_1 = \frac{\tilde{s}_{1,L} \Delta x}{\tilde{s}_{1,L} - \tilde{s}_{1,R}}, \qquad \theta_2 = \frac{\tilde{s}_{2,R} \Delta x}{\tilde{s}_{2,R} - \tilde{s}_{2,L}}, \tag{4.25}$$

$$u_1^{n+1} = u_1^* - \frac{(p_I + j_{1,I} - p_L)\Delta t}{\rho \theta_1}, \qquad u_2^{n+1} = u_2^* - \frac{(p_R - p_I - j_{2,I})\Delta t}{\rho \theta_2}. \tag{4.26}$$

Line 8: if a velocity sample $u_i(x)$ of fluid phase i is not updated in line 7, which happens if there are no neighboring nodes of fluid i, we use constant extrapolation from valid velocity samples in $\mathbf{u}_i$ to override $u_i(x)$. Note that even though velocity may be defined and valid for other fluid phases at location x, we will not use them.

Chapter 5

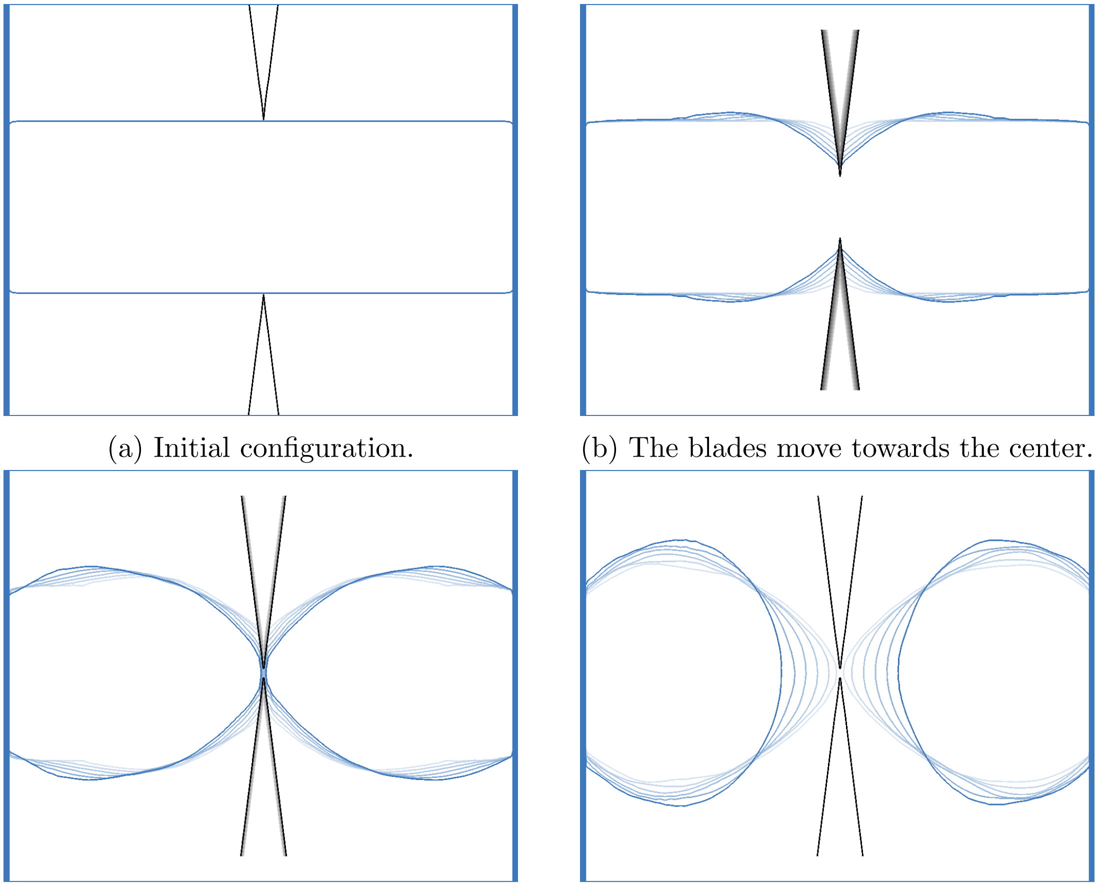
Experiments

In this chapter, we present some numerical experiments on our proposed algorithm. We implemented our algorithm in C++, with Gurobi [24] as our MLCP solver. Here we show the results of a series of experiments that demonstrate different aspects of our algorithm. We did not spend much effort optimizing our code, but the Gurobi MLCP solver is the bottleneck even though it is the only parallelized part of our implementation. Experiment configurations and timing results are summarized in table 5.1.

Hydrophobic knife. This experiment showcases the difference between a Neumann boundary condition and a Signorini boundary condition. As shown in figure 5.1, a rectangular chunk of fluid is placed between a pair of hydrophilic walls (modelled by Neumann boundary conditions) and a pair of hydrophobic blades (modelled by Signorini boundary conditions), in a zero gravity environment. As the hydrophobic blades contact the fluid, the fluid deforms under surface tension forces. As soon as the fluid is cut into halves, it detaches from the hydrophobic blades immediately while continuing to adhere to the outer Neumann boundaries. The difference between a hydrophobic surface and a hydrophilic surface is well modelled by the different boundary conditions.

Normal and tangential separation. Two initially contacting fluid bodies as shown in figure 5.2a are assigned opposite velocity in the normal (figures 5.2c and 5.2d) or tangential direction (figure 5.2b). The fluid bodies are in a zero gravity environment and the surface tension coefficient is 0.

With Signorini conditions on the interface, the fluid PDEs have a trivial solution: $p = 0$ and $\mathbf{u}_i(t) = \mathbf{u}_i(0)$. This suggests that both fluid bodies will move with their initial velocity



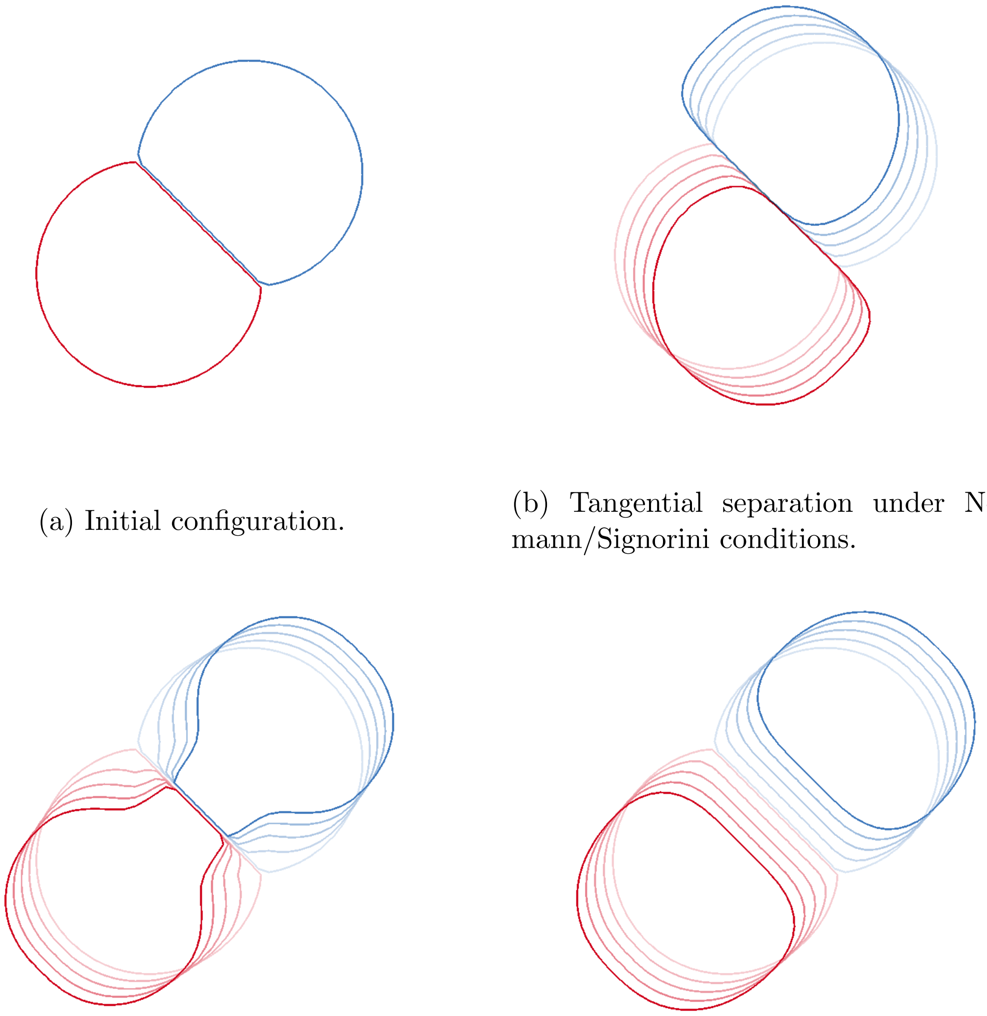
(a) Initial configuration.

(b) The blades move towards the center.

(c) The blades meet and cut the fluid into halves.

(d) The fluid detaches from the hydrophobic blades and stays on the wall.

Figure 5.1: Hydrophobic blades. Different shades of curves indicate a sequence of samples evenly spaced in time. Earlier samples have lighter shades.



(a) Initial configuration.

(b) Tangential separation under Neumann/Signorini conditions.

(c) Normal separation under Neumann conditions

(d) Normal separation under Signorini conditions

Figure 5.2: Normal and tangential separation.

	HK	NS/TS	BD	DB2
Surface tension coefficient σ	1	0	1	0
Density ρ	50	50	50	50
Gravity g_y	0	0	-5	-1
Resolution	100^2	50^2	200×120	100^2
Δx	0.01	0.1	0.01	0.1
Δt	2×10^{-4}	0.01	10^{-3}	0.02
Total time steps	600	50	3000	1000
Total MLCP solve time (s)	103	3.0	3590	1280
Total simulation time (s)	124	3.6	3780	1320

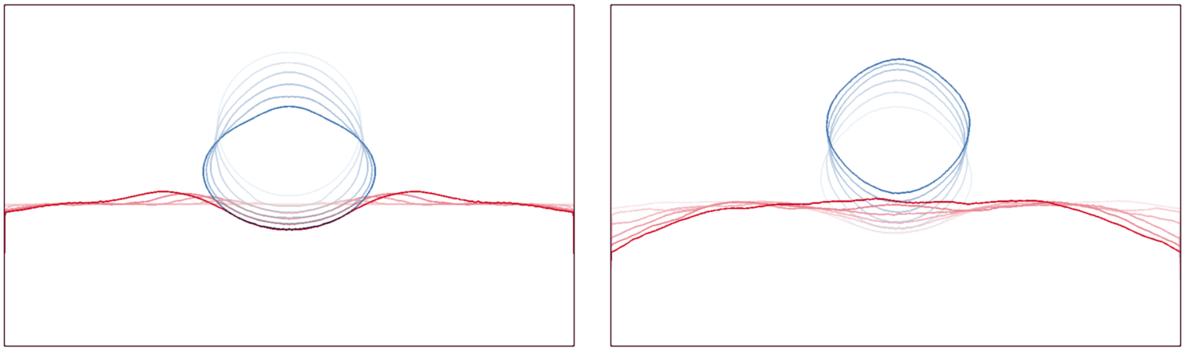
Table 5.1: Experiment configurations and timing results. Experiments: HK – hydrophobic knife; NS/TS – normal/tangential separation; BD – bouncing droplet; DB2 – 2-phase dam break.

without deforming. Under Neumann (free-slip) condition, the tangential slip may happen, but the separation in the normal direction is disallowed.

In figure 5.2b, the fluid bodies are given opposite velocities in the tangential direction of the interface. The behavior of the fluid bodies are the same under Neumann and Signorini conditions. In figures 5.2c and 5.2d, the fluid bodies are given opposite velocities in the normal direction. the Neumann condition is enforced in figure 5.2c. As a result, the two fluid bodies drag each other. In figure 5.2d, the Signorini condition is enforced and they separate without dragging.

Bouncing droplet. In this experiment, we simulate the collision of a droplet with a water pool. A round droplet hit the pool with a downward velocity $v_y = -8$ (equivalent to falling from a height that is 3.2 times of the width of the water pool). Figure 5.3a shows the motion of fluids at the beginning of the impact and figure 5.3b shows the motion after the impact. Under Signorini conditions, the droplet bounces off the surface instead of sticking to it.

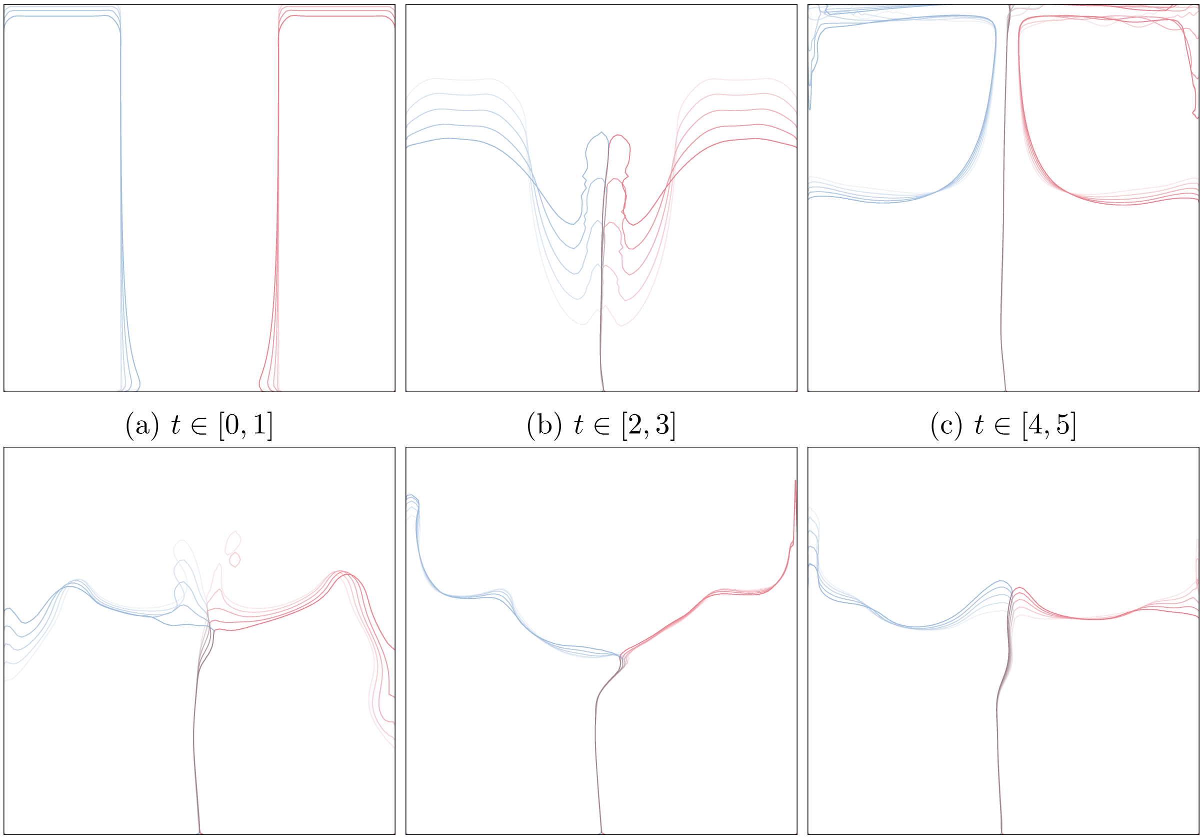
2-phase dam break. In figure 5.4, we showed the evolution of two fluid bodies from their initial configuration (figure 5.4a). We use this experiment to demonstrate the effects of the corrective term $\mathbf{b_3}$ and the volume compensation term $\mathbf{b}_4$. We execute the simulation three times. The first time is the original simulation (figure 5.4). In the second and third run, we turn off $\mathbf{b}_3$ and $\mathbf{b}_4$ respectively.



(a) The droplet (blue) hits the pool (red).

(b) The droplet bounces up.

Figure 5.3: Bouncing droplet.



(a) $t \in [0, 1]$

(b) $t \in [2, 3]$

(c) $t \in [4, 5]$

(d) $t \in [8, 9]$

(e) $t \in [10, 11]$

(f) $t \in [12, 13]$

Figure 5.4: 2-phase dam break test.

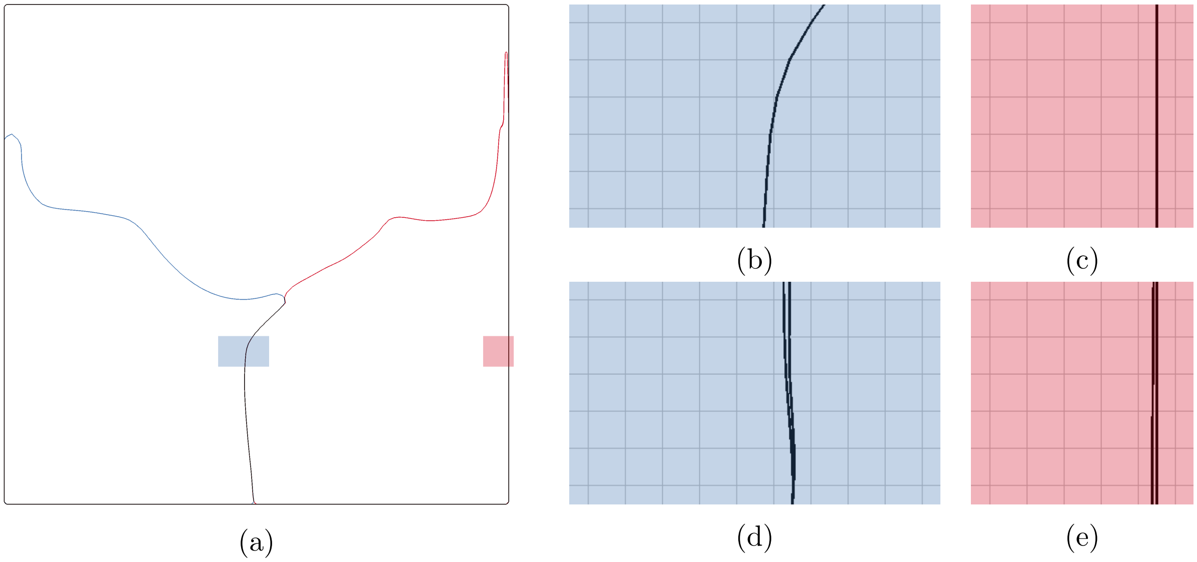
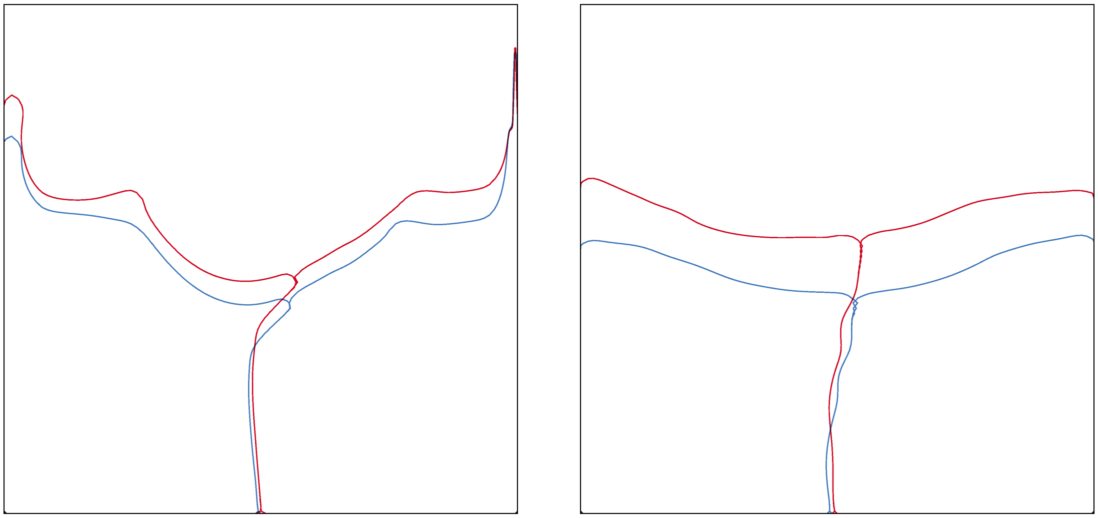


Figure 5.5: Effect of the corrective term $\mathbf{b}_3$. (a) A still frame from a simulation with the corrective term. (b)(c) Zoom in views of fluid-fluid and fluid-solid interfaces. (d)(e) Same patches in the corresponding frame of the simulation without the corrective term.



(a) The 500th frame ($t = 10$).

(b) The 1000th frame ($t = 20$).

Figure 5.6: Effect of the volume compensation term $\mathbf{b}_4$. The red curves are the fluids in the simulation with volume compensation, while blue curves are the fluids at the same time step of the simulation without volume compensation.

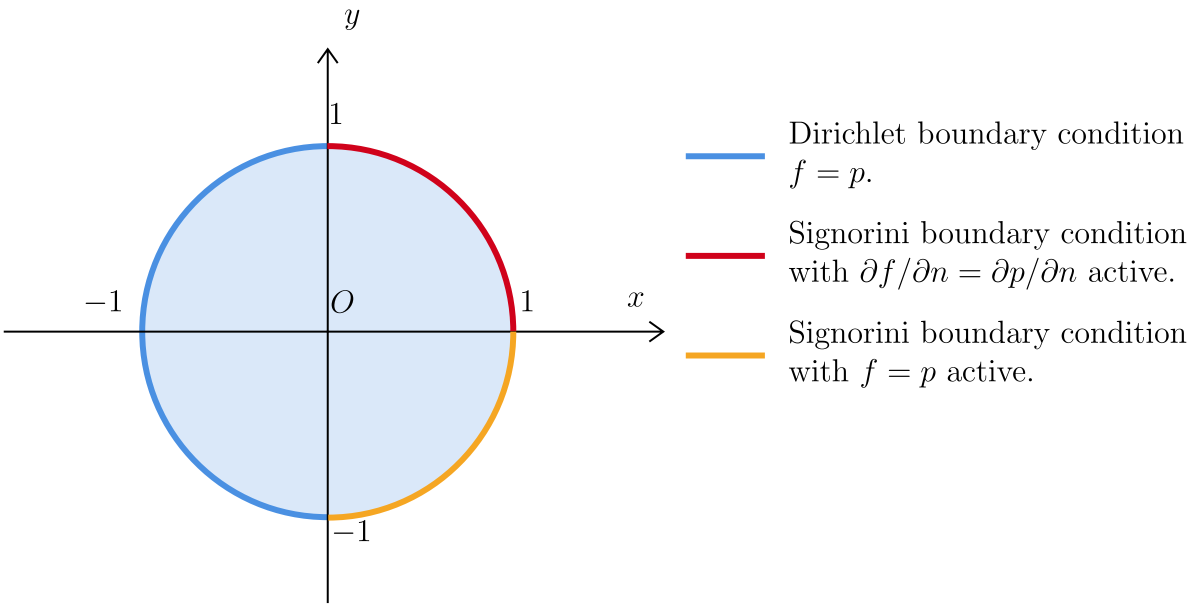

y
1
-1
O
1
x
-1
Dirichlet boundary condition $f = p$.
Signorini boundary condition with $\partial f/\partial n = \partial p/\partial n$ active.
Signorini boundary condition with $f = p$ active.

Figure 5.7: The domain and boundaries of the model problem.

In the simulation without $\mathbf{b}_3$, gaps emerge at fluid-fluid and fluid-solid interfaces. Figure 5.5a is a still frame from the two-phase simulation in the original simulation, showing the relative position of the two fluid phases. Figures 5.5b and 5.5c are the zoom in views of the corresponding patches in figure 5.5a. The level sets of the fluid phases and solid agree at the interfaces. Figures 5.5d and 5.5e are extracted from the same patches of the corresponding frame in the simulation without the corrective term. There are visible gaps between the surfaces and such gaps are common in the entire simulation.

Figure 5.6 shows the how the simulation without $\mathbf{b}_4$ (blue) differs from the original simulation (red) after 500 and 1000 time steps. At the 500th time step, $V_{\text{red}} = 99.2\%V_0$ and $V_{\text{blue}} = 89.0\%V_0$. At the 1000th time step, $V_{\text{red}} = 98.5\%V_0$ and $V_{\text{blue}} = 79.6\%V_0$. We see clearly that volume loss is significant without volume compensation.

Convergence test. Like other first order finite element methods, our method for solving the Signorini boundary problem is second order accurate, which is validated by the following numerical experiment. However, we point out that second order accuracy is only achieved with no perturbation of geometries (cf. algorithm 1, line 5) and accurate integration of boundary conditions. These two premises often do not hold for the full Euler equations.

Let $p(x, y) = e^x \cos y$. We construct a model problem about p and solve for it numeri-

Δx	L_∞ error	$\varepsilon_{n-1}/\varepsilon_n$
1/4	0.0274	
1/8	8.04×10^{-3}	3.41
1/16	2.03×10^{-3}	3.96
1/32	3.70×10^{-4}	5.48
1/64	1.32×10^{-4}	2.80

Table 5.2: Convergence test results.

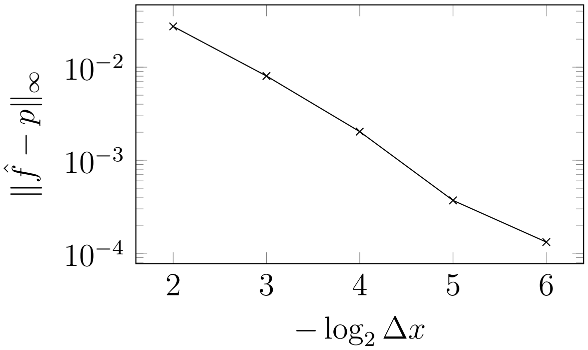


Figure 5.8: $\varepsilon - \Delta x$ plot.

cally:

$$\begin{aligned}
\nabla \cdot \nabla f &= 0 && x^2 + y^2 < 1, && (5.1a)\\
f &= p && x^2 + y^2 = 1 \text{ and } x < 0, && (5.1b)\\
(f - (p - |xy|)) \perp^{\geq 0} \left(\frac{\partial f}{\partial n} - \frac{\partial p}{\partial n}\right) & && x^2 + y^2 = 1 \text{ and } x, y \geq 0, && (5.1c)\\
(f - p) \perp^{\geq 0} \left(\frac{\partial f}{\partial n} - \left(\frac{\partial p}{\partial n} - |xy|\right)\right) & && x^2 + y^2 = 1 \text{ and } y < 0 < x. && (5.1d)
\end{aligned}$$

The domain and boundary conditions are illustrated in figure 5.7. Signorini conditions (5.1c) and (5.1d) relax on f and its gradient respectively. The positive relaxation term $|xy|$ should not affect the solution ($f = p$) as the relaxed condition becomes inactive.

We solve for the approximate fields $\hat{f}_n$ on grids with resolution $(4 \times 2^n)^2$, and record the errors $\varepsilon_n = \|\hat{f}_n - p\|_\infty$ in table 5.2. Figure 5.8 shows the $\varepsilon - \Delta x$ plot in logarithmic scale. A regression analysis suggests

$$\log_2 \varepsilon \approx 1.984 \log_2 \Delta x - 1.141. \tag{5.2}$$

The results show that our algorithm correctly identifies the active condition in a Signorini condition and is approximately second order accurate.

Chapter 6

Conclusions and discussions

In this thesis, we introduced the Signorini boundary condition for the Euler equations and proposed a numerical solution to the problem in two dimensions. We further evaluated this method in some representative scenarios and demonstrated that it successfully achieved the intended behavior. In this chapter, we will make some additional remarks on our algorithm, share some lessons we learned during our research, and discuss some potential future work.

Challenges of discretization. We identify two challenges in the discretization of Signorini problems, regardless of the numerical schemes being used. Our 2D FEM discretization meets both requirements.

1. Signorini boundary conditions are ambiguous at the time of discretization. The DOFs must be defined in a way that they could handle both Dirichlet and Neumann (or free-slip) conditions.
2. The discretization must be continuous across the fluid-fluid interface. No matter how the MLCP is derived, each individual inequality for an interface DOF eventually boils down to a discrete pressure-flux (velocity) complementarity condition, which requires the DOFs being shared by the two phases. E.g., if the two phases are discretized with unrelated tetrahedron meshes, the contacting area must be remeshed to properly set up DOFs.

Colliding velocity tolerance. For any method that uses multiple independent level sets and velocity fields, determining whether two surfaces are in contact (separated by numerical errors) or separated by a small physical distance is a tricky task. The corrective term introduced in section 4.2.5 is a neat solution to this problem. It handles the ambiguity in a smooth and robust way without relying on sophisticated geometry processing techniques.

Disadvantages of FEM. We have discretized 2D domains into finite elements relatively easily. However, this is not as easy in 3D. For partial boundary cells in 3D, there are up to 15 different cut-cell shapes (there are 3 in 2D, cf. figure 4.2) that we need to further subdivide into tetrahedron or pyramid elements. Certain cut-cell shapes require more than 10 tetrahedrons for the subdivision. This also makes coding more complicated. In terms of efficiency, 2D FEM has a 9-point stencil, which is much larger than the 5-point stencil resulting from a standard finite difference discretization. The MLCP solve is therefore slower.

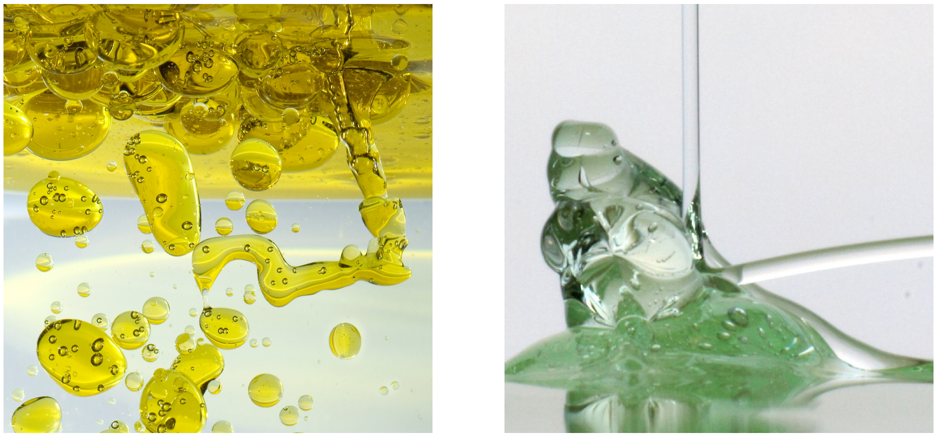
We also list some potential directions for future work.

Implementation in 3D. It is the obvious next step. Standard 3D tri-linear cube elements will cause a 27-point stencil. A lumped stiffness matrix may help improve efficiency.

Better surface tracking. Volume compensation is an effective workaround to reduce volume loss, but after all it has no physical foundation. It is possible to reduce volume loss by using better surface tracking methods, such as the particle level set method [17].

Implicit treatment of surface tension. Many of the fluid phenomena we are interested in are dominated by surface tension forces. Strong surface tension poses a very stringent restriction on the size of the time step. This causes increased running time. Moreover, the volume loss typical for a level set representation becomes more visible after thousands of time steps. It may help to adopt an implicit treatment for surface tension [54].

Coalescence of fluids. For many fluids, the non-coalescence phenomenon is often followed by (delayed) coalescence. In the bouncing droplet example, the droplet will sit on the water pool after a few bounces, then after a few seconds, the droplet may suddenly coalesce with the water pool. Sometimes this process generates a smaller droplet that bounces, which is referred to as the “cascading coalescence”. It would be interesting to visually simulate these phenomena.



(a) Floating oil droplets in water

(b) Kaye effect

Figure 6.1: Non-coalescence behavior of viscous fluids.

Viscous fluids. Non-coalescence is observed in viscous fluids, too. For example, when oil is poured into water, it is scattered into small droplets. The droplets float up to the surface because its density is lower. At the surface, however, those contacting oil droplets will not immediately merge into a larger one. They remain small oil droplets for some time, as shown in figure 6.1a. A similar phenomenon also happens inside a lava lamp, where molten wax blobs float to the top of the container without immediately merging. Another complicated behavior of some non-Newtonian fluids, the Kaye effect, is also believed to have connections with the non-coalescence phenomenon [59]. The Kaye effect refers to the phenomenon where a fluid jet pouring downward occasionally slips off the pile and shoots sideways, as shown in figure 6.1b.

For viscous fluid, the "no attraction" condition also involves viscous stress, instead of just the pressure. In the case of floating oil droplets, the droplets are submerged in water, instead of air which we treat as vacuum or a constant pressure field, and thus it would require support for three or more distinct phases. In the case of Kaye effect, we need to handle the self-contact of surfaces. All of these add complexity to our problem.

References

[1] Ahmed Abdelkader, Chandrajit L Bajaj, Mohamed S Ebeida, Ahmed H Mahmoud, Scott A Mitchell, John D Owens, and Ahmad A Rushdi. Vorocrust: Voronoi meshing without clipping. *ACM Transactions on Graphics (TOG)*, 39(3):1–16, 2020.

[2] David Adalsteinsson and James A Sethian. A fast level set method for propagating interfaces. *Journal of computational physics*, 118(2):269–277, 1995.

[3] Jérémie Allard, François Faure, Hadrien Courtecuisse, Florent Falipou, Christian Duriez, and Paul G Kry. Volume contact constraints at arbitrary resolution. In *ACM SIGGRAPH 2010 papers*, pages 1–10. 2010.

[4] Michael Andersen, Sarah Niebe, and Kenny Erleben. A fast linear complementarity problem (lcp) solver for separating fluid-solid wall boundary conditions. In *Proceedings of the 13th Workshop on Virtual Reality Interactions and Physical Simulations*, pages 39–48, 2017.

[5] David Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 223–232, 1989.

[6] Christopher Batty, Florence Bertails, and Robert Bridson. A fast variational framework for accurate solid-fluid coupling. In *ACM Transactions on Graphics (TOG)*, volume 26, page 100. ACM, 2007.

[7] Christopher Batty and Ben Houston. A simple finite volume method for adaptive viscous liquids. In *Proceedings of the 2011 ACM SIGGRAPH /Eurographics Symposium on Computer Animation*, SCA '11, pages 111–118, New York, NY, USA, 2011. ACM.

[8] Landon Boyd and Robert Bridson. Multiflip for energetic two-phase fluid simulation. *ACM Transactions on Graphics (TOG)*, 31(2):16, 2012.

[9] Robert Bridson. *Fluid simulation for computer graphics.* AK Peters/CRC Press, 2015.

[10] Tyson Brochu, Christopher Batty, and Robert Bridson. Matching fluid simulation elements to surface geometry and topology. In *ACM SIGGRAPH 2010 papers*, pages 1–9. 2010.

[11] John WM Bush. The new wave of pilot-wave theory. 2015.

[12] G Eo Charles and So Go Mason. The mechanism of partial coalescence of liquid drops at liquid/liquid interfaces. *Journal of Colloid Science*, 15(2):105–122, 1960.

[13] Nuttapong Chentanez, Bryan E Feldman, François Labelle, James F O'Brien, and Jonathan R Shewchuk. Liquid simulation on lattice-based tetrahedral meshes. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 219–228, 2007.

[14] Nuttapong Chentanez and Matthias Mueller-Fischer. A multigrid fluid pressure solver handling separating solid boundary conditions. *IEEE transactions on visualization and computer graphics*, 18(8):1191–1201, 2012.

[15] Pascal Clausen, Martin Wicke, Jonathan R Shewchuk, and James F O'Brien. Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Transactions on Graphics (TOG)*, 32(2):1–15, 2013.

[16] Richard W Cottle. *Linear complementarity problem.* Springer, 2009.

[17] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational physics*, 183(1):83–116, 2002.

[18] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of computational physics*, 152(2):457–492, 1999.

[19] Bryan E Feldman, James F O'Brien, and Bryan M Klingner. Animating gases with hybrid meshes. *ACM Transactions on Graphics (TOG)*, 24(3):904–909, 2005.

[20] Gaetano Fichera. Existence theorems in elasticity. In *Linear theories of elasticity and thermoelasticity*, pages 347–389. Springer, 1973.

[21] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graphical models and image processing*, 58(5):471–483, 1996.

[22] Dan Gerszewski and Adam W Bargteil. Physics-based animation of large-scale splashing liquids. *ACM Trans. Graph.*, 32(6):185–1, 2013.

[23] Arthur Guittet, Mathieu Lepilliez, Sebastien Tanguy, and Frédéric Gibou. Solving elliptic problems with discontinuities on irregular domains–the voronoi interface method. *Journal of Computational Physics*, 298:747–765, 2015.

[24] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021.

[25] FH Harlow. The marker-and-cell method. *In Von Karman Inst. for Fluid Dyn. Numerical Methods in Fluid Dyn. 38 p (SEE N79-27448 18-34*, 1972.

[26] Ásdís Helgadóttir, Yen Ting Ng, Chohong Min, and Frédéric Gibou. Imposing mixed dirichlet–neumann–robin boundary conditions in a level-set framework. *Computers & Fluids*, 121:68–80, 2015.

[27] Libo Huang, Torsten Hädrich, and Dominik L Michels. On the accurate large-scale simulation of ferrofluids. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.

[28] Noboru Kikuchi and John Tinsley Oden. *Contact problems in elasticity: a study of variational inequalities and finite element methods*. SIAM, 1988.

[29] Bryan M Klingner, Bryan E Feldman, Nuttapong Chentanez, and James F O'Brien. Fluid animation with dynamic meshes. In *ACM SIGGRAPH 2006 Papers*, pages 820–825. 2006.

[30] Junyu Lai, Yangang Chen, Yu Gu, Christopher Batty, and Justin WL Wan. Fast and scalable solvers for the fluid pressure equations with separating solid boundary conditions. In *Computer Graphics Forum*, volume 39, pages 23–33. Wiley Online Library, 2020.

[31] Egor Larionov, Christopher Batty, and Robert Bridson. Variational stokes: a unified pressure-viscosity solver for accurate viscous liquids. *ACM Transactions on Graphics (TOG)*, 36(4):101, 2017.

[32] Jun Li and Nan-jing Huang. Vector F-implicit complementarity problems in banach spaces. *Applied mathematics letters*, 19(5):464–471, 2006.

[33] Konstantin Lipnikov, Gianmarco Manzini, and Mikhail Shashkov. Mimetic finite difference method. *Journal of Computational Physics*, 257:1163–1227, 2014.

[34] Konstantin Lipnikov, Mikhail Shashkov, and Ivan Yotov. Local flux mimetic finite difference methods. *Numerische Mathematik*, 112(1):115–152, 2009.

[35] Frank Losasso, Tamar Shinar, Andrew Selle, and Ronald Fedkiw. Multiple interacting liquids. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 812–819. ACM, 2006.

[36] Per Lötstedt. Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal on Applied Mathematics*, 42(2):281–296, 1982.

[37] Marek Krzysztof Misztal, Kenny Erleben, Adam Bargteil, Jens Fursund, Brian Bunch Christensen, Jakob Andreas Bærentzen, and Robert Bridson. Multiphase flow of immiscible fluids on unstructured moving meshes. *IEEE transactions on visualization and computer graphics*, 20(1):4–16, 2013.

[38] Nathan Mitchell, Mridul Aanjaneya, Rajsekhar Setaluri, and Eftychios Sifakis. Non-manifold level sets: A multivalued implicit surface representation with applications to self-collision processing. *ACM Transactions on Graphics (TOG)*, 34(6):247, 2015.

[39] Joe J Monaghan. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30(1):543–574, 1992.

[40] José Luis Morales, Jorge Nocedal, and Mikhail Smelyanskiy. An algorithm for the fast solution of symmetric linear complementarity problems. *Numerische Mathematik*, 111(2):251–266, 2008.

[41] Rahul Narain, Abhinav Golas, and Ming C Lin. Free-flowing granular materials with two-way solid coupling. In *ACM SIGGRAPH Asia 2010 papers*, pages 1–10. 2010.

[42] G Paul Neitzel and Pasquale Dell'Aversana. Noncoalescence and nonwetting behavior of liquids. *Annual review of fluid mechanics*, 34(1):267–289, 2002.

[43] Yen Ting Ng, Chohong Min, and Frédéric Gibou. An efficient fluid–solid coupling algorithm for single-phase flows. *Journal of Computational Physics*, 228(23):8807–8829, 2009.

[44] Melissa Orme. Experiments on droplet collisions, bounce, coalescence and disruption. *Progress in Energy and Combustion Science*, 23(1):65–79, 1997.

[45] Marcel Padilla, Albert Chern, Felix Knöppel, Ulrich Pinkall, and Peter Schröder. On bubble rings and ink chandeliers. *ACM Transactions on Graphics (TOG)*, 38(4):129, 2019.

[46] Joseph Papac, Frédéric Gibou, and Christian Ratsch. Efficient symmetric discretization for the poisson, heat and stefan-type problems with robin boundary conditions. *Journal of Computational Physics*, 229(3):875–889, 2010.

[47] Mark Pauly, Dinesh K Pai, and Leonidas J Guibas. Quasi-rigid objects in contact. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 109–119, 2004.

[48] Stéphane Popinet. Numerical models of surface tension. *Annual Review of Fluid Mechanics*, 50:49–75, 2018.

[49] Daniel Ram, Theodore Gast, Chenfanfu Jiang, Craig Schroeder, Alexey Stomakhin, Joseph Teran, and Pirouz Kavehpour. A material point method for viscoelastic fluids, foams and sponges. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 157–163. ACM, 2015.

[50] WD Ristenpart, JC Bird, A Belmonte, F Dollar, and Howard A Stone. Non-coalescence of oppositely charged drops. *Nature*, 461(7262):377, 2009.

[51] Antonio Signorini. Sopra alcune questioni di elastostatica. *Atti della Societa Italiana per il Progresso delle Scienze*, 27:69, 1933.

[52] Jos Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128, 1999.

[53] Mark Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187(1):110–136, 2003.

[54] Mark Sussman and Mitsuhiro Ohta. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM Journal on Scientific Computing*, 31(4):2447–2471, 2009.

[55] Denis Terwagne, Nicolas Vandewalle, and Stéphane Dorbolo. Lifetime of a bouncing droplet. *Physical Review E*, 76(5):056311, 2007.

[56] Steven Trainoff. Watch the bouncing droplet. https://www.youtube.com/watch?v=pbGz1njqhxU. Accessed: 2019-07-07.

[57] Nicolas Vandewalle, Denis Terwagne, K Mulleners, Tristan Gilet, and Stéphane Dorbolo. Dancing droplets onto liquid surfaces. *Physics of Fluids*, 18(9):091106, 2006.

[58] Mickeal Verschoor and Andrei C Jalba. Efficient and accurate collision response for elastically deformable models. *ACM Transactions on Graphics (TOG)*, 38(2):1–20, 2019.

[59] Michel Versluis, Cor Blom, Devaraj van der Meer, Ko van der Weele, and Detlef Lohse. Leaping shampoo and the stable kaye effect. *Journal of statistical mechanics: theory and experiment*, 2006(07):P07007, 2006.

[60] Gangjoon Yoon and Chohong Min. Analyses on the finite difference method by gibou et al. for poisson equation. *Journal of Computational Physics*, 280:184–194, 2015.

[61] Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. Continuum foam: A material point method for shear-dependent flows. *ACM Transactions on Graphics (TOG)*, 34(5):160, 2015.

[62] Tie Zhang and Zheng Li. An analysis of finite volume element method for solving the signorini problem. *Applied Mathematics and Computation*, 270:830–841, 2015.

[63] Olek C Zienkiewicz, Robert L Taylor, and Jian Z Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.