



**HAL**  
open science

# Simulation and optimization of complex phenomena in multiphase flows

Tomas Fullana

► **To cite this version:**

| Tomas Fullana. Simulation and optimization of complex phenomena in multiphase flows. Mechanics [physics]. Sorbonne Université, 2022. English. NNT : 2022SORUS476 . tel-04053531

**HAL Id: tel-04053531**

**<https://theses.hal.science/tel-04053531>**

Submitted on 31 Mar 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SORBONNE UNIVERSITÉ

École doctorale Sciences Mécaniques, Acoustique, Électronique et Robotique

THÈSE DE DOCTORAT  
SPÉCIALITÉ MÉCANIQUE

---

## Simulation and optimization of complex phenomena in multiphase flows

---

*Simulation et optimisation de phénomènes complexes  
dans des écoulements multiphasiques*

Présentée par **Tomas FULLANA**

pour obtenir le grade de Docteur de Sorbonne Université

**Sous la direction de** Taraneh SAYADI, Chargée de Recherche  
Bruno DESPRÉS, Professeur

à l'Institut Jean Le Rond d'Alembert, Sorbonne Université, CNRS, UMR 7190

**Soutenue le 08/12/2022** devant le jury composé de :

Mme. Taraneh SAYADI	Directrice
M. Bruno DESPRÉS	Co-directeur
M. Christophe JOSSERAND	Rapporteur
M. Peter SCHMID	Rapporteur
M. Arnaud ANTKOWIAK	Examinateur
Mme. Marica PELANTI	Examinatrice
M. Denis SIPP	Examinateur
M. Gretar TRYGGVASON	Examinateur
M. Vincent LE CHENADEC	Invité
M. Stéphane ZALESKI	Invité



# Remerciements

Je tiens à remercier, tout d'abord, l'Institut des Sciences du Calcul et des Données et son directeur, Pascal Frey, de m'avoir permis de réaliser cette thèse, et Bruno Després d'avoir accepté de faire partie de cette aventure. Mes remerciements vont également à l'école doctorale et son directeur, Djimedo Kondo, ainsi qu'à l'Institut Jean le Rond d'Alembert dans son ensemble.

Je remercie Arnaud Antkowiak, Marica Pelanti, Denis Sipp et Gretar Tryggvason pour leurs commentaires et questions qui ont été d'une grande précision et ont ouvert la voie à de nombreuses extensions possibles de ce travail.

Je suis extrêmement reconnaissant aux rapporteurs de cette thèse, Christophe Josserand et Peter Schmid, qui m'ont suivi tout au long de ces trois dernières années et qui ont su aiguiller mes recherches, notamment pendant les périodes de confinement où tout semblait bien étrange.

Merci à Taraneh Sayadi qui, dès notre première réunion dans son bureau, a su diriger avec brio cet ambitieux projet de recherche. Je sais que la fin de la thèse ne marquera en aucun cas la fin de notre collaboration et j'en suis très heureux. Mes remerciements vont aussi à Vincent Le Chenadec, qui a été un exemple de jeune chercheur passionné et auprès de qui j'aurais énormément appris.

Merci à Stéphane Zaleski de m'avoir fait confiance il y a bientôt quatre ans et d'avoir été, depuis ce moment-là, un mentor pour moi.

Je ne saurais manquer de remercier Gustav Amberg et les membres du groupe KTH pour les précieuses discussions sur la ligne de contact.

Je tiens aussi à remercier tous mes professeurs et, en particulier, Hélène Dumontet, Catherine Weisman et Jean-Camille Chassaing pour leur soutien sans faille depuis le début de mon cursus.

Je remercie Yash, Mathis, Alexandre, tous mes collègues du Zpulse et de d'Alembert plus largement, avec qui j'ai travaillé et passé beaucoup de bons moments. Merci à Serena, Clément et Alejandro pour leur sympathie, leur bonne humeur et leur amitié, sans oublier, bien entendu, Gonzalo qui m'a supporté pendant toutes ces années en face de lui.

Je remercie Flo, Louis, Lucile et tous mes amis de m'avoir fait relativiser, voire oublier mes petits problèmes de doctorant.

Je remercie également ma famille de cœur, parisienne, qui se reconnaîtra par ces lignes.

Merci à mes oncles, tantes, cousins, cousines et à ma grand-mère Ana qui m'ont toujours soutenu même si un hémisphère nous sépare. J'ai une pensée pour mes grands-parents María-Lucía et Alberto à qui je dois beaucoup.

Merci à mes parents, Carolina et José, et à ma sœur, Emilia, d'avoir toujours été à mes côtés.

Je dédie cette thèse à Hannah, à qui, un seul merci ne suffirait pas à exprimer toute la gratitude et l'amour que je lui porte.

---

## **Simulation et optimisation de phénomènes complexes dans des écoulements multiphasiques**

**Résumé:** Dans cette thèse, nous étudions des phénomènes complexes présents dans les écoulements multiphasiques en utilisant diverses méthodes et modèles numériques. L'accent est mis, tout d'abord, sur le problème diphasique de Stefan, un problème de changement de phase où le mouvement de l'interface est lié au saut des flux conductifs de chaleur. Les formes complexes de l'interface dans la croissance cristalline, en présence d'effets d'anisotropie, sont prédites avec précision grâce à notre méthode d'ensemble de niveaux couplée à une nouvelle méthode de cellules coupées pour le transport diffusif. Une procédure d'optimisation de la forme de cette interface, basée sur la dérivation de l'adjoint sous forme continue, avec une fonction de coût de type suivi d'interface, est ensuite construite et testée sur différentes configurations physiques. En considérant les équations de Navier-Stokes dans l'approximation de Boussinesq, une étude sur l'apparition de l'instabilité de Rayleigh-Bénard en présence d'une frontière de fusion est également réalisée. Dans la deuxième partie, nous étudions les problèmes de lignes de contact dynamiques, où une interface fluide-fluide est en contact avec une frontière solide. Plusieurs nouveaux modèles sont mis en place au sein d'une méthode du volume de fluide, tel que la condition limite généralisée de Navier, où la vitesse de la ligne de contact dépend de la contrainte de Young non compensée, ou encore le modèle-jouet, où le flux massique existant dans les méthodes à interface diffuse est mis en relation avec la courbure de l'interface, résultant en une simple relation d'angle de contact dynamique.

**Mots-clés:** problèmes de Stefan, adjoint continu, méthode d'ensemble de niveaux, méthode de cellules coupées, lignes de contact, méthode du volume de fluide

---

---

## Simulation and optimization of complex phenomena in multiphase flows

**Abstract:** In this thesis, we study complex phenomena in multiphase flows using various new numerical methods and models. The focus is, first, set on the two-phase Stefan problem, a phase-change problem where the motion of the interface is related to the jump in conductive heat fluxes. The complex interfacial shapes in crystal growth, in the presence anisotropy effects, are accurately predicted thanks to our level set method coupled with a novel Cut Cell method for diffusive transport. A shape optimization procedure with a tracking-type cost functional based on a continuous adjoint derivation is then built and tested on various physical setups, yielding favorable results. Considering the Navier-Stokes equations in the Boussinesq approximation, a study of onset of Rayleigh-Bénard instability in the presence of a melting boundary is also carried out. In the second part, we investigate dynamic contact line problems, where a fluid-fluid interface meets a solid boundary. Various novel methods are implemented in a Volume-Of-Fluid framework, such as the generalized Navier boundary condition, where the speed of the contact line depends on the uncompensated Young's stress, or the toy model, where the existing mass flux in diffuse interface models is related the interface curvature, resulting in a simple dynamic angle relation.

**Keywords:** Stefan problems, continuous adjoint, level set method, Cut Cell method, contact lines, Volume-Of-Fluid method

---

# Contents

Résumé . . . . .	i
Abstract . . . . .	ii
List of Figures . . . . .	v
<b>Introduction</b>	<b>1</b>
<b>I Shape optimization of melting and solidification processes</b>	<b>6</b>
<b>1 The two-phase Stefan problem</b>	<b>7</b>
1.1 Classical two-phase Stefan problem formulation . . . . .	7
1.2 The level set method . . . . .	9
1.3 Thermal convection and fluid flow extension . . . . .	12
<b>2 Adjoint-based optimization algorithm</b>	<b>15</b>
2.1 Minimization problem . . . . .	15
2.2 Continuous adjoint derivation . . . . .	18
2.3 Gradient-based optimization procedure . . . . .	26
<b>3 A novel Cut Cell method for diffusive transport</b>	<b>29</b>
3.1 Discrete Cut Cell operators . . . . .	29
3.2 Volume and surface capacities . . . . .	37
3.3 Marching squares algorithm . . . . .	40
3.4 Validation on stationary geometries . . . . .	43
<b>4 Numerical methods for forward and adjoint problems</b>	<b>51</b>
4.1 Algorithm for solving the forward Stefan problem . . . . .	51
4.2 Phase-change velocity . . . . .	53
4.3 The advection equation . . . . .	56
4.4 Reinitialization and initialization steps . . . . .	61
4.5 Algorithm for solving the adjoint Stefan problem . . . . .	64
<b>5 Numerical simulations of two-phase Stefan problems</b>	<b>67</b>
5.1 Flower.jl an in-house Julia package . . . . .	67
5.2 Analytical solutions of two-dimensional problems . . . . .	68
5.3 Crystal growth in an under-cooled liquid bath . . . . .	72
5.4 Rayleigh-Bénard convection with a melting boundary . . . . .	81
<b>6 Shape optimization results</b>	<b>85</b>
6.1 Actuation function parametrization . . . . .	85
6.2 Adjoint-based optimization cases . . . . .	86
6.3 Comparison with derivative-free methods . . . . .	93

---

<b>II</b>	<b>Physico-mathematical models for the moving contact line</b>	<b>94</b>
<b>7</b>	<b>Two-phase Navier-Stokes equations and Phase-Field extensions</b>	<b>95</b>
7.1	Two-phase flows in the one-fluid formulation . . . . .	95
7.2	Phase-Field extensions . . . . .	97
<b>8</b>	<b>Singularities in flows with contact lines</b>	<b>101</b>
8.1	The static contact line . . . . .	101
8.2	Paradoxes and singularities in the dynamic case . . . . .	102
<b>9</b>	<b>Numerical methods for contact lines</b>	<b>105</b>
9.1	The Volume-Of-Fluid method . . . . .	105
9.2	Classical models for contact lines . . . . .	109
9.3	Application to physical systems . . . . .	111
<b>10</b>	<b>Advances in models for contact lines</b>	<b>121</b>
10.1	The super-slip model . . . . .	121
10.2	The generalized Navier boundary condition . . . . .	126
10.3	The toy model: mimic Phase-Field at lower cost . . . . .	135
10.4	Summary of models and applications . . . . .	143
	<b>Conclusion</b>	<b>147</b>
<b>III</b>	<b>Appendices</b>	<b>150</b>
<b>A</b>	<b>Transport and shape calculus theorems</b>	<b>151</b>
<b>B</b>	<b>Cut Cell method for Navier-Stokes equations</b>	<b>153</b>
<b>C</b>	<b>Flower.jl package description</b>	<b>159</b>
<b>D</b>	<b>Spurious currents study for static drops</b>	<b>163</b>
<b>E</b>	<b>Numerical implementation of the super-slip model</b>	<b>169</b>
<b>F</b>	<b>Kinematic transport of the contact angle</b>	<b>173</b>
<b>G</b>	<b>Relaxation of the impermeability condition in the toy model</b>	<b>177</b>
	<b>Bibliography</b>	<b>191</b>

# List of Figures

1.1	Schematic of the liquid and solid subdomains. . . . .	7
1.2	Photography of a crystal growing in an under-cooled liquid bath. . .	9
1.3	View of the level set signed distance function $\phi$ . . . . .	10
1.4	Schematic of the two-phase Stefan problem in the level set framework.	11
1.5	Photography of an atmospheric surface layer. . . . .	12
1.6	Schematic of the melting boundary problem. . . . .	13
2.1	Schematic of the interface tracking term in the cost functional. . . .	17
3.1	Exact and approximate representation of a cell cut by the interface. .	32
3.2	Loss of accuracy in mesh-aligned geometries. . . . .	34
3.3	Enhanced geometric information restoring accuracy. . . . .	35
3.4	Intersection of the fluid domain with Cartesian elements. . . . .	38
3.5	Schematic of the first kind capacities. . . . .	39
3.6	Schematic of the second kind capacities. . . . .	40
3.7	Schematic of the different cases in the marching squares algorithm. .	41
3.8	Convergence of the Cut Cell method when solving inside a circle. . .	46
3.9	Convergence of the Cut Cell method when solving outside a circle. .	47
3.10	Convergence of the Cut Cell method when solving inside a square. .	48
3.11	Convergence of the Cut Cell method when solving inside a crystal. .	49
4.1	Schematic of the Johansen-Colella method for the gradient calculation.	54
4.2	Convergence of the gradient computation in a stationary geometry. .	55
4.3	Extension of the velocity field for different narrow band widths. . . .	56
4.4	Schematic of the diamond-cell strategy to compute the Hamiltonian.	58
4.5	Convergence of the Forward-Backward diffusion scheme. . . . .	61
4.6	Reinitialization of an initial perturbed signed distance function. . .	63
4.7	Schematic of the treatment of the fresh and dead cells. . . . .	64
5.1	Convergence of the temperature field in the planar interface case. . .	69
5.2	Convergence of the radius in the Frank sphere case. . . . .	70
5.3	Temperature field error map in the Frank sphere case. . . . .	71
5.4	Convergence of the temperature field in the Frank sphere case. . . .	72
5.5	Numerical instabilities in the absence of surface tension effects. . . .	73
5.6	Regularization introduced by the surface tension effects. . . . .	74
5.7	Effect of the surface tension coefficient on dendritic growth. . . . .	75
5.8	Effect of the grid refinement on dendritic growth. . . . .	76
5.9	Effect of the molecular kinematic coefficient on dendritic growth. . .	77
5.10	Anisotropy effects on crystal growth with $\alpha_0 = \pi/2$ . . . . .	79
5.11	Anisotropy effects on crystal growth with $\alpha_0 = \pi/4$ . . . . .	80

5.12	Effective Rayleigh number for different global Rayleigh numbers. . . . .	82
5.13	Average height for different global Rayleigh numbers. . . . .	83
5.14	Times series of the temperature field with $Ra = 10^6$ . . . . .	84
6.1	Convex cost functional as a function of two parameters $a_1$ and $b_1$ . . . . .	86
6.2	Optimization results and interface shapes in the melting circle case. . . . .	87
6.3	Cost functional variation in the melting circle case. . . . .	88
6.4	Optimization results and interface shapes in the Mullins-Sekerka case. . . . .	89
6.5	Cost functional variation in the Mullins-Sekerka case. . . . .	90
6.6	Optimization results and interface shapes in the growing crystals case. . . . .	91
6.7	Cost functional variation in the growing crystals case. . . . .	92
7.1	Density and viscosity representation in the one-fluid formulation. . . . .	96
8.1	Schematic of the static contact line. . . . .	101
9.1	Schematic of the 2D height-functions near the contact line . . . . .	106
9.2	Schematic of the quadtree discretization and tree representation. . . . .	108
9.3	Schematic of geometrical flux estimation. . . . .	108
9.4	Schematic of the Navier boundary condition. . . . .	109
9.5	Schematic of the curtain coating configuration. . . . .	112
9.6	Time series of a curtain coating simulation with $Re = 12$ , $Ca = 1.33$ . . . . .	112
9.7	Grid resolution near the contact line with $\lambda = 10 \mu m$ . . . . .	114
9.8	Steady state solutions for the three flow configurations. . . . .	114
9.9	Stability analysis of the reduced curtain coating system. . . . .	115
9.10	Convergence of the contact line and inflection point positions. . . . .	116
9.11	Stability analysis and comparison with the Blake experiments. . . . .	118
9.12	Stability analysis and comparison with the Marston experiments. . . . .	118
9.13	Comparison with the Lavi experiments for a spreading drop. . . . .	120
10.1	Radius as a function of time for the super-slip model. . . . .	123
10.2	Local capillary number as a function of time for the super-slip model. . . . .	123
10.3	Contact angle as a function of time for the super-slip model. . . . .	124
10.4	Shear stress as a function of time for the super-slip model. . . . .	124
10.5	Schematic of the angle extrapolation using the GNBC. . . . .	127
10.6	Steady state meniscus for $Ca = 0.03$ using the GNBC. . . . .	128
10.7	Contact line position as a function of time using the GNBC. . . . .	131
10.8	Contact angle as a function of time using the GNBC. . . . .	132
10.9	Shear stress as a function of time using the GNBC. . . . .	133
10.10	Comparison between the no-slip model and the GNBC. . . . .	134
10.11	Schematic of the PF and VOF computational domains. . . . .	140
10.12	Spreading drop diameter comparison of the PF and toy model. . . . .	141
10.13	Spreading drop interfaces of the PF and toy model. . . . .	142
10.14	Zoomed interfaces of the PF and toy model. . . . .	143
10.15	Summary of the different models for the dynamic contact line. . . . .	145

---

D.1	Initialization of a shifted or un-shifted drop with or without contact.	165
D.2	Spurious currents for a static drop with no initial shift. . . . .	166
D.3	Spurious currents for a static drop with a quarter of a cell shift. . . . .	167
D.4	Spurious currents for a static drop with half a cell shift. . . . .	168
E.1	Velocity profiles of a Poiseuille flow with the super-slip model. . . . .	170
E.2	Convergence of the super-slip model with $\lambda_2 = 0$ . . . . .	171
E.3	Convergence of the super-slip model with $\lambda_2 = \Delta^2$ . . . . .	172
F.1	Validation of the angle extrapolation method for the GNBC. . . . .	175
G.1	Total mass of the drop with or without sink velocity . . . . .	178
G.2	Comparison of the interfaces for different computational boundaries. . . . .	179

# Introduction

The study of fluid flows has been a lasting human endeavor, pursued to gain a further understanding of Nature and industrial processes. From a mathematical point of view, these systems are modeled by partial differential equations where the ‘unknowns’ are both space and time dependent. Due to the highly complex nature of the phenomena arising from these flows – even in seemingly simple systems – these equations are practically unsolvable by analytical techniques. Numerical analysis, however, is an approach that is well suited to study these systems, by using discrete approximations of the partial differential equations. In the context of multiphase flows – as subset of fluid flows corresponding to the simultaneous flow of materials with two or more thermodynamic phases – the main difficulty comes from the high non-linearity introduced by the interface and specific numerical methods are required to track the interface with accuracy.

In this work we will study different physical systems using several original numerical methods and models. In the first part, we will consider phase-change problems – so-called Stefan problems – involving thermal energy exchanges between a solid and a liquid phase. In the second part, we will investigate problems involving dynamic contact lines, where a fluid-fluid interface moves over a solid substrate.

**Two-phase Stefan problems** Stefan problems, named after the Austrian mathematician Joseph Stefan [Stefan 1891, Šarler 1995] due to his substantial contribution to research on moving and free boundaries, model transport and transfer phenomena, in particular solid-liquid phase change in evaporating or chemically reacting flows. Such phenomena govern the interface motion in many engineering related problems such as dendritic solidification [Osher & Sethian 1988, Juric & Tryggvason 1996], phase transformation in metallic alloys [Segal *et al.* 1998], and solid fuel combustion [Hassan *et al.* 2021].

In the applications of interest to this study, the Stefan condition arises from the interaction of liquid and solid phases – both considered incompressible – resulting in a moving liquid-solid interface, typically freezing or melting front. The speed of the front is directly related to the jump in the conductive heat flux across the interface.

In one dimension, this problem has been studied in depth using various numerical algorithms [Brattkus & Meiron 1992, Rose 1993, Javierre *et al.* 2006]. In higher dimensions, various methods have been used such as the level set method [Osher & Sethian 1988, Limare *et al.* 2023] and front-tracking method [Juric & Tryggvason 1996]. One of the main challenges associated with modeling such problems in multiple dimensions is due to the unstable dendritic pattern formation [Mullins & Sekerka 1964, Langer 1980, Woods 1992]. In crystal growth, for example, under-cooling triggers an instability mechanism, causing the solid phase of the material to grow into the liquid phase in a finger-like or dendritic fashion, resulting in complex interfacial shapes, which are challenging to predict

numerically. In addition, parameters such as anisotropy and surface tension or curvature effect (Gibbs-Thomson) are shown to have a large impact on the dendritic shape of the crystal, which in turn need to be modeled accurately for the simulations to remain predictive. In this work, we present a general framework for tracking and modeling crystal growth in the presence of curvature effects. This algorithm then serves as a vessel to materialize the second, and main, objective of this study which is extracting optimization strategies to control the resulting solidification process.

The shape of the interface strongly effects the outcome and time-frame of the production process in many industrial applications involving phase change. As a result, while predicting and modeling the resulting solidification process remains at the forefront of many research areas, it is as desirable to extract efficient control strategies to manipulate the motion of the interface, for instance, by tracking a prescribed trajectory.

Two major types of optimization methods in use today are (i) gradient-based, and (ii) derivative free methods. While an efficient class of generic algorithms – belonging to the class of derivative free methods – based on the surrogate management framework [Marsden *et al.* 2008] and artificial neural networks [Pierret *et al.* 2007] have been used for optimization in fluid mechanics, mainly in the area of aerodynamic shape optimization, these often require many function evaluations, for training purposes for example. When detailed simulations of interfacial flows are concerned, each function evaluation commands a full – potentially unsteady – Computational Fluid Dynamics (CFD) simulation, causing gradient-based methods to be at an advantage.

Most common methods in extracting the gradient information, on the other hand, are analytical or use finite differences, neither of which are suited to the configuration of interest to this work. Adjoint-based algorithms present a suitable alternative, as they allow the determination of the gradient at a cost comparable to a single function evaluation [Giles & Pierce 2000]. The use of adjoint methods for design and optimization has been an active area of research which started with the pioneering work of Pironneau [Pironneau 1974] with applications in fluid mechanics, and has been extensively used in aeronautical shape optimization by Jameson and co-workers [Jameson 1988, Jameson *et al.* 1998]. Ever since these groundbreaking studies, adjoint-based methods have been widely used in fluid mechanics particularly in the areas of aero- and thermo-acoustics [Juniper 2010, Lemke *et al.* 2013].

More recently flow regimes dominated by nonlinear dynamics, such as separation and mixing have also been analysed using adjoint-based techniques [Duraismy & Alonso 2012, Schmidt *et al.* 2013, Foures *et al.* 2014, Rabin *et al.* 2014]. Adjoint-based methods have also been employed for the purpose of sensitivity analysis or control in flows in the presence of large gradients [Ou & Jameson 2011, Braman *et al.* 2015, Lemke *et al.* 2019, Fikl *et al.* 2020] – for example in flames or interfaces – showing great promise, and therefore are adopted here to carry out the optimization procedure.

In the context of Stefan problems, various control strategies have been employed to track the location of the interface. In a one-dimensional setting, for

example, set-valued fixed point equations [Hoffmann & Sprekels 1982] or linear-quadratic defect minimization [Knabner 1985] have been used to control the location of the front. Adjoint-based algorithms have also been applied to a Stefan problem caused by heterogeneous reactions on a surface of a one-dimensional solid particle [Hassan *et al.* 2021] to extract sensitivities with respect to various kinetic parameters. Alternatively, in a two-dimensional setting, adjoint-based algorithms have been utilized previously together with finite element and finite difference approaches to track and control the location of interface by imposing heat flux (or temperature) at the boundary in order to realize the desired interface motion [Kang & Zabarav 1995, Yang 1997, Hinze & Ziegenbalg 2007]. In particular, Bernauer and Herzog [Bernauer & Herzog 2011], making use of shape calculus tools, derived the set of adjoint equations to extract control strategies for Stefan problems with a sharp representation of the interface.

Similar to the approach of [Bernauer & Herzog 2011] shape calculus tools have been employed in this work to extract the corresponding adjoint equations. However, contrary to the previous studies, control strategies are extracted here to suppress instabilities – as dendritic formation – of the solidification process. The addition of curvature effects on the interface – Gibbs-Thomson relation – and the complex shapes encountered during the growth of the crystal require dedicated numerical algorithms capable of modeling both the highly nonlinear forward problem – correct representation of the resulting interface – and the resulting adjoint equations. In addition, while previous studies mostly concentrated on actuation by imposing temperature or heat flux at the boundaries, alternative control strategies using the length of interface are also investigated here, to identify the most relevant and effective numerical control strategies in the context of melting and solidification processes [Fullana *et al.* 2023].

As an extension of the classical two-phase Stefan problem, we will consider the convective motion induced by a flow in the fluid phase [Quirós Rodríguez *et al.* 2022] in a different setting than the adjoint-based optimization framework defined previously. In that case, the Navier-Stokes equations in the Boussinesq approximation – defined later – will be solved in order to model the buoyancy phenomenon, where the fluid exerts a force that opposes the weight of a solid phase. In particular, we will study the melting process of a solid where an initial fluid layer is heated from below. This motion will entrain the onset of Rayleigh-Bénard instability – where convection cells appear depending on the wave length of the layer – and a parametric study on the effect of the Rayleigh number, similar to the ones of Favier [Favier *et al.* 2019] and Limare [Limare *et al.* 2023], will be carried out.

**Dynamic contact lines** The contact line formed at the intersection of a fluid-fluid and a solid boundary – so-called the triple point – is a key element of many natural and several technological processes, as described in [Blake & Shikhmurzaev 2002, Wilson *et al.* 2006, Marston *et al.* 2009]. The dynamic contact line poses – since the early work of Huh and Scriven [Huh & Scriven 1971] – a remarkable problem

because of the contradiction between the no-slip condition on the substrate and the motion of the contact line. A fixed contact line, for example, is in contradiction with simple observations of a spreading drop on a surface. Mathematically, this paradox introduces a stress singularity at the contact line and the numerical models to properly predict the motion of the triple point are still debated to this day [Cox 1986, Shikhmurzaev 1994, Blake *et al.* 1999, Eggers 2004, Eggers & Evans 2005].

In the context of continuum mechanics, in sharp-interface models, the simplest way to relax the singularity is to introduce a Navier boundary condition (NBC) – by allowing the contact line to slide over the substrate – characterized by a slip length of the order of the molecular length. This model has been proven to accurately reproduce the motion of the contact line in various physical setups by either fitting the slip length with molecular dynamics results [Lācis *et al.* 2020, Lācis *et al.* 2022] or choosing an appropriate slip related to experimental observations [Fullana *et al.* 2020].

Another essential parameter to consider is the variation of the angle formed by the fluid-fluid interface with the solid – the dynamic contact angle – as the contact line moves. In the classical Cox-Voinov relation [Voinov 1976, Cox 1986], the dynamic angle is a function of the equilibrium angle – a property of the substrate – and the contact line speed. Inspired by the work of Legendre and Maglio [Legendre & Maglio 2015], where they tested different dynamic angle models, we will implement and benchmark them against experiments of a spreading squalane drop [Lavi & Marmur 2004].

In an attempt to further understand the physics involved in this problem, we develop new and original numerical methods for dynamic contact lines. The two previously described models – NBC and dynamic angle, called classical models in this dissertation – will serve as a basis for the implementation of the advanced ones.

In the super-slip model, a further regularization of the contact line is provided through a second-order slip length. This model, in analogy with porous media theory [Devauchelle *et al.* 2007], allows a relaxation of the capillary pressure present in the NBC. A study on the effect of the second-order slip in the case of a spreading drop will be carried out.

The second advanced model is the generalized Navier boundary condition (GNBC). By considering the effect of the uncompensated Young’s stress we allow an angle deviation with respect to the equilibrium angle. This boundary condition, originally derived for diffuse interface models [Qian *et al.* 2003b, Qian *et al.* 2006] has been proven to be well-posed in the sharp interface limit [Fricke 2020]. The GNBC will be implemented in our Volume-Of-Fluid framework and we will study its effect on the onset of wetting failure – where a meniscus transitions to a liquid film – in a withdrawing plate setup.

Finally, we will use a toy model to establish a relation between the mass flux present in the Phase-Field (PF) model [Jacqmin 2000, Amberg 2003, Carlson 2012] to a curvature relation at a given distance of the contact line. Through simple boundary conditions in VOF model, we will be able to reproduce the diffusion-driven wetting motion existing in the PF, at a much lower cost.

**Original contributions in this work** One of the major contributions of our work is the derivation of the continuous adjoint two-phase Stefan problem in the presence of curvature effects and the construction of the corresponding adjoint-based algorithm. The numerical methods described in Chapters 3 and 4 constitute a novelty with respect to the existing ones, in particular, the Cut Cell method for diffusive transport. The aforementioned methods and test cases are coded in *Flower.jl* (see Chapter 5 and Appendix C for further details), a new Julia package developed as part of this work. All the contact line models – NBC, dynamic contact angle, super-slip, GNBC, toy model – are implemented in *Basilisk*, an existing platform for the solution of partial differential equations (presented in Chapter 9).

**Outline** The first part of this work will be focused on the optimization of two-phase Stefan problems. In Chapter 1, the governing equations in continuous form in a level set framework will be described. Additionally, the fluid flow extension will be described. In Chapter 2, we present the derivation of the continuous adjoint of the Stefan problem in presence of surface tension effects and the resulting adjoint-based optimization algorithm. In Chapter 3, the newly developed Cut Cell method for diffusive transport and its numerical implementation will be presented in depth. Chapter 4 presents the rest of the numerical methods used to solve both the forward and adjoint Stefan problems. In Chapter 5, a numerical validation of our scheme against analytical solutions will be provided. Moreover, we will carry out studies on dendritic crystal grows and on the onset of Rayleigh-Bénard convection cells with a melting boundary. Finally, in Chapter 6, we will present shape optimization cases – with variable complexity – of melting and solidification processes and compare the results obtained with the adjoint-based algorithm with a derivative-free one.

In the second part, focus will be set on the dynamic contact line. In Chapter 7, the one-fluid formulation of the two-phase Navier-Stokes equations and the Phase-Field extensions will be described. Chapter 8 discusses the paradoxes and singularities present in dynamic contact line problems. In Chapter 9, the Volume-Of-Fluid method and the implementation of the classical models for contact lines are described. These models will be tested on different physical systems. Finally, in Chapter 10, the newly developed advanced models for contact lines will be detailed and, again, tested on various physical problems.

## Part I

# Shape optimization of melting and solidification processes

# The two-phase Stefan problem

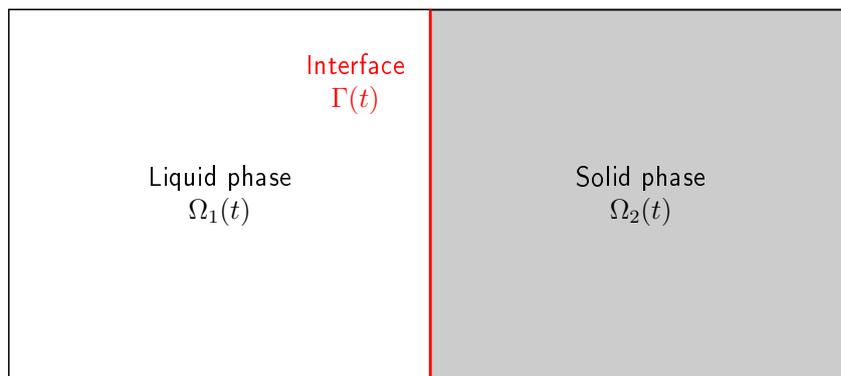
## Contents

1.1	Classical two-phase Stefan problem formulation . . . . .	7
1.2	The level set method . . . . .	9
1.3	Thermal convection and fluid flow extension . . . . .	12

## 1.1 Classical two-phase Stefan problem formulation

A Stefan problem is a specific type of ‘free boundary problem’ characterized by the temperature distribution of two different phases. The position of the interface between those two phases is *a priori* unknown and will be part of the solution of partial differential equations describing this system. As a first approximation, we will consider the two-phase Stefan problem in the presence of two immiscible phases with matching densities (one liquid and the other solid).

We define a domain  $\Omega$  partitioned into two time-dependent subdomains  $\Omega_1(t)$  and  $\Omega_2(t)$  occupied by the liquid (1) and solid (2) phases, respectively. The external boundary of the domain, denoted  $\partial\Omega$ , is fixed whereas the interface separating both phases  $\Gamma(t) = \overline{\Omega}_1(t) \cap \overline{\Omega}_2(t)$  evolves in time. A schematic of this configuration is shown in Figure 1.1.



**Figure 1.1:** Schematic of the liquid and solid subdomains considered in the Stefan problem.

The energy transport mechanism in each phase is the heat produced by a difference in temperature. The partial differential equation governing this quantity is the heat equation. Let  $T_i: (\mathbf{x}, t) \rightarrow \mathbb{R}^+$  denote the temperature field in either

phase ( $i = 1, 2$ ), where  $\mathbf{x} = (x, y)$  is the Cartesian coordinate vector. When the densities  $\rho_1$  and  $\rho_2$  are equal, convective heat transfer vanishes. In addition, when the background pressure is assumed constant, the heat transport equation simplifies to

$$\forall i \in \llbracket 1, 2 \rrbracket, \quad \rho_i c_i \frac{\partial T_i}{\partial t} = \nabla \cdot (k_i \nabla T_i), \quad t > 0, \quad \mathbf{x} \in \Omega_i, \quad (1.1)$$

where  $\rho_i$ ,  $c_i$  and  $k_i$  denote the density, the specific heat capacity at constant pressure, and the thermal conductivity, for each phase.

As the phase change phenomena occurs, there will appear a latent heat which either is absorbed or released. The condition of heat conservation at a given point on the moving interface corresponds to the rate at which heat is generated at the boundary balanced by the rate at which this heat flows in either phase. Therefore, along the interface, energy balance states that

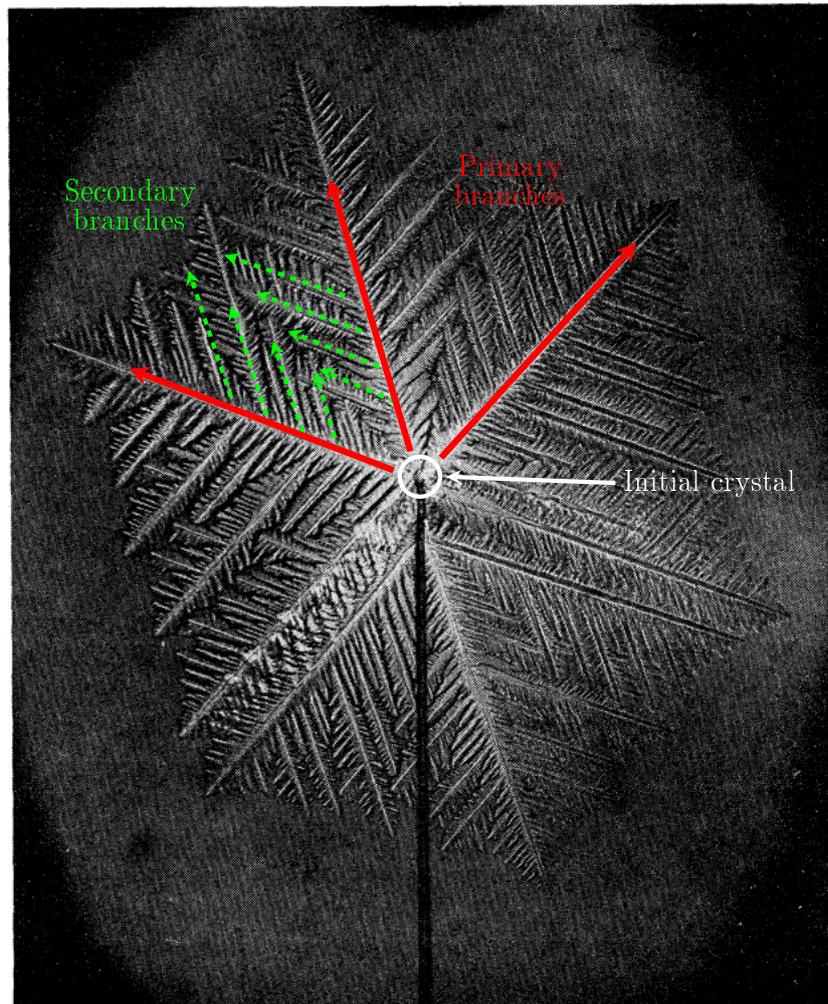
$$v_S = -L_H^{-1} \left( k_1 \frac{\partial T_1}{\partial \mathbf{n}} - k_2 \frac{\partial T_2}{\partial \mathbf{n}} \right), \quad \mathbf{x} \in \Gamma, \quad (1.2)$$

where  $L_H$  denotes the latent heat of solidification and  $\mathbf{n}$  is the outward normal unit vector at the interface. This jump is taken from phase 1 to phase 2 with  $\partial T_i / \partial \mathbf{n}$  denoting the normal component of the temperature gradient in phase  $i$ . Equation 1.2 is commonly referred to as the Stefan condition.

In classical Stefan problem, the temperature is set to  $T_D(\mathbf{x}, t) = T_M$  at the interface where  $T_M$  is a constant equal to the melting temperature of the material. For problems involving crystal growth however [Langer 1980], surface tension effects must be added to the thermodynamic boundary condition by introducing a dependence in the curvature at the front. For example, in the formation of crystals (Figure 1.2), a reduction of the melting temperature for small particles through the surface tension effects provides the necessary stabilizing forces allowing the creation of patterns. Moreover, from molecular kinetic theories, we know that the melting temperature might also depend on the speed at which the interface moves. For that purpose, we use the classical Gibbs-Thomson relation, as defined in Chen [Chen *et al.* 1997]

$$T_D(\mathbf{x}, t) = T_M - \varepsilon_V v_S - \varepsilon_\kappa \kappa, \quad \mathbf{x} \in \Gamma, \quad (1.3)$$

where  $\kappa$  denotes the curvature at the interface (positive if the center of curvature lies in the solid phase),  $v_S$  the velocity of the interface,  $\varepsilon_\kappa$  the surface tension coefficient, and  $\varepsilon_V$  the molecular kinetic coefficient. Unless stated otherwise, in the rest of this dissertation, both  $\varepsilon_\kappa$  and  $\varepsilon_V$  are considered to be constants, and the heat capacities, thermal conductivities and latent heat are all set to unity. In addition, to ease the notation, the jump in gradient of temperature is denoted as  $[\nabla T]_2^1$  (also defined in Equation 1.2). This set of equations characterize the two-phase Stefan problem. When solving the system numerical, the main difficulty arises from the treatment of the interface due to the non-linear relation between the Stefan condition and the two domains where the heat equations are solved.



**Figure 1.2:** Photography of a crystal growing in an under-cooled liquid bath exhibiting a pattern formation with primary and secondary branches [Fujioka 1978].

## 1.2 The level set method

When dealing with the numerical approximation of interfacial flows, two classes of methods are commonly used to represent the interface, namely

- Lagrangian or ‘front-tracking’ methods,
- Eulerian or ‘front-capturing’ methods.

The former uses a parameterisation of the interface location (e.g. markers or moving meshes), and has already been used in Stefan problems [Juric & Tryggvason 1996] but has not been adopted in this work due to the inherent difficulty of deriving the continuous adjoint equations with such methods (see Chapter 2). The latter can broadly speaking be divided into two categories: Volume-Of-Fluid (VOF) and

Level Set methods. The VOF method will be described in the second part of this dissertation when dealing with moving contact lines (see Chapter 9).

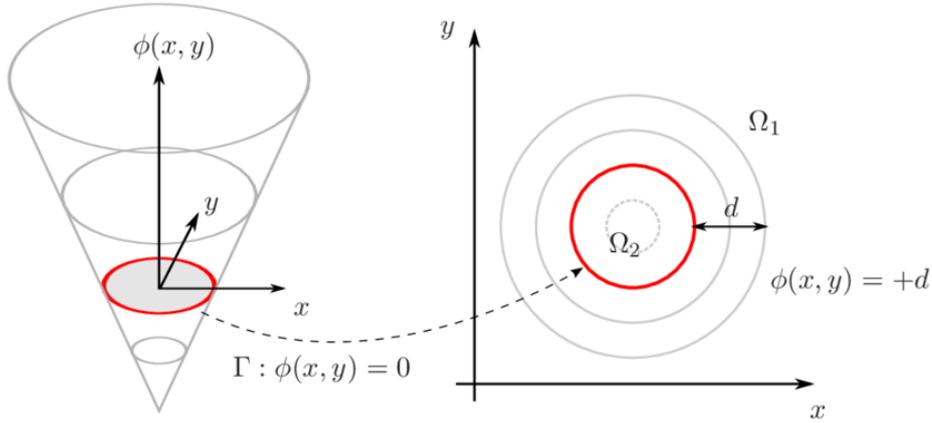
In adjoint-based optimization, the VOF method [Fikl *et al.* 2020] may lead to numerical complications due to the piece-wise linear reconstruction of the interface. On the other hand, the level set method, where the interface is implicitly defined as a continuous function, has been proven to be well suited in continuous adjoint-based optimization, specifically for Stefan problems [Bernauer & Herzog 2011]. Moreover, this method has been shown to accurately reproduce dendritic pattern formation [Chen *et al.* 1997, Limare *et al.* 2023].

An implicit signed distance strategy has therefore been used here to represent the interface. A level set function  $\phi: (\mathbf{x}, t) \rightarrow \mathbb{R}$  is constructed, such that, at any time  $t$ , the front is equal to the zero level set of the function

$$\Gamma(t) = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}, t) = 0\}. \quad (1.4)$$

The level set function is initially set to the signed distance function, with  $d$  the distance to the front, such that

$$\phi(\mathbf{x}, 0) = \begin{cases} +d, & \mathbf{x} \in \Omega_2, \\ 0, & \mathbf{x} \in \Gamma, \\ -d, & \mathbf{x} \in \Omega_1. \end{cases} \quad (1.5)$$



**Figure 1.3:** View of the level set signed distance function  $\phi$ . The interface corresponds to the interface of  $\phi$  and the 0-level plane.

In practice, the level set function is a function of dimension  $D + 1$ , where  $D$  is the dimension of the problem considered ( $D = 2$  in this case). The intersection between  $\phi$  and the 0-level plane corresponds to the interface (Figure 1.3). One of the advantages of this method is the direct definition of the normal vector and the curvature, quantities that are necessary to compute the Stefan condition and the

Gibbs-Thomson relation. The normal unit vector  $\mathbf{n}$  is defined by

$$\mathbf{n} = \frac{\nabla\phi(\mathbf{x}, t)}{|\nabla\phi(\mathbf{x}, t)|}, \quad \mathbf{x} \in \Gamma(t), \quad (1.6)$$

and the curvature  $\kappa$  by

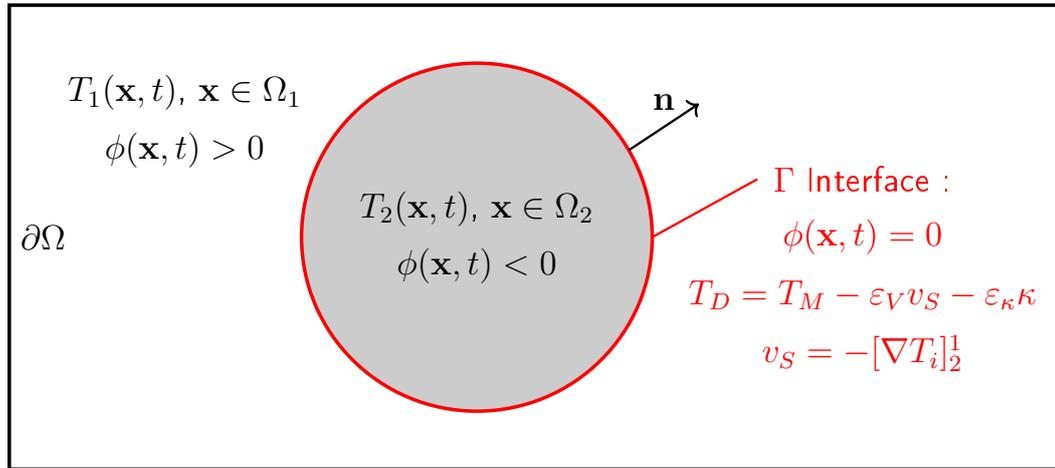
$$\kappa = \nabla \cdot \left( \frac{\nabla\phi(\mathbf{x}, t)}{|\nabla\phi(\mathbf{x}, t)|} \right), \quad \mathbf{x} \in \Gamma(t). \quad (1.7)$$

In two dimensions, the curvature  $\kappa$  at the front is computed in a non-conservative form using

$$\kappa = \frac{(\phi_y^2 \phi_{xx} - 2\phi_x \phi_y \phi_{xy} + \phi_x^2 \phi_{yy})}{(\phi_x^2 + \phi_y^2)^{3/2}}, \quad (1.8)$$

where  $\phi_x, \phi_{xx}, \phi_y, \phi_{yy}$  denote the first and second derivative of  $\phi$  with respect to  $x$  and  $y$ . The interface is moved through the level set advection equation, where  $F$  is a speed function defined everywhere

$$\frac{\partial\phi}{\partial t} + F|\nabla\phi| = 0. \quad (1.9)$$



**Figure 1.4:** Schematic of the two-phase Stefan problem in the level set framework.

In the Stefan problem, the velocity is only defined at the interface (Equation 1.2). The velocity is numerically extended in the normal direction of the interface, around a ‘narrow band’ (defined in Chapter 4). By combining Equations 1.2 and 1.9 and using the normal definition (Equation 1.6), we obtain the following equation of motion

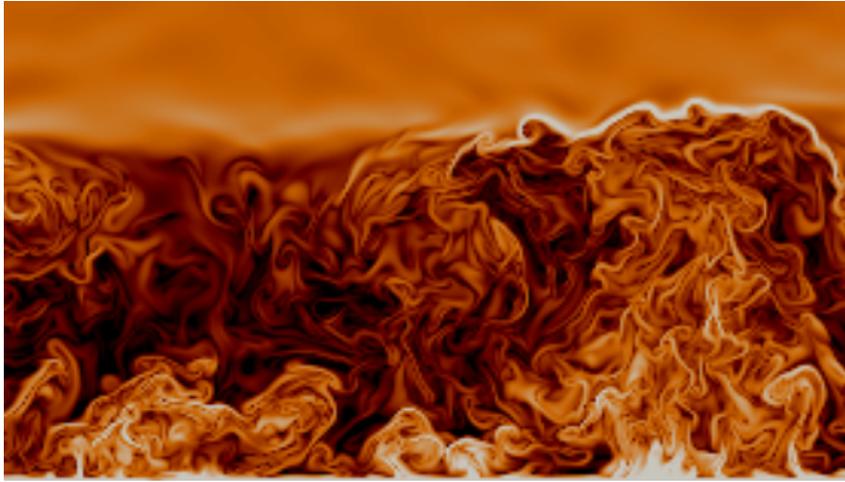
$$\frac{\partial\phi}{\partial t} = -[\nabla T]_2^1 \cdot \nabla\phi. \quad (1.10)$$

Equation (1.10) will move  $\phi$  with the speed such that  $\Gamma$  remains close to the zero level set of  $\phi$ . In practice, and this is one of the main drawbacks of this method, the level

set function needs to be reinitialized such that it retains its signed distance function properties. The resulting two-phase Stefan problem in the level set framework is described in Figure 1.4.

### 1.3 Thermal convection and fluid flow extension

The model defined so far assumes that the flow in the fluid phase can be neglected. This is an appropriate hypothesis when considering equal densities in the fluid and solid phases. However, in some applications, this assumption is too restrictive. In Figure 1.5, an example of Rayleigh-Bénard convection cells appearing in a layer moving upwards is shown.

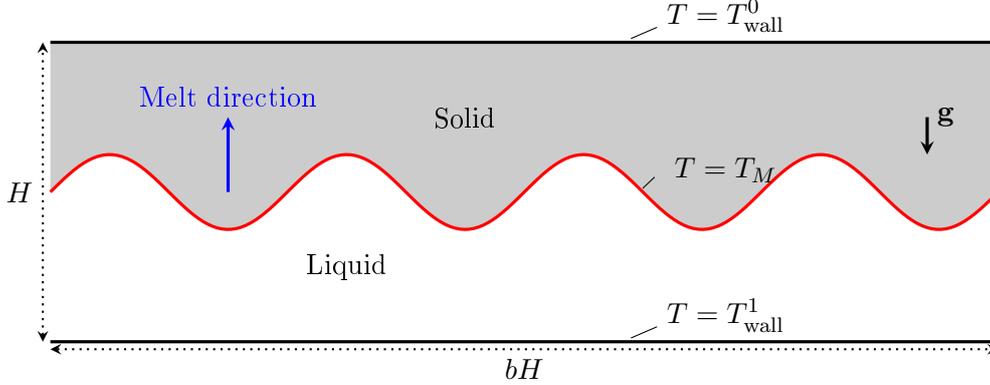


**Figure 1.5:** Photography of an atmospheric surface layer [Pandey *et al.* 2018].

To model such phenomena, one has to take into account both convection and the Navier-Stokes equations in the Boussinesq approximation in the fluid phase. In this work, we will focus on the same setup as [Favier *et al.* 2019], where they consider the evolution of a horizontal layer heated from below and comprised between two walls at a distance  $H$  apart. The gravity is pointing downwards  $\mathbf{g} = -g\mathbf{e}_y$  and the horizontal size of the domain is  $bH$  where  $b$  is the aspect ratio. The temperature at the bottom wall is  $T = T_{\text{wall}}^1$  and the one at the top wall is  $T = T_{\text{wall}}^0$  such that  $T_{\text{wall}}^0 < T_M < T_{\text{wall}}^1$  where  $T_M$  is the melting temperature. A schematic of this configuration is shown in Figure 1.6. The governing dimensionless equations in the fluid phase are now

$$\begin{aligned} \text{Pr}^{-1} \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) &= -\nabla P + \text{Ra} T \mathbf{e}_y + \nabla^2 \mathbf{u}, \\ \nabla \cdot \mathbf{u} &= 0, \\ \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T &= \nabla^2 T, \end{aligned} \tag{1.11}$$

where  $\mathbf{u}$  is the liquid velocity  $\mathbf{u} = (u_x, u_y)$ ,  $P$  is the pressure and  $T$  is the temperature in liquid phase.



**Figure 1.6:** Schematic of the melting boundary problem.

The dimensionless numbers governing this equation are the Prandlt number

$$\text{Pr} = \frac{\nu}{k_1}, \quad (1.12)$$

defined by the ratio of the liquid kinematic viscosity  $\nu$  and thermal diffusivity in the liquid phase  $k_1$ , and the Rayleigh number

$$\text{Ra} = \frac{g\alpha_t\Delta TH^3}{\nu k_1}, \quad (1.13)$$

where  $g$  is the constant gravitational acceleration,  $\alpha_t$  the coefficient of thermal expansion and  $\Delta T = T_{\text{wall}}^1 - T_{\text{wall}}^0$  the difference in temperature between the two plates. In the solid phase, when considering the non-isothermal general case, with  $T_{\text{wall}}^0 \neq T_M$ , the heat equation applies

$$\frac{\partial T}{\partial t} = \nabla^2 T. \quad (1.14)$$

In this configuration, we will be able to study the onset of the Rayleigh convection cells and the changes in topology as the interface melts through the solid (Section 5.4). In the next chapter, we present the optimization procedure involving the derivation of the continuous adjoint of the two-phase Stefan problem in the absence of fluid flow.



# Adjoint-based optimization algorithm

---

## Contents

---

<b>2.1</b>	<b>Minimization problem</b>	<b>15</b>
<b>2.2</b>	<b>Continuous adjoint derivation</b>	<b>18</b>
2.2.1	Lagrange functional	18
2.2.2	Adjoint temperature	19
2.2.3	Adjoint level set	22
2.2.4	The gradient equation	25
<b>2.3</b>	<b>Gradient-based optimization procedure</b>	<b>26</b>

---

## 2.1 Minimization problem

This chapter presents the optimization procedure for the two-phase Stefan problem described in Section 1.1 together with the respective cost functional and the corresponding adjoint equations. The control variable is a Neumann boundary condition that acts on the domain boundary. The adjoint heat equations and adjoint level set equations are derived in the same fashion as [Bernauer & Herzog 2011]. The resulting gradient-based optimization algorithm, as well as the derivative-free methods, used throughout the various cases considered in this study are described. By considering the two-phase Stefan problem with the Gibbs-Thomson relation, the generic forward two-phase Stefan problem (FP) can be recast in the level set framework given below, which is later used in the adjoint derivation.

Find a function  $T : \Omega \times [0, t_f] \rightarrow \mathbb{R}$  and a function  $\phi : \Omega \times [0, t_f] \rightarrow \mathbb{R}$  such that

$$\left\{ \begin{array}{ll} \frac{\partial T_1}{\partial t} = \nabla^2 T_1 & \text{in } \Omega_1(t) \text{ (FP.a)} \\ \frac{\partial T_2}{\partial t} = \nabla^2 T_2 & \text{in } \Omega_2(t) \text{ (FP.b)} \\ T(x, 0) = T_0(x) & \text{in } \Omega \text{ (FP.c)} \\ \frac{\partial T(x, t)}{\partial n} = w(x, t) & \text{on } \partial\Omega \text{ (FP.d)} \\ T(x, t) = T_M - \varepsilon_V v_S - \varepsilon_\kappa \kappa & \text{on } \Gamma(t) \text{ (FP.e)} \\ \frac{\partial \phi}{\partial t} = -[\nabla T_i]_2^1 \cdot \nabla \phi & \text{on } \Gamma(t) \text{ (FP.f)} \\ \phi(x, 0) = \phi_0(x) & \text{in } \Omega \text{ (FP.g)} \end{array} \right. \quad (\text{FP})$$

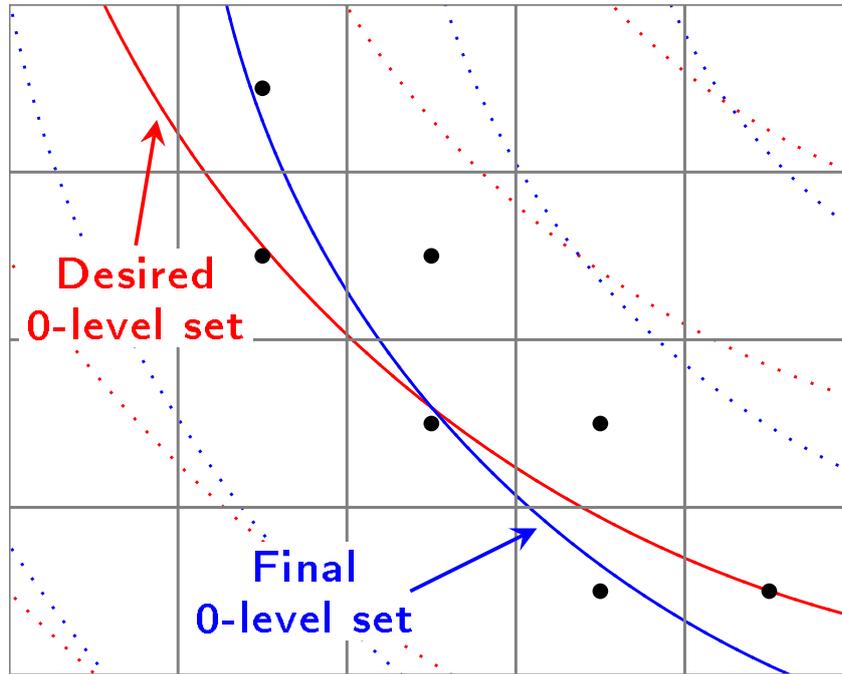
Here,  $T_i$  denotes the restrictions of  $T$  to  $\Omega_i(t)$ , and  $T_0$  and  $\phi_0$  are the initial conditions at  $t = 0$  of the temperature field and the level set function. The first two equations FP.a and FP.b correspond to the heat equations in phases 1 and 2 respectively. Equations FP.c and FP.g correspond to the initial temperature distribution in the whole domain and the initial position of the front respectively. The equation FP.d correspond to the control variable  $w$ , the heat flux acting on the domain boundary. Equation FP.e is the Dirichlet boundary condition at the interface with the Gibbs-Thomson relation taken into account. Finally, FP.f corresponds to the advection equation of the level function as defined in Equation 1.10 where  $-\left[\nabla T_i\right]_2^1$  is the Stefan condition.

The next step is to define the desired temperature field  $T^d$  and the desired position of the interface  $\phi^d$ . The control variable  $w$  is then defined through the optimization procedure to reach these desired quantities.

These conditions are known a priori and will drive our cost functional towards its minimum. The following tracking-type cost functional provides a mathematical description of the control goals stated above

$$\begin{aligned} \mathcal{J}(T, \phi, w) = & \frac{\beta_1}{2} \int_{\Omega} |T^f - T^d|^2 dx + \frac{\beta_2}{2} \int_{\Gamma^f} |\phi^f - \phi^d|^2 ds \\ & + \frac{\beta_3}{2} \int_{\Gamma^f} ds + \frac{\beta_4}{2} \int_0^{t_f} \int_{\partial\Omega} |w|^2 ds dt, \end{aligned}$$

with constants  $\beta_1$  to  $\beta_4$  acting as weights,  $T^f$  and  $\phi^f$  the final temperature field and level set function of the FP and  $\Gamma^f$  the final position of the interface (0-level set of  $\phi^f$ ).



**Figure 2.1:** Schematic of the computation of the second term of the cost functional. The values of  $\phi^f$  and  $\phi^d$  are computed in the mixed cells (black dots) determined by the 0-level set of the final level set function. The dotted lines represent the -2,-1,1,2 level sets of both functions.

- The first term monitors the final temperature distribution and is mostly used as an initializer for the adjoint temperature field for the adjoint problem (see Section 2.2).
- The second term controls the relative position of the level set function with respect to the desired one. Taking advantage of the signed distance function property of both  $\phi^d$  and  $\phi^f$ , the discrete form is simply the square of the difference between both functions computed in the points belonging to the final interface  $\Gamma^f$ . An example of the computation is shown in Figure 2.1. The weight  $\beta_2$  associated to the interface position is always larger than  $\beta_1$  as the position of the interface dictates not only the shape itself but also the location of both domains  $\Omega_1$  and  $\Omega_2$  thus influencing directly the temperature distribution.
- The third term controls the length of the interface. It is used in cases where instabilities cause an increase in interface length, for example when controlling the Mullins-Sekerka instability (see Chapter 6).
- Finally, the last term penalizes the control cost and leads to the gradient equation in the adjoint problem.

Algorithm 1 summarizes the steps needed to compute  $\mathcal{J}$ . We assume that each choice of the control variable  $w$  leads to unique states  $T(w)$  and  $\phi(w)$ . Therefore, the minimization problem (MP) reads

$$\begin{aligned} \min_w \mathcal{J}(T(w), \phi(w), w) \\ \text{subject to (FP).} \end{aligned} \tag{MP}$$

---

**Algorithm 1:** Cost functional computation
 

---

**input :**  $u, T^f, T^d, \phi^f, \phi^d, \beta_1, \beta_2, \beta_3, \beta_4, t_f$   
**output:**  $\mathcal{J}$

$\mathcal{J} = 0, X$

**for**  $x \in \partial\Omega$  **do**

| **for**  $t \in [0, t_f]$  **do**

| |  $\mathcal{J} += \beta_4 w(x, t)^2$

| **end**

**end**

**for**  $x \in \Omega$  **do**

|  $\mathcal{J} += \beta_1 (T^f(x)^2 - T^d(x)^2)$

| **if**  $x \in \Gamma^f$  **then**

| |  $\mathcal{J} += \beta_2 (\phi^f(x)^2 - \phi^d(x)^2)$

| | Store the interface centroid position in  $X$

| **end**

**end**

Compute the total length of the interface  $L = \text{arclength}(X)$

$\mathcal{J} += \beta_3 L$

---

## 2.2 Continuous adjoint derivation

### 2.2.1 Lagrange functional

In this work we chose the "differentiate-then-discretize" approach to solve the minimization problem (MP). We derive the corresponding adjoint problem in a continuous way and discretize afterwards. Let  $\Theta$  be the adjoint temperature and  $\psi$  the adjoint level set function. In order to compute the gradient of  $\mathcal{J}(w)$ , we introduce

the following Lagrange functional  $\mathcal{L}$  ( $dx$ ,  $dt$  and  $ds$  are omitted for brevity)

$$\begin{aligned}
\mathcal{L}(T, \phi, w, \Theta, \Theta_D, \Theta_I, \psi) = & \mathcal{J}(T, \phi, w) \\
& - \int_0^{t_f} \int_{\Omega_1} \left( \frac{\partial T_1}{\partial t} - \nabla^2 T_1 \right) \Theta_1 \\
& - \int_0^{t_f} \int_{\Omega_2} \left( \frac{\partial T_2}{\partial t} - \nabla^2 T_2 \right) \Theta_2 \\
& - \int_0^{t_f} \int_{\partial\Omega} (T - w) \Theta_D \\
& - \int_0^{t_f} \int_{\Gamma} (T - (T_M - \varepsilon_V v_S - \varepsilon_\kappa \kappa)) \Theta_I \\
& - \int_0^{t_f} \int_{\Gamma} \left( \frac{\partial \phi}{\partial t} + [\nabla T_i]_2^1 \cdot \nabla \phi \right) \psi.
\end{aligned} \tag{2.1}$$

Similarly to the temperature field,  $\Theta_i: (x, t) \rightarrow \mathbb{R}^+$  denotes the adjoint temperature field in either phase ( $i = 1, 2$ ). The Lagrange multipliers  $\Theta_D$  and  $\Theta_I$  are used for the boundary conditions on the domain and the interface respectively.

The adjoint system is obtained by setting to zero the derivatives of the Lagrange functional with respect to  $T$  and  $\phi$ :  $\mathcal{L}_T(\cdot) = \mathcal{L}_\phi(\cdot) = 0$ . Setting the initial conditions to

$$\begin{aligned}
T(x, 0) &= T_0(x), \\
\phi(x, 0) &= \phi_0(x),
\end{aligned} \tag{2.2}$$

leads to  $\delta T(x, 0) = \delta \phi(x, 0) = 0$  when calculating the derivatives in the direction  $\delta T$  and  $\delta \phi$ . In the following sections, we detail the procedure of extracting the adjoint Stefan problem (AP) using the Lagrange function  $\mathcal{L}$  (Equation 2.1).

### 2.2.2 Adjoint temperature

We start by deriving the adjoint temperature equations by setting  $\mathcal{L}_T(\cdot) = 0$

$$\begin{aligned}
\mathcal{L}_T \delta T = & - \int_{\Omega} \beta_1 (T^f - T^d) \delta T^f \\
& - \int_0^{t_f} \int_{\Omega_1} \left( \frac{\partial \delta T_1}{\partial t} - \nabla^2 \delta T_1 \right) \Theta_1 \\
& - \int_0^{t_f} \int_{\Omega_2} \left( \frac{\partial \delta T_2}{\partial t} - \nabla^2 \delta T_2 \right) \Theta_2 \\
& - \int_0^{t_f} \int_{\partial\Omega} \delta T \Theta_D \\
& - \int_0^{t_f} \int_{\Gamma} \delta T \Theta_I \\
& - \int_0^{t_f} \int_{\Gamma} \psi [\nabla \delta T_i]_2^1 \cdot \nabla \phi.
\end{aligned}$$

As expected, the terms in the Lagrangian that do not depend on the temperature vanish. We now move the spatial and temporal derivatives towards the adjoint state  $\Theta$ . We apply integration by parts, once with respect to time (using the Reynolds transport theorem A) and twice with respect to space (using Green's formula)

$$\begin{aligned}
\mathcal{L}_T \delta T &= - \int_{\Omega} \beta_1 (T^f - T^d) \delta T^f \\
&\quad - \int_{\Omega_1^f} \delta T^f \Theta_1^f + \int_{\Omega_1^0} \delta T^0 \Theta_1^0 \\
&\quad - \int_0^{t_f} \int_{\Omega_1} \frac{\partial \Theta_1}{\partial t} \delta T + \int_0^{t_f} \int_{\partial \Omega_1} \delta T \Theta_1 v_{S1} \\
&\quad + \int_0^{t_f} \int_{\partial \Omega_1} \frac{\partial \delta T}{\partial n} \Theta_1 - \int_0^{t_f} \int_{\Omega_1} \nabla \delta T \nabla \Theta_1 \\
&\quad \quad - \int_{\Omega_2^f} \delta T^f \Theta_2^f + \int_{\Omega_2^0} \delta T^0 \Theta_2^0 \\
&\quad - \int_0^{t_f} \int_{\Omega_2} \frac{\partial \Theta_2}{\partial t} \delta T + \int_0^{t_f} \int_{\partial \Omega_2} \delta T \Theta_2 v_{S2} \\
&\quad + \int_0^{t_f} \int_{\partial \Omega_2} \frac{\partial \delta T}{\partial n} \Theta_2 - \int_0^{t_f} \int_{\Omega_2} \nabla \delta T \nabla \Theta_2 \\
&\quad \quad - \int_0^{t_f} \int_{\partial \Omega} \delta T \Theta_D - \int_0^{t_f} \int_{\Gamma} \delta T \Theta_I \\
&\quad \quad \quad - \int_0^{t_f} \int_{\Gamma} \psi [\nabla \delta T_i]_2^1 \cdot \nabla \phi \\
&\iff \mathcal{L}_T \delta T = - \int_{\Omega} \beta_1 (T^f - T^d) \delta T^f \\
&\quad - \int_{\Omega_1^f} \delta T^f \Theta_1^f + \int_{\Omega_1^0} \delta T^0 \Theta_1^0 \\
&\quad - \int_0^{t_f} \int_{\Omega_1} \frac{\partial \Theta_1}{\partial t} \delta T + \int_0^{t_f} \int_{\partial \Omega_1} \delta T \Theta_1 v_{S1} \\
&\quad + \int_0^{t_f} \int_{\partial \Omega_1} \frac{\partial \delta T}{\partial n} \Theta_1 - \int_0^{t_f} \int_{\partial \Omega_1} \frac{\partial \Theta_1}{\partial n} \delta T + \int_0^{t_f} \int_{\Omega_1} \delta T \nabla^2 \Theta_1 \\
&\quad \quad - \int_{\Omega_2^f} \delta T^f \Theta_2^f + \int_{\Omega_2^0} \delta T^0 \Theta_2^0 \\
&\quad - \int_0^{t_f} \int_{\Omega_2} \frac{\partial \Theta_2}{\partial t} \delta T + \int_0^{t_f} \int_{\partial \Omega_2} \delta T \Theta_2 v_{S2} \\
&\quad + \int_0^{t_f} \int_{\partial \Omega_2} \frac{\partial \delta T}{\partial n} \Theta_2 - \int_0^{t_f} \int_{\partial \Omega_2} \frac{\partial \Theta_2}{\partial n} \delta T + \int_0^{t_f} \int_{\Omega_2} \delta T \nabla^2 \Theta_2 \\
&\quad \quad - \int_0^{t_f} \int_{\partial \Omega} \delta T \Theta_D - \int_0^{t_f} \int_{\Gamma} \delta T \Theta_I \\
&\quad \quad \quad - \int_0^{t_f} \int_{\Gamma} \psi [\nabla \delta T_i]_2^1 \cdot \nabla \phi,
\end{aligned}$$

with  $v_{S1}$  and  $v_{S2}$  the velocities of the control volumes  $\Omega_1$  and  $\Omega_2$  respectively with  $v_{S1} = \vec{v}_S \cdot n$  and  $v_{S2} = -\vec{v}_S \cdot n$ . These two terms are only non-zero on  $\Gamma$ . Sorting the terms by their domain of integration and setting  $\delta T^0 = 0$ , we obtain

$$\begin{aligned}
\mathcal{L}_T \delta T &= \int_0^{t_f} \int_{\Omega_1} \left( \frac{\partial \Theta_1}{\partial t} + \nabla^2 \Theta_1 \right) \delta T \\
&\quad + \int_0^{t_f} \int_{\Omega_2} \left( \frac{\partial \Theta_2}{\partial t} + \nabla^2 \Theta_2 \right) \delta T \\
&\quad - \int_{\Omega_1} \left( \Theta_1^f - \beta_1 (T^f - T^d) \right) \delta T^f \\
&\quad - \int_{\Omega_2} \left( \Theta_2^f - \beta_1 (T^f - T^d) \right) \delta T^f \\
&\quad + \int_0^{t_f} \int_{\partial \Omega} \left( -\frac{\partial \Theta}{\partial n} \delta T + \frac{\partial \delta T}{\partial n} (\Theta - \Theta_D) \right) \\
&\quad + \int_0^{t_f} \int_{\Gamma} \left( \Theta_1 \vec{v}_S \cdot n - \Theta_2 \vec{v}_S \cdot n - \frac{\partial \Theta_1}{\partial n} + \frac{\partial \Theta_2}{\partial n} - \Theta_I \right) \delta T \\
&\quad + \int_0^{t_f} \int_{\Gamma} (\Theta - \psi |\nabla \phi|) [\nabla \delta T]_2^1 \cdot n \quad \forall \delta T = 0,
\end{aligned}$$

where the second to last term, corresponds to the normal jump in adjoint temperature, which gives  $\Theta_I = -[\nabla \Theta]_2^1 \cdot n$  because  $\Theta_1 = \Theta_2$  on  $\Gamma$ . By altering the directions of variations and eliminating certain terms, we obtain the adjoint temperature problem

$$\left\{ \begin{array}{ll}
-\frac{\partial \Theta_1}{\partial t} = \nabla^2 \Theta_1 & \text{in } \Omega_1(t) \\
-\frac{\partial \Theta_2}{\partial t} = \nabla^2 \Theta_2 & \text{in } \Omega_2(t) \\
\Theta(x, t_f) = \beta_1 (T^f - T^d) & \text{in } \Omega \\
\frac{\partial \Theta(x, t)}{\partial n} = 0 & \text{on } \partial \Omega \\
\Theta(x, t) = \psi |\nabla \phi| & \text{on } \Gamma(t) \\
\Theta_I = -[\nabla \Theta]_2^1 \cdot n & \text{on } \Gamma(t)
\end{array} \right. \quad (\text{AT})$$

The first two equations of the adjoint temperature problem (AT) are the heat equations in reverse time. The third equation is the initial condition for the adjoint temperature field, that depends on the desired temperature distribution  $T^d$ . The fourth equation is a homogeneous Neumann boundary condition for the adjoint field mapped from equations FP.d. The second to last equation is the Dirichlet bound-

any condition at the interface that now depends on the value of the adjoint level set function  $\psi$ . We can already note that the adjoint level set function will not behave as a signed distance function but as an auxiliary variable that acts on the temperature field through the Dirichlet boundary condition at the interface  $\Gamma$ . Finally, the last equation states that the multiplier  $\Theta_I$ , defined on  $\Gamma$ , must be equal to the jump in normal gradient of  $\Theta$ .

### 2.2.3 Adjoint level set

The geometric non-linearity induced by  $\phi$  on the domains of integration requires a careful treatment of each term as described in [Bernauer & Herzog 2011]. We denote  $D[H(\phi); \delta\phi]$  the variation of  $H(\cdot)$  in the direction  $\delta\phi$ . The adjoint level set equations, derived by setting  $\mathcal{L}_\phi(\cdot) = 0$ , then reads

$$\begin{aligned} \mathcal{L}_\phi \delta\phi &= D[\mathcal{J}(T, \phi, w); \delta\phi] \\ &+ D\left[\int_0^{t_f} \int_\Omega \left(\frac{\partial T}{\partial t} - \nabla^2 T\right) \Theta; \delta\phi\right] \\ &- D\left[\int_0^{t_f} \int_{\partial\Omega} \left(\frac{\partial T}{\partial n} - u\right) \Theta_D; \delta\phi\right] \\ &- D\left[\int_0^{t_f} \int_\Gamma (T - T_M - \varepsilon_V v_S - \varepsilon_\kappa \kappa) \Theta_I; \delta\phi\right] \\ &- D\left[\int_0^{t_f} \int_\Gamma \left(\frac{\partial \phi}{\partial t} + [\nabla T_i]_2^1 \cdot \nabla \phi\right) \psi; \delta\phi\right]. \end{aligned}$$

We divide the contributions of the adjoint level set term-by-term. The contribution from the cost functional, removing the terms that do not depend on  $\phi$ , simplifies to

$$D[\mathcal{J}(T, \phi, w); \delta\phi] = D\left[\frac{\beta_2}{2} \int_{\Gamma^f} |\phi^f|^2; \delta\phi\right] + D\left[\frac{\beta_3}{2} \int_{\Gamma^f} 1; \delta\phi\right].$$

By using the theorem on the derivative of a boundary integral (Theorem B), we obtain

$$\begin{aligned} D[\mathcal{J}(T, \phi, w); \delta\phi] &= -\frac{\beta_2}{2} \int_{\Gamma^f} \frac{\delta\phi^f}{|\nabla\phi|} \left(\frac{\partial|\phi^f|^2}{\partial n} + \kappa|\phi^f|^2\right) - \frac{\beta_3}{2} \int_{\Gamma^f} \frac{\delta\phi^f}{|\nabla\phi|} \kappa \\ &= -\frac{\beta_2}{2} \int_{\Gamma^f} \frac{\delta\phi^f}{|\nabla\phi|} \left(\frac{\partial|\phi^f|^2}{\partial n} + \kappa\left(|\phi_{t_f}^f|^2 + \frac{\beta_3}{\beta_2}\right)\right). \end{aligned}$$

By specifying that  $\delta\phi = 0$  on  $\delta\Omega$ , the second and third terms are then equal to zero. For the fourth term, we need to assume that the adjoint level set  $\phi$  and the multiplier  $\Theta_I$  are defined on all of  $\Omega$ . Under such assumptions, we can apply the same boundary integral theorem (Theorem B)

$$-D\left[\int_0^{t_f} \int_\Gamma (T - T_M - \varepsilon_V v_S - \varepsilon_\kappa \kappa) \Theta_I; \delta\phi\right]$$

$$= - \int_0^{t_f} \int_{\Gamma} \frac{\delta\phi}{|\nabla\phi|} \left( \frac{\partial(T - T_M - \varepsilon_V v_S - \varepsilon_{\kappa}\kappa)}{\partial n} + \kappa(T - T_M - \varepsilon_V v_S - \varepsilon_{\kappa}\kappa) \right) \Theta_I.$$

The value of the temperature on  $\Gamma$  is always equal to  $T - T_M - \varepsilon_V v_S - \varepsilon_{\kappa}\kappa$ . Moreover, using the fact that  $T_M$  is constant along the interface, the above equation simplifies to

$$\begin{aligned} -D \left[ \int_0^{t_f} \int_{\Gamma} (T - T_M - \varepsilon_V v_S - \varepsilon_{\kappa}\kappa) \Theta_I; \delta\phi \right] &= - \int_0^{t_f} \int_{\Gamma} \frac{\delta\phi}{|\nabla\phi|} \left( \frac{\partial(T - \varepsilon_V v_S - \varepsilon_{\kappa}\kappa)}{\partial n} \right) \Theta_I \\ &= - \int_0^{t_f} \int_{\Gamma} \frac{\delta\phi}{|\nabla\phi|} \left( \frac{\partial T}{\partial n} - \varepsilon_V \frac{\partial v_S}{\partial n} - \varepsilon_{\kappa} \frac{\partial \kappa}{\partial n} \right) \Theta_I. \end{aligned}$$

Looking at the resulting terms in the integral, we have

- $\partial T / \partial n$  corresponding to the normal derivative of the temperature, which is well defined in all  $\Omega$ ,
- $\partial v_S / \partial n$  that is constant as the velocity of the interface will be extended numerically in the normal direction by a procedure defined in Chapter 4,
- $\partial \kappa / \partial n$ , the normal derivative of the curvature exists in the level set framework and can be computed in all of  $\Omega$  but has no true "physical" meaning hence we ignore this term when advecting the variables, leading to an incomplete adjoint derivation. Nevertheless, the optimization results obtained in Chapter 6, in cases where  $\varepsilon_{\kappa}$  is non-zero, confirm that this assumption is acceptable.

Finally, by applying the same boundary integral theorem together with the chain rule, the last term simplifies to

$$\begin{aligned} & -D \left[ \int_0^{t_f} \int_{\Gamma} \left( \frac{\partial\phi}{\partial t} + [\nabla T_i]_2^1 \cdot \nabla\phi \right) \psi; \delta\phi \right] \\ &= - \int_0^{t_f} \int_{\Gamma} \frac{\delta\phi}{|\nabla\phi|} \left( \frac{\partial}{\partial n} \left( \left( \frac{\partial\phi}{\partial t} + [\nabla T_i]_1^2 \cdot \nabla\phi \right) \psi \right) + \kappa \left( \frac{\partial\phi}{\partial t} + [\nabla T_i]_1^2 \cdot \nabla\phi \right) \psi \right) \\ & \quad + \int_0^{t_f} \int_{\Gamma} \left( \frac{\partial\delta\phi}{\partial t} + [\nabla T_i]_1^2 \cdot \nabla\delta\phi \right) \psi \\ &= - \int_0^{t_f} \int_{\Gamma} \frac{\delta\phi}{|\nabla\phi|} \frac{\partial}{\partial n} \left( \frac{\partial\phi}{\partial t} + [\nabla T_i]_1^2 \cdot \nabla\phi \right) \psi + \int_0^{t_f} \int_{\Gamma} \left( \frac{\partial\delta\phi}{\partial t} + [\nabla T_i]_1^2 \cdot \nabla\delta\phi \right) \psi. \end{aligned}$$

By using the assumption of constant velocity  $v_S$  extended in the normal direction on all of  $\Omega$ , we obtain that

$$-D \left[ \int_0^{t_f} \int_{\Gamma} \left( \frac{\partial\phi}{\partial t} + [\nabla T_i]_2^1 \cdot \nabla\phi \right) \psi; \delta\phi \right] = - \int_0^{t_f} \int_{\Gamma} \frac{\delta\phi}{|\nabla\phi|} \left( \frac{\partial\delta\phi}{\partial t} + [\nabla T_i]_1^2 \cdot \nabla\delta\phi \right) \psi.$$

In order to obtain the adjoint Stefan condition, we need to move the derivatives from  $\delta\phi$  to  $\psi$ . Assuming that  $\psi$  is defined on all of  $\Omega$ , we can use the corollary C

on integration by parts in time on a moving surface

$$\begin{aligned}
& - \int_0^{t_f} \int_{\Gamma} \frac{\delta\phi}{|\nabla\phi|} \left( \frac{\partial\phi}{\partial t} + [\nabla T_i]_1^2 \cdot \nabla\phi \right) \psi = - \int_{\Gamma^f} \delta\phi^f \psi^f + \int_{\Gamma^0} \delta\phi^0 \psi^0 \\
& - \int_0^{t_f} \int_{\Gamma} \left( \delta\phi \frac{\partial\psi}{\partial t} + \nabla(\delta\phi \psi) \cdot v_S + \delta\phi \psi \operatorname{div}_{\Gamma} v_S + \psi [\nabla T_i]_1^2 \cdot \nabla\delta\phi \right) \\
& \quad = - \int_{\Gamma^f} \delta\phi^f \psi^f + \int_{\Gamma^0} \delta\phi^0 \psi^0 - \int_0^{t_f} \int_{\Gamma} \delta\phi \frac{\partial\psi}{\partial t} \\
& - \int_0^{t_f} \int_{\Gamma} \delta\phi (\nabla(\psi) \cdot v_S + \psi \operatorname{div}_{\Gamma} v_S) + \int_0^{t_f} \int_{\Gamma} (\psi [\nabla T_i]_1^2 \cdot \nabla\delta\phi - \psi \nabla\delta\phi \cdot v_S).
\end{aligned}$$

The last term cancels due to the fact that  $[\nabla T_i]_1^2 = v_S$  on  $\Gamma$ . The resulting terms of the adjoint level set, using  $\delta\phi^0 = 0$ , give

$$\begin{aligned}
\mathcal{L}_{\phi} \delta\phi &= -\frac{\beta_2}{2} \int_{\Gamma^f} \frac{\delta\phi^f}{|\nabla\phi|} \left( \frac{\partial|\phi^f|^2}{\partial n} + \kappa \left( |\phi^f|^2 + \frac{\beta_3}{\beta_2} \right) \right) \\
& \quad - \int_0^{t_f} \int_{\Gamma} \frac{\delta\phi}{|\nabla\phi|} \left( \frac{\partial T}{\partial n} - \varepsilon_V \frac{\partial v_S}{\partial n} \right) \Theta_I \\
& \quad - \int_{\Gamma^f} \delta\phi^f \psi^f - \int_0^{t_f} \int_{\Gamma} \delta\phi \frac{\partial\psi}{\partial t} \\
& \quad - \int_0^{t_f} \int_{\Gamma} \delta\phi (\nabla(\psi) \cdot v_S + \psi \operatorname{div}_{\Gamma} v_S).
\end{aligned}$$

By regrouping the terms by their domains of integration and requiring  $\mathcal{L}_{\phi} \delta\phi = 0 \forall \delta\phi$ , we obtain that

$$\begin{aligned}
\psi^f &= -\frac{1}{|\nabla\phi|} \frac{\beta_2}{2} \left( \frac{\partial|\phi^f|^2}{\partial n} + \kappa \left( |\phi^f|^2 + \frac{\beta_3}{\beta_2} \right) \right) \\
-\frac{\partial\psi}{\partial t} &= \nabla\psi \cdot v_S + \psi \operatorname{div}_{\Gamma} v_S + \frac{1}{|\nabla\phi|} \left( \frac{\partial T}{\partial n} - \varepsilon_V \frac{\partial v_S}{\partial n} \right) \Theta_I.
\end{aligned}$$

Recalling that  $\Theta_I$  was defined in the adjoint temperature and that  $v_S$  is constant in the normal direction, we have

$$\begin{aligned}
\psi^f &= -\frac{1}{|\nabla\phi|} \frac{\beta_2}{2} \left( \frac{\partial|\phi^f|^2}{\partial n} + \kappa \left( |\phi^f|^2 + \frac{\beta_3}{\beta_2} \right) \right) \\
\frac{\partial\psi}{\partial t} + \operatorname{div}(\psi v_S) &= \frac{1}{|\nabla\phi|} \left( \frac{\partial T}{\partial n} - \varepsilon_V \frac{\partial v_S}{\partial n} \right) [\nabla\Theta]_2^1 \cdot n
\end{aligned}$$

This equation can be interpreted as a first-order conservation law on  $\Gamma$  and the source term on the right-hand-side can be extended in the neighborhood of  $\Gamma$ .

### 2.2.4 The gradient equation

The last term to complete the adjoint Stefan problem is the gradient equation. By setting  $\mathcal{L}_w \delta w = 0$  in Equation 2.1, we have

$$\begin{aligned} \mathcal{L}_w \delta w &= \left[ \left[ \frac{\beta_4}{2} \int_0^{t_f} \int_{\partial\Omega} |w|^2; \delta w \right] - \left[ \int_0^{t_f} \int_{\partial\Omega} \left( \frac{\partial T}{\partial n} - w \right) \Theta_D; \delta w \right] \right] \\ &= \int_0^{t_f} \int_{\partial\Omega} (\beta_4 |w| - \Theta) \delta w, \end{aligned}$$

thus obtaining at the gradient equation  $0 = \beta_4 w + \Theta$  on  $\partial\Omega$ . Here, we used the previously defined multiplier  $\Theta_D$  that is identically equal to  $\Theta$  on  $\partial\Omega$  (see adjoint temperature derivation in Section 2.2.2). We now gather the adjoint temperature and level set with the gradient equation to construct the adjoint Stefan problem (AP) below.

Find a function  $\Theta : \Omega \times [t_f, 0] \rightarrow \mathbb{R}$  and a function  $\psi : \Omega \times [t_f, 0] \rightarrow \mathbb{R}$  such that

$$\left\{ \begin{array}{ll} -\frac{\partial \Theta_1}{\partial t} = \nabla^2 \Theta_1 & \text{in } \Omega_1(t) \text{ (AP.a)} \\ -\frac{\partial \Theta_2}{\partial t} = \nabla^2 \Theta_2 & \text{in } \Omega_2(t) \text{ (AP.b)} \\ \Theta(x, t_f) = \beta_1 (T(t_f) - T_{t_f}) & \text{in } \Omega \text{ (AP.c)} \\ \frac{\partial \Theta(x, t)}{\partial n} = 0 & \text{on } \partial\Omega \text{ (AP.d)} \\ \Theta(x, t) = \psi |\nabla \phi| & \text{on } \Gamma(t) \text{ (AP.e)} \\ \frac{\partial \psi}{\partial t} + \text{div}(\psi \vec{v}_S) = \frac{1}{|\nabla \phi|} \left( \frac{\partial T}{\partial n} - \varepsilon_V \frac{\partial v_S}{\partial n} \right) [\nabla \Theta_i]_2^1 \cdot n & \text{on } \Gamma(t) \text{ (AP.f)} \\ \psi(x, t_f) = -\frac{\beta_2}{2} \left( \frac{\partial}{\partial n} |\phi_{t_f}|^2 + \kappa \left( |\phi_{t_f}|^2 + \frac{\beta_3}{\beta_2} \right) \right) & \text{in } \Omega \text{ (AP.g)} \\ 0 = \beta_4 w + \Theta & \text{on } \partial\Omega \text{ (AP.h)} \end{array} \right. \quad (\text{AP})$$

- Equations AP.a and AP.b correspond to the heat equations in reverse time.
- Equation AP.c is used to initialize the adjoint temperature field in both phases.
- In Equation AP.e, the in-homogeneous Neumann boundary condition imposed on  $\partial\Omega$  existing in the forward problem (FP) is mapped to an homogeneous alternative.
- The Dirichlet boundary condition at the interface is defined in Equation AP.f where the adjoint temperature is equal to the adjoint level set value  $\psi$  augmented by the Hamiltonian of the level set function  $|\phi|$ .

- In Equation AP.f, the adjoint level set advection equation is defined. The source term corresponds to the adjoint Stefan condition and takes into account the normal jump in gradient of adjoint temperature across the interface.
- The adjoint level set function is initialized using Equation AP.g.
- The last Equation (AP.h) represents the optimality condition, where the left-hand side represents the gradient used to update the control variable  $w$ .

### 2.3 Gradient-based optimization procedure

In optimization problems where the information on the gradient can be extracted through the adjoint, the convergence is strongly improved and this will be shown when comparing our adjoint-based procedure with a derivative-free one. In the family of the adjoint-based methods, the gradient-based ones are the methods of choice to prove that the optimal control approach is reasonable due to their straight forward implementation. Therefore, we chose to solve the minimization problem (MP) by using the limited memory BFGS (L-BFGS) method, a quasi-Newton method originally described in [Liu. & Nocedal 1989]. The main characteristic of this method is that it determines the descent direction by preconditioning the gradient with an approximation of the Hessian matrix. This information is obtained using past approximations (the number of approximations is determined by the memory length parameter which is set to  $m = 10$  in all our optimization test cases) as well as the gradient. As an initial guess for the initial Hessian, we use the scaled identity matrix as described in [Wright & Nocedal 2006]. Algorithm 2 summarizes the L-BFGS method used in our numerical examples. The algorithm is stopped at a given iteration  $n$  if one of the following criteria is fulfilled

- The relative difference in control variable  $\left| \frac{w^{n-1} - w^n}{w^n} \right| < 10^{-8}$ .
- The relative difference in cost functional  $\left| \frac{\mathcal{J}^{n-1} - \mathcal{J}^n}{\mathcal{J}^n} \right| < 10^{-8}$ . This criterion can be relaxed to allow temporary increase of the cost functional, for example to ‘escape’ a local minimum.
- The norm of the gradient  $|\nabla \mathcal{J}^n| < 10^{-6}$ .

In the next Chapter, we will present a novel Cut Cell method to solve diffusive transport. The heat equations in both the forward and adjoint problem will be solved by this method that modifies the standard finite difference formula to take into account the boundary conditions of a given interface in the underlying Cartesian grid.

---

**Algorithm 2:** Optimization procedure using the L-BFGS method

---

**input :**  $w^0, m = 10$

**output:**  $w, T, \phi, \Theta, \psi$

$k \leftarrow 0, l \leftarrow 0$

**while** *not converged* **do**

    Solve the forward Stefan problem (FP) for  $T^k$  and  $\phi^k$

    Solve the adjoint Stefan problem (AP) for  $\Theta^k$  and  $\psi^k$

    Compute the gradient:

$$\nabla \mathcal{J}^k = \beta_4 w^k + \Theta^k$$

**if**  $k \geq 1$  **then**

$$s^{k-1} = \sigma^{k-1} d^{k-1} \quad g^{k-1} = \nabla \mathcal{J}^k - \nabla \mathcal{J}^{k-1}$$

**if**  $(s^{k-1})^T g^{k-1} \leq 0$  **then**

$l \leftarrow 0$

**else if**  $(s^{k-1})^T g^{k-1} > 0$  **then**

$l \rightarrow l + 1$

**if**  $l > m$  **then**

                Remove  $\{s^{l-m}, g^{l-m}\}$

**end**

            Add  $\{s^{l-m}, s^{l-m}\}$

**end**

**end**

    Choose an initial approximation to the inverse of the Hessian  $H_0^k$

    Construct the direction  $d^k = -H^k \nabla \mathcal{J}^k$

    Determine  $\sigma^k$  using a Line Search algorithm with backtracking where

$$\sigma^k = \operatorname{argmin} \mathcal{J}(w^k + \sigma^k d^k)$$

    Update  $w^{k+1} = w^k + \sigma^k d^k$

$k \rightarrow k + 1$

**end**

---



# A novel Cut Cell method for diffusive transport

---

## Contents

---

<b>3.1</b>	<b>Discrete Cut Cell operators</b>	<b>29</b>
3.1.1	Motivation	29
3.1.2	Governing principles	31
3.1.3	Loss of accuracy with interpolation	33
3.1.4	Additional geometric information to restore accuracy	34
<b>3.2</b>	<b>Volume and surface capacities</b>	<b>37</b>
3.2.1	Mesh and geometry input	37
3.2.2	Capacities of the first kind	37
3.2.3	Capacities of the second kind	39
<b>3.3</b>	<b>Marching squares algorithm</b>	<b>40</b>
<b>3.4</b>	<b>Validation on stationary geometries</b>	<b>43</b>

---

## 3.1 Discrete Cut Cell operators

### 3.1.1 Motivation

This section motivates the choices underlying the design of the proposed Cut Cell operators. The focus is set on the numerical solution of the Poisson problem

$$\nabla^2 T = \sigma, \quad (3.1)$$

where  $\sigma$  is a specified source term and  $T$  is also subject to a Dirichlet boundary condition  $D$ . Simply put, the question addressed here is: what is the minimal amount of geometric information required to discretize the Poisson equation on a Cartesian grid but on an arbitrary domain, while guaranteeing that the discrete Laplacian operator

1. Preserves a classical three-point star-shaped stencil, and
2. Guarantees first order accuracy in the mesh-aligned cases.

Many numerical methods have been developed to handle complex geometries. Unstructured meshes are one common way to work around simulations with arbitrary complex geometries. Such mesh elements are usually triangular or quadrilateral and naturally allow for local adaptivity, but generating high-quality unstructured meshes continues to be a challenge. The efficiency and robustness of such algorithms remain the key issues in this area of research [Mavriplis 1995]. Other widely used methods include so-called overset (or chimera) grids. However, conservation can generally not be achieved for these methods due to the extensive use of interpolation between the various grid levels. Similarly, provable stability properties are difficult to obtain [Sherer & Scott 2005].

Another class of approaches to complex geometries include immersed boundary methods. Immersed boundary methods can be separated into two large classes with diffuse or sharp interface representations. Historically, diffuse interfaces have been used in IB methods, starting with the work of Peskin [Peskin 1972]. Early adaptations of the IB method required arbitrary tuning parameters either for describing the forcing effect of the immersed boundary [Peskin 1972, Peskin 1977, Goldstein *et al.* 1993, Lai & Peskin 2000, Beyer & Leveque 2006] or characterizing the boundary velocity [Peskin 2003]. The second branch of IB methods require the strong imposition of boundary condition at a sharp interface. These are also known as Cut Cell methods. Calhoun and Leveque [Calhoun & Leveque 2000] developed a widely used finite volume algorithm for complex geometries for solving scalar equations in a collocated grid. More recently, Cheny and Botella [Cheny & Botella 2010] developed a fully conservative cut-cell method on staggered grids. The common trade-off in cut-cell methods is between explicit handling of many special cases in which a computational cell can be intersected by object boundaries and making use of black-box least-squares methods to construct derivatives. The second method can be shown to lack the desired conservation properties.

The level set function, defined in Chapter 1, is used there to compute geometric moments, such as the centroid coordinates of wet volumes or the area of wet faces. One advantage of the proposed method is that these geometric fields are the only information required to modify classical finite differences formulas in the vicinity of boundaries, while recovering well-known second-order formulas away from them and accommodating any stretching. The definition of these geometric fields and their number is determined from accuracy considerations. It will be shown, in particular, that the proposed operators degenerate to classical formulas for the mesh-aligned boundaries. The discrete calculus of Morinishi [Morinishi *et al.* 1998, Morinishi 2010] is leveraged to provide concise expressions for the discrete operators, whether for Dirichlet or Neumann boundary conditions. In this work, we will only focus on the Dirichlet case. Validation are provided that assess the scheme's second order accuracy, stability and ability to accurately represent transfers.

### 3.1.2 Governing principles

A subset of the operators defined by [Morinishi *et al.* 1998] is reemployed, namely the second order differentiation, interpolation, and permanent product operators on unitary grids defined respectively as

$$\left. \frac{\delta\phi}{\delta\xi_x} \right|_{i+1/2,j} = \phi_{i+1,j} - \phi_{ij}, \quad (3.2)$$

$$\left. \overline{\phi}^x \right|_{i+1/2,j} = \frac{\phi_{i+1,j} + \phi_{ij}}{2}, \quad (3.3)$$

and

$$\left. \widetilde{\phi\psi}^x \right|_{i+1/2,j} = \frac{\phi_{i+1,j}\psi_{ij} + \psi_{i+1,j}\phi_{ij}}{2}. \quad (3.4)$$

These definitions are straightforward to extend to either staggered ( $\phi_{i+1/2,j}$ ) and ( $\phi_{i,j+1/2}$ ) or nodal ( $\Phi_{i+1/2,j+1/2}$ ) fields. Likewise, operators acting upon the second direction,  $\delta(\cdot)/\delta\xi_y$ ,  $(\cdot)^y$  and  $(\widetilde{\cdot,\cdot})^y$ , are defined in the same manner. Finally, the restrictions to the one-dimensional case is obtained by dropping the last index. These operators will be used in the subsequent formula when deriving the discrete operators.

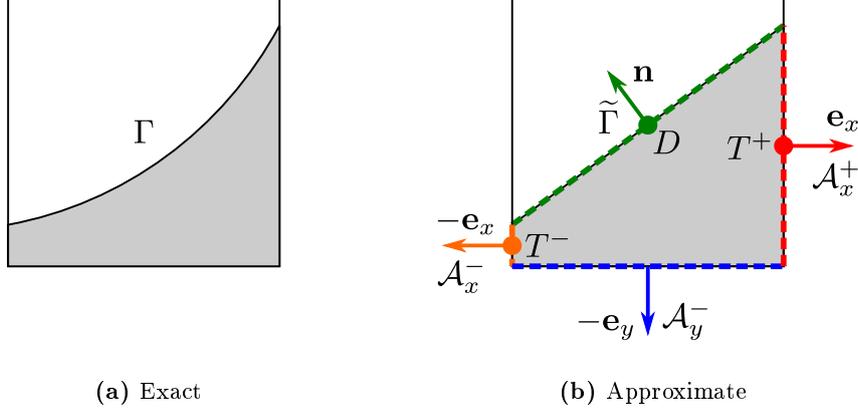
Cut Cell Methods are firmly grounded in the Finite Volume Method, which defines the primary discrete variables as cell-wise averages over mesh elements (as opposed to point-wise values in the Finite Difference Method, for example). The design of the Finite Volume operators is then based on the application of Stokes' theorem. For example, given a scalar field  $T$ , this theorem states that in a Cartesian coordinate system, the  $x$  component of the gradient  $\mathbf{q} \equiv \nabla T$  averaged over a cell  $\Omega$  may be computed as

$$|\Omega| q_x = \int_{\Omega} \frac{\partial T}{\partial x} dV = \oint_{\partial\Omega} T \mathbf{e}_x \cdot d\mathbf{S} \quad (3.5)$$

where  $|\cdot|$  denotes the measure operator,  $d\mathbf{S}$  the outward-pointing surface element,  $\mathbf{e}_x$  the unit vector along the  $x$  direction and  $(\partial\cdot)$  the contour operator.

For the sake of presentation, the case displayed in Figure 3.1a is considered, where  $\Omega$  consists of the intersection of a phase domain and a computational cell (a right hexahedron). The contour  $\partial\Omega$  then consists of the union of the three planar faces  $\mathcal{A}_x^-$ ,  $\mathcal{A}_x^+$  and  $\mathcal{A}_y^-$  as well as the boundary surface  $\Gamma$ . A piece-wise linear approximation of  $\Gamma$ , denoted  $\widetilde{\Gamma}$ , of length  $|\widetilde{\Gamma}|$  and unit normal  $(n_x, n_y)$ , can be defined as done in Figure 3.1b. Applying Equation 3.5 to  $\widetilde{\Omega}$  with  $T = 1$  then yields

$$\int_{\widetilde{\Omega}} \frac{\partial 1}{\partial x} dV = |\mathcal{A}_x^+| - |\mathcal{A}_x^-| + n_x |\widetilde{\Gamma}| = 0,$$



**Figure 3.1:** Example of an exact and approximate representation of a Cartesian cell cut by the interface.

which highlights the existence of a fundamental relation, sometimes referred to as a Surface Conservation Law (SCL)

$$|\mathcal{A}_x^+| - |\mathcal{A}_x^-| = -n_x |\tilde{\Gamma}|. \quad (3.6)$$

In other words, the knowledge of  $(|\mathcal{A}_\alpha|)_{\alpha \in \{x,y,z\}}$  implicitly defines a piece-wise linear approximation to the boundary. As a consequence, this surface information, henceforth referred to as the surface capacities, may serve to approximate the right-hand side of Equation 3.5. If the unknowns  $(T_{x/y}^\pm)$  are defined as averages over the wet areas  $(|\mathcal{A}_{x/y}^\pm|)$ , the formula

$$\oint_{\partial\tilde{\Omega}} T \mathbf{e}_x \cdot d\mathbf{S} = |\mathcal{A}_x^+| T^+ - |\mathcal{A}_x^-| T^- - (|\mathcal{A}_x^+| - |\mathcal{A}_x^-|) D,$$

is exact provided  $D$  is the Dirichlet condition averaged over the approximate boundary  $\tilde{\Gamma}$ . To complete the definition of the averaged  $x$ -component of the gradient, the volume capacity  $\mathcal{V} \equiv |\Omega|$  is also required, which results in the following tentative gradient operator

$$q_x^{\mathcal{V}1} \simeq (|\mathcal{A}_x^+| T^+ - |\mathcal{A}_x^-| T^- - (|\mathcal{A}_x^+| - |\mathcal{A}_x^-|) D) / \mathcal{V}.$$

It is worth stressing that the use of the SCL (Equation 3.6) in  $q_x^{\mathcal{V}1}$  guarantees that the discrete gradient vanishes when the solution and boundary values are matching constants ( $T^+ = T^- = D$ ). This notation can be generalized to arbitrary dimensions for any boundary geometry using the differentiation operator  $\delta \cdot / \delta \xi_\alpha$ ,  $\alpha \in \{x, y\}$

(defined in Equation 3.2), as follows

$$\forall \alpha \in \{x, y\}, \quad \text{grad}_\alpha^{\text{v1}}(T_\alpha, D) = \frac{1}{V} \left( \frac{\delta A_\alpha T_\alpha}{\delta \xi_\alpha} - \frac{\delta A_\alpha}{\delta \xi_\alpha} D \right), \quad (3.7)$$

where all components of the discrete vector field  $\underline{Q} = (q_\alpha)$ ,  $\alpha \in \{x, y, z\}$ , are collocated with  $D$ . In this first version of the gradient operator, the primary unknowns  $T_\alpha$  are collocated with the surface areas  $A_\alpha$ , whereas the Dirichlet boundary condition  $D$  is staggered in-between.

This construction, referred to as Arakawa E-grid [Arakawa & Lamb 1977], relies on the definition of multiple temperature fields. Such a grid configuration is not the one adopted by the MAC approach [Harlow & Welch 1965], which favors the C-grid that defines a single temperature field collocated with the  $D$  field here. A C-grid however means that the temperature unknowns  $T$  and surface capacities  $A_\alpha$  are staggered, in which case the latter together with  $V$  should be interpolated as follows

$$\forall \alpha \in \{x, y\}, \quad \text{grad}_\alpha^{\text{v2}}(T, D) = \frac{1}{\bar{V}^\alpha} \left( \frac{\delta \bar{A}^\alpha T}{\delta \xi_\alpha} - \frac{\delta \bar{A}^\alpha}{\delta \xi_\alpha} D \right), \quad (3.8)$$

which uses the interpolation operator  $\bar{\cdot}^\alpha$ ,  $\alpha \in \{x, y, z\}$  (defined in Equation 3.3).

### 3.1.3 Loss of accuracy with interpolation

It should be pointed out that formulas other than Equation 3.8 can also be written without interpolation of the geometric capacities, for example by collocating all surface capacities ( $A_\alpha$ ) with the primary variable  $T$ . However in the context of a second order operator such as the scalar Laplacian (Equation 3.1), the need for interpolation will resurface in the approximation of the divergence operator.

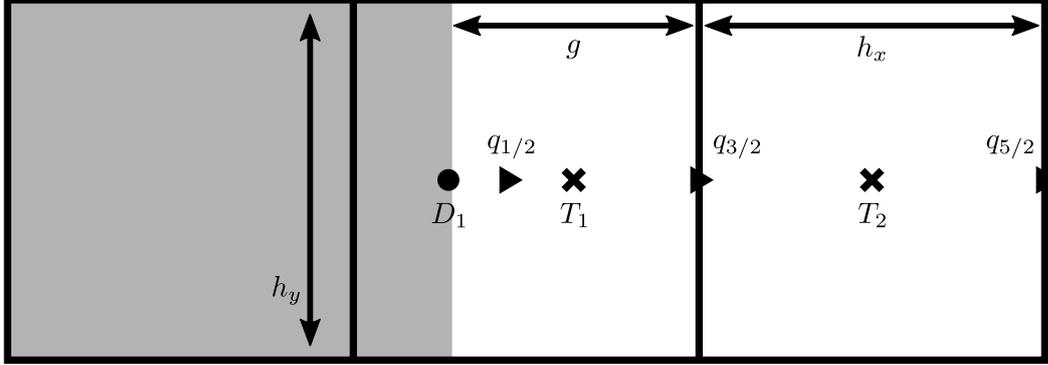
This section therefore focuses on the limitations of the second tentative formula (Equation 3.8), more specifically its failure to revert to a classical first order approximation of the second order derivative in the limit where the boundary is orthogonal to the direction of interest. This is the central point of the proposed cut-cell method, namely the enhancement of the geometric description of the boundary by means of additional volume and surface capacities to revise the gradient and divergence operators so as to achieve first-order accuracy in the vicinity of mesh-aligned boundaries.

To illustrate the limitation of the tentative gradient formula (Equation 3.8), the discretization of the second-order derivative along  $x$  in the mesh-aligned two-dimensional configuration displayed in Figure 3.2 is considered, where the fluid occupies the rightmost cells. This configuration is characterized by  $V_0 = 0$ ,  $V_1 = (h_x - g)h_y$ ,  $V_2 = h_x h_y$ ,  $A_{1/2} = 0$  and  $A_{3/2} = A_{5/2} = h_y$  (here,  $A$  stands for  $A_x$  since only the  $x$  contribution is considered). Using these expressions, Equation 3.8 simplifies to  $q_{-1/2} = 0$ ,

$$q_{1/2} = \frac{T_1 - D_1}{g}, \quad (3.9)$$

and

$$q_{3/2} = \frac{T_2 - (T_1 + D_1) / 2}{(g + h_x) / 2}. \quad (3.10)$$



**Figure 3.2:** Insufficient geometric information resulting in loss of accuracy in mesh-aligned geometries.

This approximation of the gradient is problematic for two reasons. At the boundary, the  $x$ -gradient value ( $q_{1/2}$ ) is under predicted by a factor of 2, since the denominator of the right-hand side of Equation 3.9 stands at  $g$  when it should match the distance between the points where  $D_1$  and  $T_1$  are defined,  $g/2$ . Away from the boundary, the  $x$ -gradient value ( $q_{3/2}$ ) depends on the boundary condition  $D_1$ , when one would simply expect the difference ( $T_2 - T_1$ ) to appear in the numerator of the right-hand side of Equation 3.10.

This simple exercise highlights the loss of accuracy associated with the interpolation of the geometric capacities. This can be associated with the fact that the capacities are defined as volume and surface integrals of the characteristic function of the fluid domain  $\Omega^f \subset \Omega$ , defined as

$$\forall \mathbf{x} \in \Omega, \quad H^f(\mathbf{x}) \equiv \int_{\mathbf{y} \in \Omega^f} \delta(\mathbf{x} - \mathbf{y}) dV, \quad (3.11)$$

where  $\Omega$  denotes the computational domain and  $\delta$  the multi-dimensional Dirac delta function.  $H^f$  is not differentiable in the classical sense, and one should tread carefully not to interpolate or differentiate its surface- or volume-averaged values.

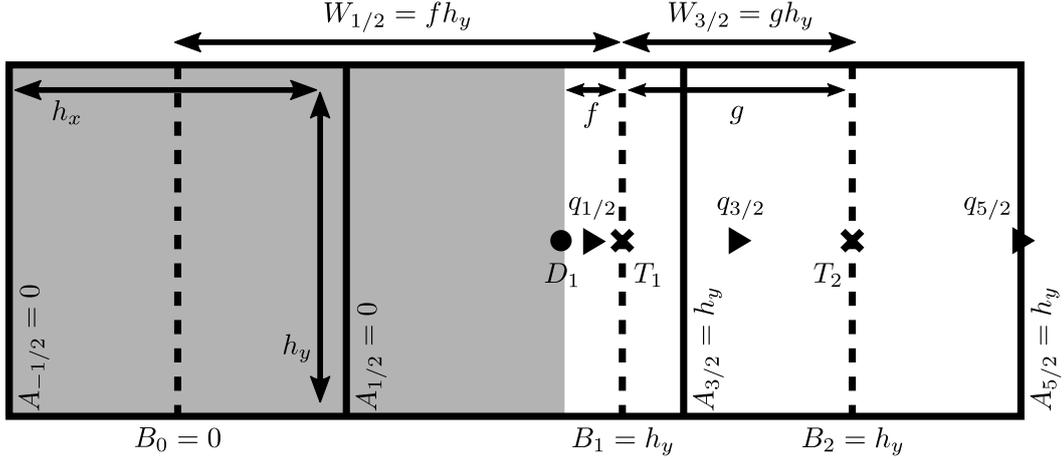
### 3.1.4 Additional geometric information to restore accuracy

An intuitive idea to alleviate the interpolations in Equation 3.8 is to add new information where the volume (cell-centered and denoted  $V$ ) and surface (face-centered and denoted  $(A_\alpha)$ ) capacities were previously interpolated. These new quantities, referred to as second-kind capacities, complement the already used first-kind capacities  $V$  and  $(A_\alpha)$ , also come in volume (face-centered and denoted  $(W_\alpha)$ ) and surface (cell-centered and denoted  $(B_\alpha)$ ) forms for arbitrary geometries. This additional

quantities yield the final gradient formula

$$\forall \alpha \in \{x, y\}, \quad \text{grad}_\alpha(T, D) = \frac{1}{W_\alpha} \left[ \frac{\delta B_\alpha T}{\delta \xi_\alpha} + \frac{\delta(\overline{A_\alpha}^\alpha - B_\alpha)D}{\delta \xi_\alpha} - \frac{\delta \overline{A_\alpha}^\alpha}{\delta \xi_\alpha} D \right], \quad (3.12)$$

that supersedes  $\text{grad}_\alpha^{(v1)}$  and  $\text{grad}_\alpha^{(v2)}$ .



**Figure 3.3:** Enhanced geometric information restoring accuracy in mesh-aligned geometries.

To show how the addition of the second-kind capacity restores first-order accuracy in the gradient computation, the configuration displayed in Figure 3.3 is considered. Since only  $x$  derivatives are considered,  $A$  again will stand for  $A_x$ , whereas  $W$  and  $B$  will respectively stand for  $W_x$  and  $B_x$ . Bearing this in mind, the configuration under study is characterized by  $V_0 = 0$ ,  $V_1 = 2fh_y$  and  $V_2 = h_x h_y$ ,  $A_{-1/2} = A_{1/2} = 0$ ,  $A_{3/2} = A_{5/2} = h_y$ ,  $B_0 = 0$ ,  $B_1 = B_2 = h_y$  and finally  $W_{-1/2} = 0$ ,  $W_{1/2} = fh_y$ ,  $W_{3/2} = gh_y$  and  $W_{5/2} = h_x h_y$ . Using these expressions, Equation 3.12 simplifies to  $q_{-1/2} = 0$  and

$$q_{1/2} = \frac{T_1 - D_1}{f},$$

$$q_{3/2} = \frac{T_2 - T_1}{g}$$

and

$$q_{5/2} = \frac{T_3 - T_2}{h_x}.$$

$T_0$  does not appear since it is outside of the fluid domain, and the boundary condition ( $D$ ) appears only in the faces adjacent to the boundary. The formulas obtained from Equation 3.12 are classical formulas since  $f$ ,  $g$  and  $h_x$  are the distances over which the differences  $T_1 - D_1$ ,  $T_2 - T_1$  and  $T_3 - T_2$  are defined. Finally, in the fluid domain and away from the boundaries, Equation 3.12 simply reverts to the classical gradient

formula

$$q_{x,i+1/2} = \frac{T_{i+1} - T_i}{x_{i+1} - x_i}.$$

In fact, the addition of the second-kind capacities is also sufficient to define the (cell-centered) volume-weighted divergence operator, which consists of the sum of the following contributions  $\forall \alpha \in \{x, y\}$  where  $N_\alpha$  denotes the boundary value of  $q_\alpha$ . If one sets the divergence to the product of the volume  $V$  with the local value of the source term  $\sigma$  as in the original Poisson problem (Equation 3.1), the configuration displayed in Figure 3.3 yields the trivial equation  $0 = 0$  in the first cell, and

$$h_y (q_{3/2} - N_{1/2}) = 2fh_y\sigma_1,$$

$$h_y (q_{5/2} - q_{3/2}) = h_x h_y \sigma_2,$$

in the others. Again classical formulas that degenerate to the classical formula

$$h_y (q_{i+1/2} - q_{i-1/2}) = h_y (x_{i+1/2} - x_{i-1/2}) \sigma_i,$$

in the fluid domain away from the boundary. Finally, the unknown  $\mathbf{N} = (N_\alpha)$  can be eliminated by substituting the gradient formula (Equation 3.12) in the divergence formula defined below

$$\operatorname{div}_\alpha (\underline{q}, \underline{N}) = \frac{\delta A_\alpha Q_\alpha}{\delta \xi_\alpha} + \frac{\delta(\bar{B}_\alpha - A_\alpha)N_\alpha}{\delta \xi_\alpha} - \frac{\overline{\delta B_\alpha}^\alpha}{\delta \xi} N_\alpha. \quad (3.13)$$

The boundary contribution (the last two terms in the right-hand side of Equation 3.13) are set to

$$\sum_\alpha \left[ \frac{\delta(\bar{B}_\alpha - A_\alpha)N_\alpha}{\delta \xi_\alpha} - \frac{\overline{\delta B_\alpha}^\alpha}{\delta \xi} N_\alpha \right] = \sum_\alpha \left[ \frac{\delta(\bar{B}_\alpha - A_\alpha)Q_\alpha}{\delta \xi_\alpha} - \frac{\overline{\delta B_\alpha}^\alpha}{\delta \xi} Q_\alpha \right],$$

which amount to identifying the heat flow through the boundary to the normal component of the temperature gradient. In the configuration displayed in Figure 3.3 this yields only one non-trivial equation,  $N_1 = q_1$ . Putting it all together, the proposed gradient and divergence operators, defined for arbitrary boundary geometries in Equations 3.12 and 3.13, discretize the Poisson problem (Equation 3.1) in the configuration displayed in Figure 3.3 as  $0 = 0$ ,

$$h_y \left( \frac{T_2 - T_1}{g} - \frac{T_1 - D}{f} \right) = 2fh_y\sigma_1,$$

and

$$h_y \left( \frac{T_3 - T_2}{h_x} - \frac{T_2 - T_1}{g} \right) = h_x h_y \sigma_2,$$

in the three cells displayed, while reverting to the classical formula

$$h_y \left( \frac{T_{i+1} - T_i}{x_{i+1} - x_i} - \frac{T_i - T_{i-1}}{x_i - x_{i-1}} \right) = h_x h_y \sigma_i,$$

in the fluid domain away from the boundary. As a consequence, formulas Equations 3.12 and 3.13 can be interpreted as generalizations of the classical second-order formulas to accommodate the presence of arbitrary boundaries while preserving first-order accuracy in the presence of mesh-aligned cases. The next section clarifies the notation employed thus far, in particular the definition of volume and surface capacities of the first and second kinds.

## 3.2 Volume and surface capacities

### 3.2.1 Mesh and geometry input

As far as the Cartesian mesh is concerned, a rectilinear mesh with  $n_x \times n_y$  cells is defined by specifying the following sets of user-defined abscissas

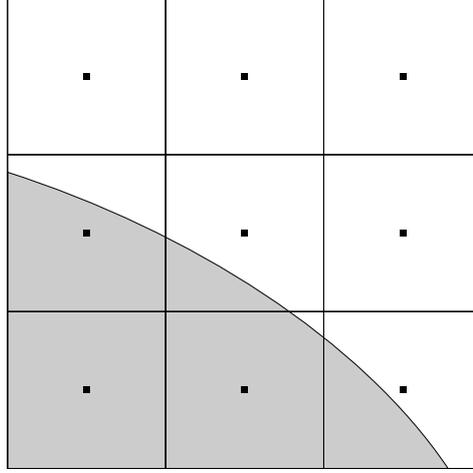
$$\begin{aligned} x_{1/2} &< x_{3/2} < \cdots < x_{n_x+1/2}, \\ y_{1/2} &< y_{3/2} < \cdots < y_{n_y+1/2}. \end{aligned}$$

Importantly, the mesh need not be uniform. Any given cell  $\Omega_{ij}$ , identified by a multi-index  $ij$ ,  $(i, j) \in \llbracket 1, n_x \rrbracket \times \llbracket 1, n_y \rrbracket$ , corresponds to the set of points  $(x, y)$  that simultaneously satisfy  $x_{i-1/2} < x < x_{i+1/2}$  and  $y_{j-1/2} < y < y_{j+1/2}$ . Regarding the boundary description, there exists a wide range of techniques to define a fluid domain, such as simplicial meshes or Constructive Solid Geometry (CSG) primitives and operations. Implicit representations by means of a void fraction or distance function (Level Set) are also commonly used. Regardless of the method employed, the assembly of the cut cell operators requires the computation of areas and volumes that correspond to the intersection of the fluid domain with Cartesian elements (faces or cells), as displayed in Figure 3.4. In this work, these computations are performed using a Marching Squares algorithm, that only requires the level set function values at the computational cells (detailed in the following section). Following the computation of the capacities, the geometry input is discarded.

### 3.2.2 Capacities of the first kind

Consider the Cartesian mesh displayed in Figure 3.4, partitioned into fluid ( $\Omega^f$ ) and solid ( $\Omega^s$ ) domains separated by a boundary ( $\Gamma$ ). In a finite volume setting, the primary variables  $\Phi_{ij}$  consist of averages of any given continuous field  $(x, y) \mapsto \phi(x, y)$  over the intersection of the fluid domain with any given hexahedral cell, defined as follows

$$V_{ij} \equiv \int_{\Omega_{ij}} \phi(\mathbf{x}) H^f(\mathbf{x}) d^2\mathbf{x},$$



**Figure 3.4:** Intersection of the fluid domain with Cartesian elements.

$$\phi_{ij} V_{ij} \equiv \int_{\Omega_{ij}} \phi(\mathbf{x}) H^f(\mathbf{x}) d^2\mathbf{x},$$

where  $H^f$  is the fluid characteristic function defined in Equation 3.11. The set  $V \equiv (V_{ij})$  is referred to as the volume capacities of the first kind.

When the field under consideration is linear, these averages coincide with the values at the fluid center of mass, displayed in Figure 3.5, defined for cells fully or partially occupied by the fluid (referred to as mixed cells). Although it does not appear explicitly in the Cut Cell operators, the coordinates of the fluid center of mass (displayed with crosses in Figure 3.5) are still required to define the second kind capacities, and are therefore temporarily stored. They are denoted as  $X$  and  $Y$  and defined for any cell  $\Omega_{ij}$  as

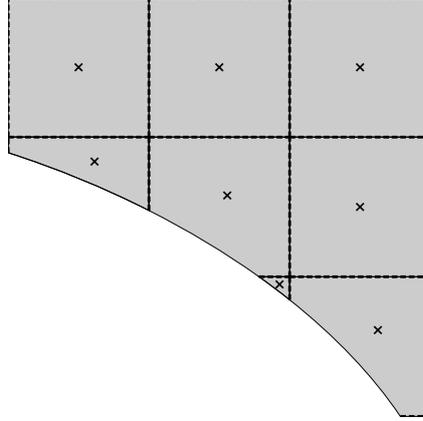
$$\begin{pmatrix} X_{ij} \\ Y_{ij} \end{pmatrix} V_{ij} \equiv \int_{\Omega_{ij}} \begin{pmatrix} x \\ y \end{pmatrix} H^f(\mathbf{x}) d^2\mathbf{x}.$$

The second step consists in computing the area of each of the faces wet by the fluid. Because the mesh is Cartesian, the faces adjacent to each cell are labelled based on the direction they are orthogonal to. These quantities, referred to as surface capacities, are staggered and denoted  $(A_\alpha)$  ( $\alpha \in \{x, y\}$ ). They are defined as

$$A_{i+1/2,j}^x \equiv \int_{y_{j-1/2}}^{y_{j+1/2}} H^f(x_{i+1/2}, y) dy,$$

and

$$A_{i,j+1/2}^y \equiv \int_{x_{i-1/2}}^{x_{i+1/2}} H^f(x, y_{j+1/2}) dx.$$



**Figure 3.5:** First kind capacities:  $V$  (filled areas),  $A_1$  (dashed vertical lines),  $A_2$  (dashed horizontal lines) and  $X$  and  $Y$  (crosses).

### 3.2.3 Capacities of the second kind

The coordinates of the fluid center of mass are used as follows. For each direction, the volume information is enriched by measuring how much fluid lies between each center of mass. This yields as many sets of staggered volumes, namely

$$W_{i+1/2,j}^x \equiv \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{X_{ij}}^{X_{i+1,j}} H^f(\mathbf{x}) d^2\mathbf{x},$$

and

$$W_{i,j+1/2}^y \equiv \int_{Y_{ij}}^{Y_{i,j+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} H^f(\mathbf{x}) d^2\mathbf{x},$$

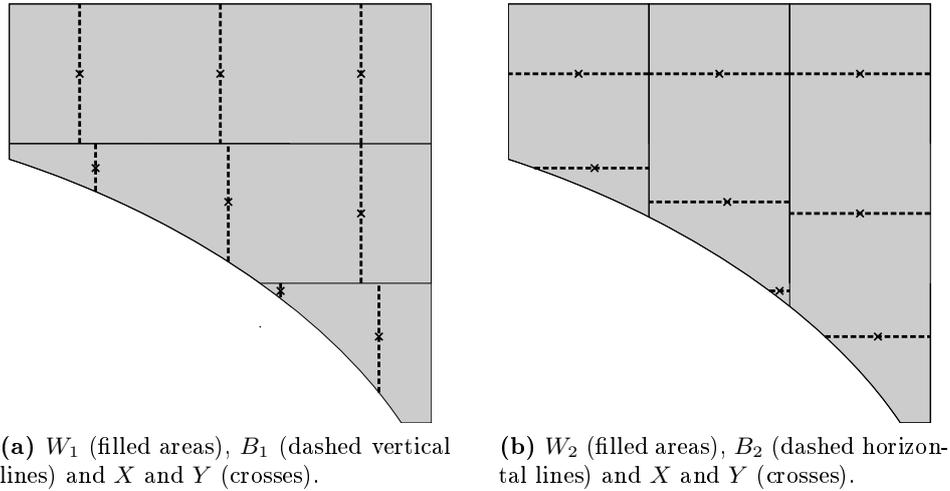
that will be denoted ( $W^\alpha$ ),  $\alpha \in \{x, y\}$ , and referred to as volume capacities of the second kind. The capacities  $W^x$  and  $W^y$  are represented as colored areas in the configuration displayed in Figure 3.6a and 3.6b, respectively. Likewise, the area wet by the fluid for the mesh-aligned faces that intercept the fluid center of mass will be required in each cell. This yields an additional set of cell-centered quantities,

$$B_{ij}^x = \int_{y_{j-1/2}}^{y_{j+1/2}} H^f(X_{ij}, y) dy,$$

and

$$B_{ij}^y = \int_{x_{i-1/2}}^{x_{i+1/2}} H^f(x, Y_{ij}) dx,$$

referred to as surface capacities of the second kind. The capacities  $B^x$  and  $B^y$  are represented as colored dashed lines in the configurations displayed in Figure 3.6a and 3.6b, respectively.



**Figure 3.6:** Schematic of the second kind capacities.

### 3.3 Marching squares algorithm

The computation of areas and volumes is carried out by using a marching squares algorithm using the values of the level set. This algorithm is used to determine one of the 16 different possible cases in two dimension (Figure 3.7), thus allowing the computation of the crossing points between the interface and the edges. As a preliminary step, we want to locate the mixed cells in our domain. We look at values of the level set function which are of different sign than the neighbors in each direction. With this condition, we ensure that we are ‘close’ to the interface. Note that this is not sufficient to determine if the cell is mixed or not as one could tag the cells adjacent to the mixed ones. Once this first sub-set of cells is known, we can apply the marching squares algorithm. The algorithm works as follows

1. The values of the corners of a cell, characterized by these 4 cardinal positions (south-west, south-east, north-east, north-west), are computed using a bi-quadratic interpolation.
2. Given the signs of the corners, the unique isovalue of the cell is computed by the formula

$$\text{iso} = \text{SW} + 2\text{SE} + 4\text{NE} + 8\text{NW}. \quad (3.14)$$

3. Depending on the isovalue (case), the intersection points are located re-using the bi-quadratic interpolation.

To determine the values of the corners, we perform a bi-quadratic interpolation on the  $3 \times 3$  stencil centered on the cell of interest. For this exercise, we assume a constant spacing of the Cartesian grid in all dimensions. We want to calculate the

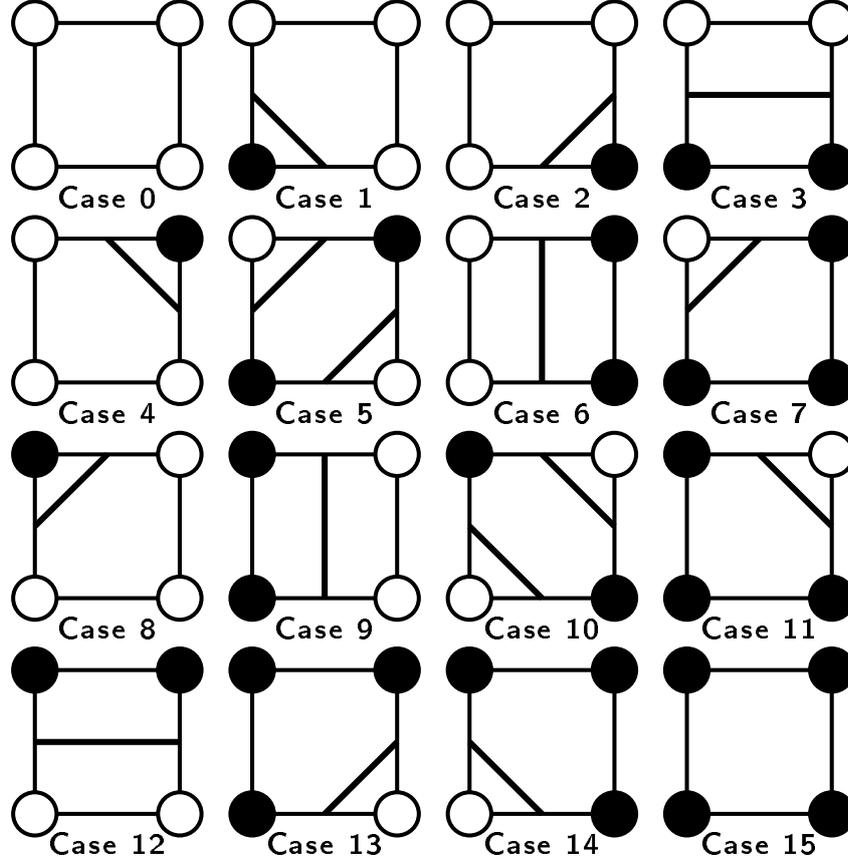


Figure 3.7: The 16 different cases of the marching squares algorithm.

interpolation matrix  $M$  given by

$$\begin{aligned}
 \phi(x, y) &= \sum_{i=0}^2 \sum_{j=0}^2 m_{ij} x^i y^j \\
 &= \begin{bmatrix} x^2 & x & 1 \end{bmatrix} \begin{bmatrix} m_{2,2} & m_{2,1} & m_{2,0} \\ m_{1,2} & m_{1,1} & m_{1,0} \\ m_{0,2} & m_{0,1} & m_{0,0} \end{bmatrix} \begin{bmatrix} y^2 \\ y \\ 1 \end{bmatrix} \\
 &= XMY^T,
 \end{aligned}$$

with  $\phi(x, y)$  the set of known data points of the level set function, yielding

$$\begin{aligned}
 \Phi &= GMG^T \\
 \begin{bmatrix} \phi_{-1,-1} & \phi_{-1,0} & \phi_{-1,1} \\ \phi_{0,-1} & \phi_{0,0} & \phi_{0,1} \\ \phi_{1,-1} & \phi_{1,0} & \phi_{1,1} \end{bmatrix}_{\Phi} &= \begin{bmatrix} (-1)^2 & -1 & 1 \\ 0^2 & 0 & 1 \\ 1^2 & 1 & 1 \end{bmatrix}_G \begin{bmatrix} m_{2,2} & m_{2,1} & m_{2,0} \\ m_{1,2} & m_{1,1} & m_{1,0} \\ m_{0,2} & m_{0,1} & m_{0,0} \end{bmatrix}_M \begin{bmatrix} (-1)^2 & 0 & (1)^2 \\ -1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{G^T}.
 \end{aligned}$$

The interpolation matrix  $M$  can now directly be computed by

$$\begin{aligned} M &= G^{-1}\Phi(G^T)^{-1} \\ &= G^{-1}\Phi(G^{-1})^T, \end{aligned}$$

with the matrix  $G^{-1}$  defined as

$$G^{-1} = \begin{bmatrix} 0.5 & -1 & 0.5 \\ -0.5 & 0 & 0.5 \\ 0 & 1 & 0 \end{bmatrix}.$$

Once the interpolation matrix is determined, we have the interpolated values of the level set function at the corner nodes  $\phi_{i\pm 1/2, j\pm 1/2}$  (ie. at the points  $(x = \pm 0.5, y = \pm 0.5)$ ). The isovalue corresponding to the cell is then determined by the relation described in Equation 3.14. Note that cases 5 and 10 do not exist, as we assume that the interface is crossing the underlying Cartesian grid only once per cell. The same interpolation matrix is then used to find the location of the intersection point by finding the zeros of the bi-quadratic interpolation along the edges of interest. With the crossing points, we can determine the centroid of the wetted area by using the following formula. Given a polygon defined by  $n$  vertices  $(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})$ , the point  $C = (C_x, C_y)$  is the centroid where

$$\begin{aligned} C_x &= \frac{1}{6\mathcal{A}} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i), \\ C_y &= \frac{1}{6\mathcal{A}} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i), \end{aligned}$$

and  $\mathcal{A}$  is the signed area defined by the shoelace formula

$$\mathcal{A} = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i).$$

The location of the centroid of the cell will not only be used in the computation of the second kind capacities but also in the initialization of the temperature field in the fresh cells when the interface uncovers new cells as the front moves (see Chapter 4). Moreover, we define the interface centroid as the mid point of the segment crossing the cell which will be used in the computation of the Stefan condition. Algorithm 3 summarizes the steps presented in this section.

The purpose of this algorithm is two-folded : (i) it constructs the discrete Laplacian operators using the faces and volume capacities, and (ii) it locates the interface in our Cartesian grid as well as the cell and interface centroids; informations which will be used in most of the numerical steps of solving the Stefan problem.

**Algorithm 3:** Cut Cell method pseudo-code

---

```

input :  $\phi, N$  (size of the domain)
output:  $L$  (Laplacian operator)
for  $(i, j) \in (2 : N - 1, 2 : N - 1)$  do
  if  $\phi^{i,j} \phi^{i\pm 1, j\pm 1} < 0$  then
    Construct the  $3 \times 3$  stencil  $\Phi$ 
    Compute the interpolation matrix  $M = G^{-1} \Phi (G^{-1})^T$ 
    Determine the sign of the vertices located at
       $(-0.5, -0.5)_{\text{SW}}, (0.5, -0.5)_{\text{SE}}, (0.5, 0.5)_{\text{NE}}, (-0.5, 0.5)_{\text{NW}}$ 
    Compute the isovalue  $\text{iso} = \text{SW} + 2 \text{SE} + 4 \text{NE} + 8 \text{NW}$ 
    if  $\text{iso} \neq 0$  and  $\text{iso} \neq 15$  then
      Store the indices  $(i, j)$  of the mixed cell
      Locate the intersection points depending on the case
      Compute and store the cell centroid on both sides of the interface
      Compute and store the interface centroid
      Compute the faces and volume capacities of first kind  $A_1, A_2, V$ 
        and second kind  $B_1, B_2, W_1, W_2$ 
    end
  end
end
Construct the Laplacian operator  $L$ 

```

---

### 3.4 Validation on stationary geometries

With the discrete Laplacian operators (one for each phase) now constructed, we can solve the heat equations on both domains

$$\frac{\partial T}{\partial t} = LT, \quad (3.15)$$

on both domains. Here,  $L$  is the discrete linear operator. We couple the Cut Cell space discretization with a Crank-Nicolson time discretization, where  $\tau$  is the time step,  $\Delta$  the uniform grid spacing and  $n$  the current iteration, resulting in the following discrete system

$$\frac{T^n - T^{n-1}}{\tau} = \frac{1}{2} \left[ \frac{1}{\Delta^2} LT^n + \frac{1}{\Delta^2} LT^{n-1} \right], \quad (3.16)$$

which requires the solution of a linear system forming a pentadiagonal matrix. We validate the method in different stationary setups. A convergence study is carried out for these cases, where the reference solution is taken as the simulation with the highest number of points per dimension. We look at the  $L - 2$  norm of the error in mixed cells, full cells (cells that are not mixed) and in all the cells. In each case, the initial temperature field is set to zero and we impose a Dirichlet boundary condition at the interface. The ratio  $\tau/\Delta^2 = 0.5$  is kept constant as we increase the number

of points. We consider the following test cases:

1. A solid circle of radius  $R = 0.85$ , initialized in a  $2 \times 2$  domain. The level set function is defined as

$$\phi(x, y) = \sqrt{x^2 + y^2} - R.$$

The Dirichlet  $T_D = 1$  boundary condition is imposed at the interface. We solve only for the phase inside of the circle until a final time  $t_f = 0.03125$ . The simulation is carried out for different resolutions  $N = 16, 32, 64, 128$  corresponding to 4, 16, 64, 256 iterations, respectively. The reference solution is taken for  $N = 256$ . The results are summarized in Figure 3.8. As expected, the order of convergence of the error in full cells is close to 2 while the order of convergence in mixed cells is slightly less than 2. This drop in order in mixed cells is due to the assumption of a piece-wise linear interface approximation as well as the accumulation of errors of the bi-quadratic interpolation. The maximal errors are localized in cells where the wetted area is small (typically smaller than 5% of  $\Delta^2$ ). Nevertheless, the global order of convergence in all cells is exactly 2.

2. Similar to the first case, we initialize a solid circle of radius  $R = 0.75$ . The domain size are the same as well as the considered grid resolutions than in the previous. This time, we solve outside of the circle with a Dirichlet boundary condition  $T_D = 1$  at the interface and insulated boundary conditions at the domain boundary. The results, presented in Figure 3.9 are similar to case 1 with a slight drop in absolute error. This is due to the fact that there are less points per diameter than previously. This case validates the implementation of the Neumann boundary condition imposed at the domain boundaries.
3. In this third case, we initialize a square of area  $1.6 \times 1.6$  in a  $2 \times 2$  domain. The level set function is defined as

$$\phi(x, y) = \max((x - 0.8), -(x + 0.8), (y - 0.8), -(y + 0.8)).$$

We impose a Dirichlet boundary condition  $T_D = 1$  and solve inside of the square until the same final time  $t_f = 0.03125$  with the same resolutions considered previously. In Figure 3.10, we can see that the maximal errors are located at the corners. The order of convergence for full cells is similar to the circle cases as well as for all cells. This case exhibits the robustness of the method when dealing with mesh aligned geometries as explained in Section 3.2.

4. Finally, in the last case, we consider a crystal in a  $2 \times 2$  domain where the level set function is defined as

$$\phi(x, y) = \sqrt{x^2 + y^2} - R - 0.2 \cos(6\alpha),$$

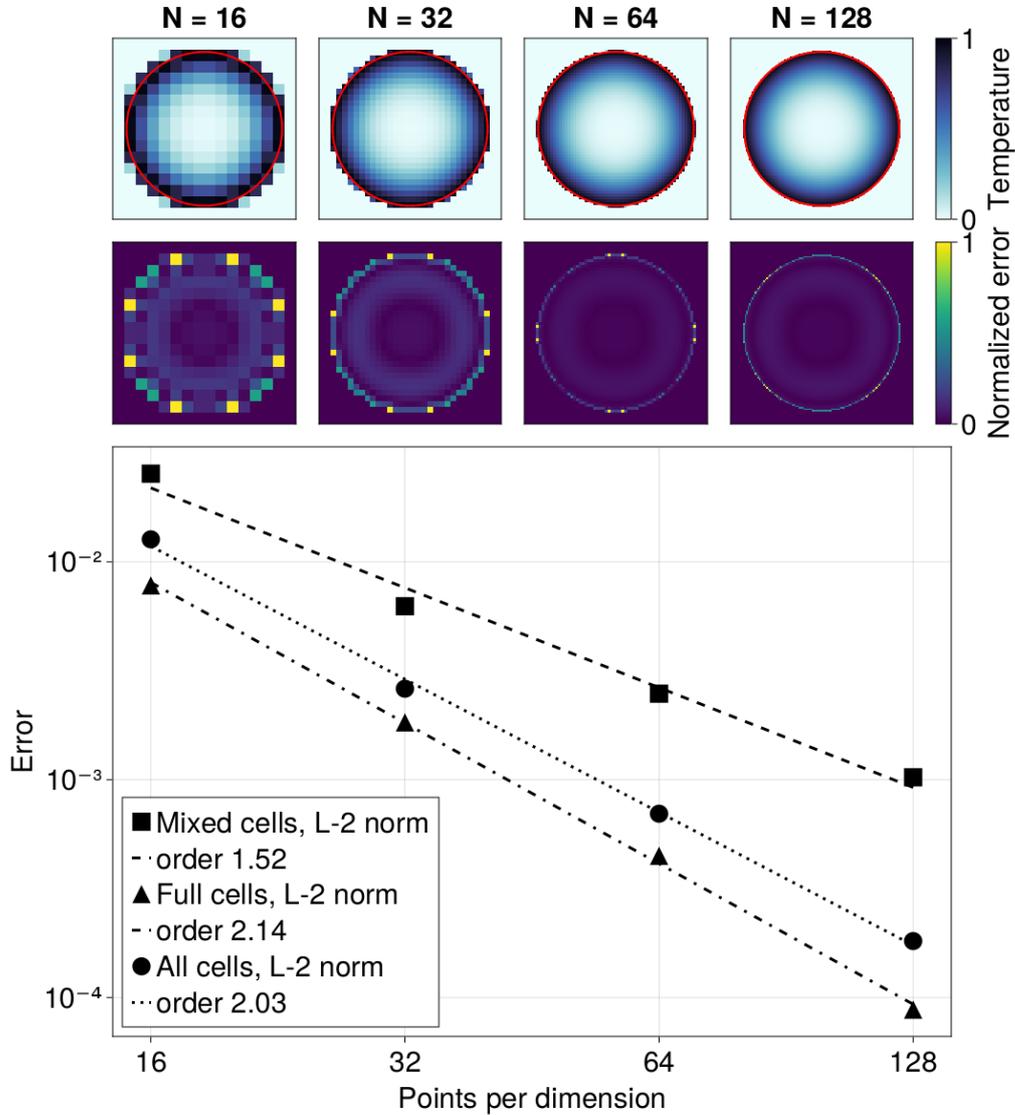
where  $\alpha$  is the angle of the interface with respect to the  $x$  axis and  $R = 0.7$ .

At the interface, we impose the Gibbs-Thomson relation

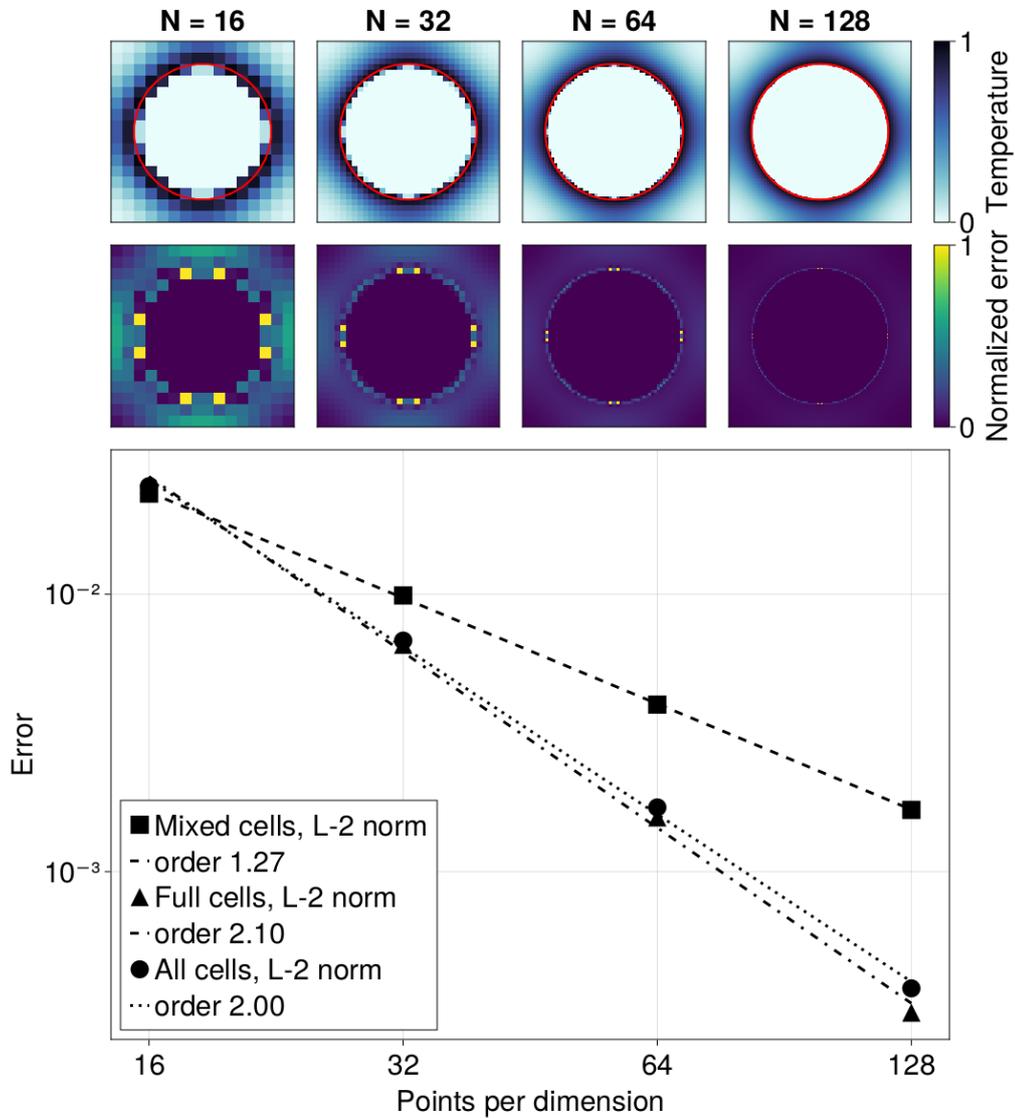
$$T_D = \varepsilon_\kappa \kappa,$$

with  $\kappa$  the curvature and  $\varepsilon_\kappa = 0.01$ . The resulting temperature field will now depend on the sign and amplitude of  $\kappa$ . We solve only for the phase inside of the circle until a final time  $t_f = 0.0078125$ . The simulation is performed for different resolutions  $N = 32, 64, 128, 256$  corresponding to 4, 16, 64, 256 iterations respectively. The reference solution is taken for  $N = 512$ . In Figure 3.11, we can observe a drop in order of convergence for full cells with respect to the cases where  $T_D$  was constant. This is explained by the accuracy of the curvature computation (Equation 1.8). The error is maximal in regions where the radius of curvature is large, where the interface is quasi aligned with the grid.

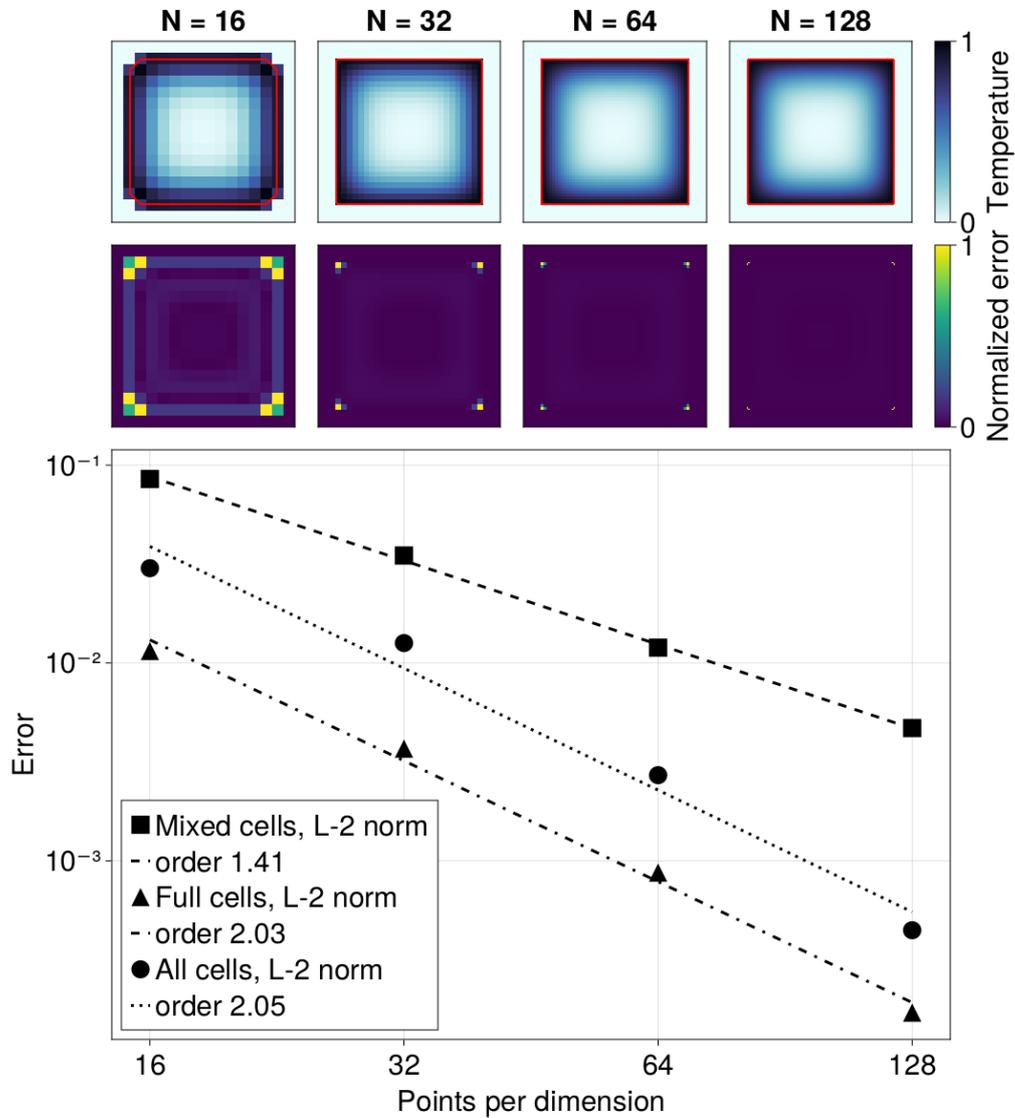
With these validation cases, we close the chapter on the Cut Cell method for diffusive transport. In the next chapter, we describe the rest of the numerical steps of the two-phase Stefan problem.



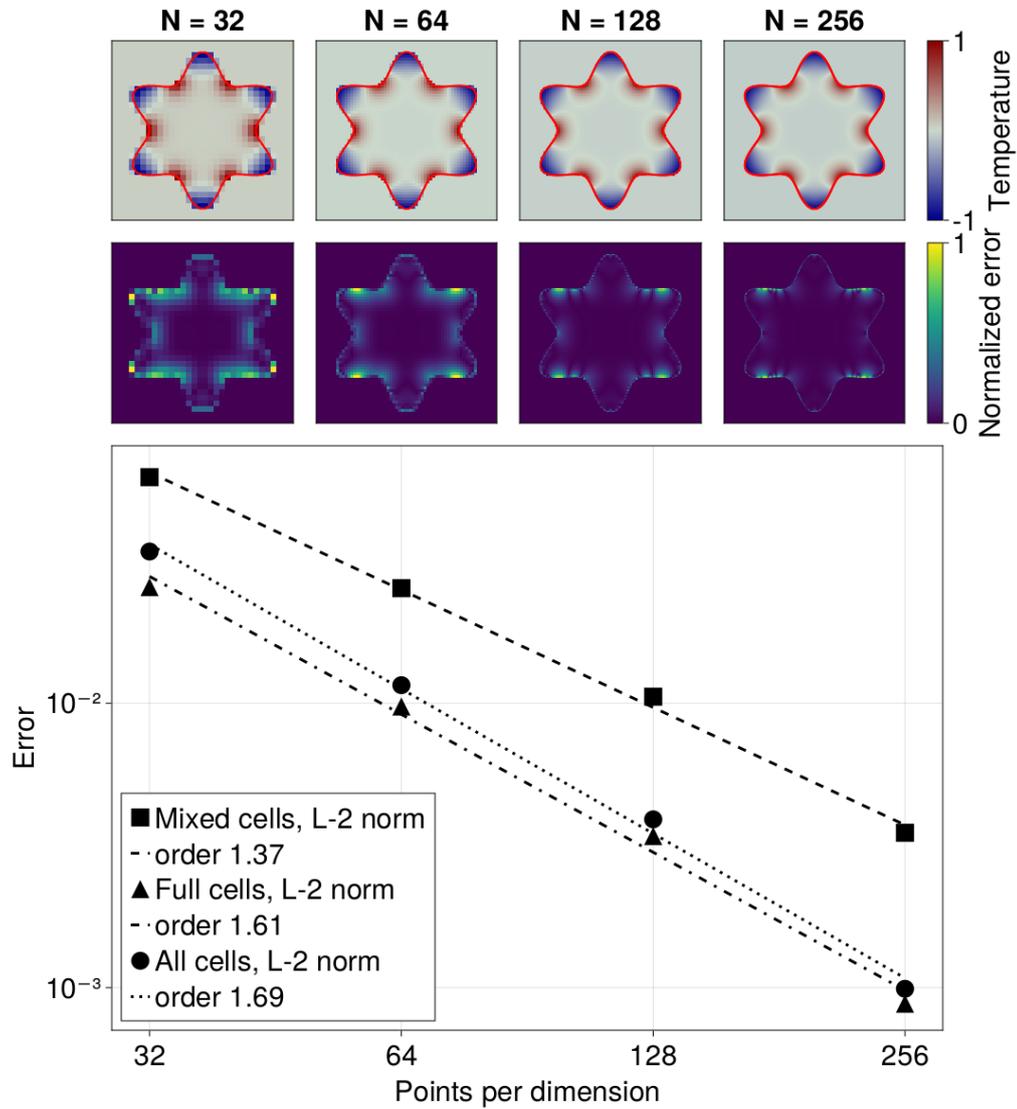
**Figure 3.8:** Convergence study of the Cut Cell method coupled with a Crank-Nicolson scheme when solving the heat equation inside a stationary circle with a Dirichlet boundary condition  $T_D = 1$  imposed at the interface. The top figures show the position of the interface in red and the temperature field at final time  $t_f = 0.03125$  for  $N = 16, 32, 64, 128$ . The middle figure shows the normalized error in temperature field with respect to the reference solution taken for  $N = 256$ . The bottom figure shows the convergence rate of the method in mixed cells, full cells and in all cells.



**Figure 3.9:** Convergence study of the Cut Cell method coupled with a Crank-Nicolson scheme when solving the heat equation outside a stationary circle with a Dirichlet boundary condition  $T_D = 1$  imposed at the interface and insulated boundary conditions. The top figures show the position of the interface in red and the temperature field at final time  $t_f = 0.03125$  for  $N = 16, 32, 64, 128$ . The middle figure shows the normalized error in temperature field with respect to the reference solution taken for  $N = 256$ . The bottom figure shows the convergence rate of the method in mixed cells, full cells and in all cells.



**Figure 3.10:** Convergence study of the Cut Cell method coupled with a Crank-Nicolson scheme when solving the heat equation inside a stationary square with a Dirichlet boundary condition  $T_D = 1$  imposed at the interface. The top figures show the position of the interface in red and the temperature field at final time  $t_f = 0.03125$  for  $N = 16, 32, 64, 128$ . The middle figure shows the normalized error in temperature field with respect to the reference solution taken for  $N = 256$ . The bottom figure shows the convergence rate of the method in mixed cells, full cells and in all cells.



**Figure 3.11:** Convergence study of the Cut Cell method coupled with a Crank-Nicolson scheme when solving the heat equation inside a stationary crystal with a Dirichlet boundary condition  $T_D = \varepsilon_\kappa \kappa$  imposed at the interface. The top figures show the position of the interface in red and the temperature field at final time  $t_f = 0.0078125$  for  $N = 32, 64, 128, 256$ . The middle figure shows the normalized error in temperature field with respect to the reference solution taken for  $N = 512$ . The bottom figure shows the convergence rate of the method in mixed cells, full cells and in all cells.



# Numerical methods for forward and adjoint problems

---

## Contents

---

<b>4.1</b>	<b>Algorithm for solving the forward Stefan problem . . .</b>	<b>51</b>
<b>4.2</b>	<b>Phase-change velocity . . . . .</b>	<b>53</b>
4.2.1	Discrete Stefan condition . . . . .	53
4.2.2	Normal extension of the velocity field . . . . .	54
<b>4.3</b>	<b>The advection equation . . . . .</b>	<b>56</b>
4.3.1	Forward-Backward diffusion scheme . . . . .	56
4.3.2	Numerical validation . . . . .	59
<b>4.4</b>	<b>Reinitialization and initialization steps . . . . .</b>	<b>61</b>
4.4.1	Reinitialization algorithm . . . . .	61
4.4.2	Fresh and dead cells . . . . .	63
<b>4.5</b>	<b>Algorithm for solving the adjoint Stefan problem . . . .</b>	<b>64</b>

---

## 4.1 Algorithm for solving the forward Stefan problem

In this chapter, the methods used to solve both the forward and the adjoint Stefan problems are discussed.

The solution algorithm of the classical two-phase Stefan problem is split into seven main steps:

1. Computation of the geometrical information.
2. Solution of the heat equation in each phase.
3. Computation of the Stefan condition.
4. Extension of the velocity field normal to the interface.
5. Propagation of the level-set function.
6. Reinitialization of the level-set function (so as to satisfy the signed distance property).
7. Handling of the dead and fresh cells.

It should be noted that in cases where the flow in the liquid phase is considered, the algorithm is modified such that the Navier-Stokes equations in the Boussinesq approximation are solved prior to the convection-diffusion equation (Equation 1.11); thus before the second step. The solution of the Navier-Stokes equations in the Cut Cell framework are detailed in Appendix B.

Each of these steps and their corresponding convergence properties are presented in the following sections and summarized in Algorithm 4. While solving the forward problem, we store the temperature field, the interface position and the phase-change velocity for use in the adjoint algorithm (see Algorithm 6).

---

**Algorithm 4:** Generic algorithm for the solution of the forward problem
 

---

**input** :  $T^0, T_D, w, \phi^0, \mathbf{u}^0, P^0, t_f$

**output**:  $T, \phi$

Initialize the level set function with  $\phi^0$  and the temperature field with  $T^0$

Set the boundary conditions for  $T$  (Equation FP.d)

**if** *fluid flow* **then**

    Initialize the velocity field with  $\mathbf{u}^0$  and the pressure field with  $P^0$

    Set the boundary conditions for  $\mathbf{u}$  and  $P$

**end**

**repeat**

**1.** Compute the capacities and centroids (Section 3.3)

**if** *fluid flow* **then**

**2a.** Solve the Navier-Stokes equations  $\tilde{\mathbf{u}}^n \leftarrow \mathbf{u}^{n-1}$  (Appendix B)

**2b.** Solve the convection-diffusion equation  $\tilde{T}^n \leftarrow T^{n-1}, \tilde{\mathbf{u}}^n, T_D$

**else**

**2.** Solve the heat equations  $\tilde{T}^n \leftarrow T^{n-1}, T_D$  (Section 3.4)

**end**

**3.** Compute the Stefan condition  $v_S^n \leftarrow [\nabla \tilde{T}^n]_2^1$  (Section 4.2.1)

**4.** Extend the velocity field  $F^n \leftarrow v_S^n$  (Section 4.2.2)

**5.** Solve the advection equation  $\tilde{\phi}^n \leftarrow \phi^{n-1}, F^n$  (Section 4.3.1)

**6.** Reinitialize the level set function  $\phi^n \leftarrow \tilde{\phi}^n$  (Section 4.4.1)

**if** *fluid flow* **then**

**7.** Clean or Initialize dead or fresh cells  $T^n, \mathbf{u}^n \leftarrow \tilde{T}^n, \tilde{\mathbf{u}}^n$

**else**

**7.** Clean or Initialize dead or fresh cells  $T^n \leftarrow \tilde{T}^n$  (Section 4.4.2)

**end**

**until** *final time*  $t_f \leftarrow 0$ ;

---

## 4.2 Phase-change velocity

### 4.2.1 Discrete Stefan condition

As stated previously, the motion of the interface is solely dependent on the jump in the temperature gradient. It is therefore important to compute the normal gradient in temperature of each phase accurately, and to this end, the Johansen-Colella method [Johansen & Colella 1998] is used

$$\nabla T|_{\Gamma} = \frac{1}{d_B - d_A} \left( \frac{d_B}{d_A} (T_D - T_A^*) - \frac{d_A}{d_B} (T_D - T_B^*) \right), \quad (4.1)$$

with  $T_D$  the Dirichlet value imposed at the interface,  $T_A^*$  and  $T_B^*$  the interpolated values of the temperature field on points **A** and **B** respectively, and  $d_A$  and  $d_B$  the distances between the interface centroid to **A** and **B** respectively. The algorithm for computing the gradient in one phase is as follows:

1. A shifted  $3 \times 3$  stencil is chosen, as shown in Figure 4.1.
2. A line from the interface centroid is cast in the normal direction  $\mathbf{n}$ .
3. The crossing points **A** and **B** of this line and the vertical (or horizontal, depending on the normal orientation) segments of the neighboring 3 points are identified and the distances  $d_A$  and  $d_B$  are computed.
4. The values  $T_A^*$  and  $T_B^*$  are interpolated using  $T_A^1, T_A^2, T_A^3$  and  $T_B^1, T_B^2, T_B^3$ , respectively

Once the normal gradient is computed in each phase, using Equation. 4.1, the jump is computed as

$$[\nabla T]_2^1 = \nabla T_1|_{\Gamma} - \nabla T_2|_{\Gamma}. \quad (4.2)$$

The discrete velocities of the front in the partial cells are initialized with this jump and will be used as the boundary condition in the velocity extension algorithm. To validate the method within our Cut Cell framework, we consider a stationary circle of radius  $R = 0.5$  in a  $1 \times 1$  domain and we initialize the temperature field with a similarity solution of the heat equation

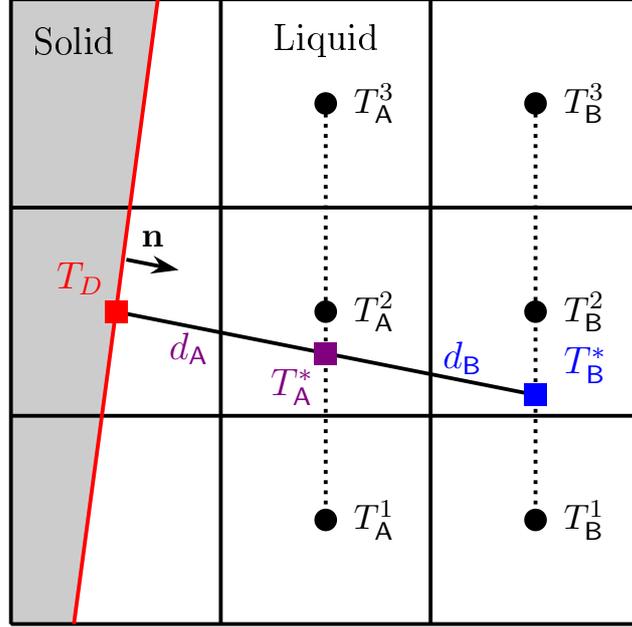
$$T(r) = \begin{cases} T_{\infty} \left( 1 - \frac{f(r)}{f(R)} \right), & r > R \\ 0, & r < R \end{cases} \quad (4.3)$$

with  $T_{\infty} = -0.5$  a given under-cooling temperature, and

$$f(r) = E_1(1/4 r^2)$$

where

$$E_1(t) = \int_x^{\infty} \frac{e^{-t}}{t} dt$$



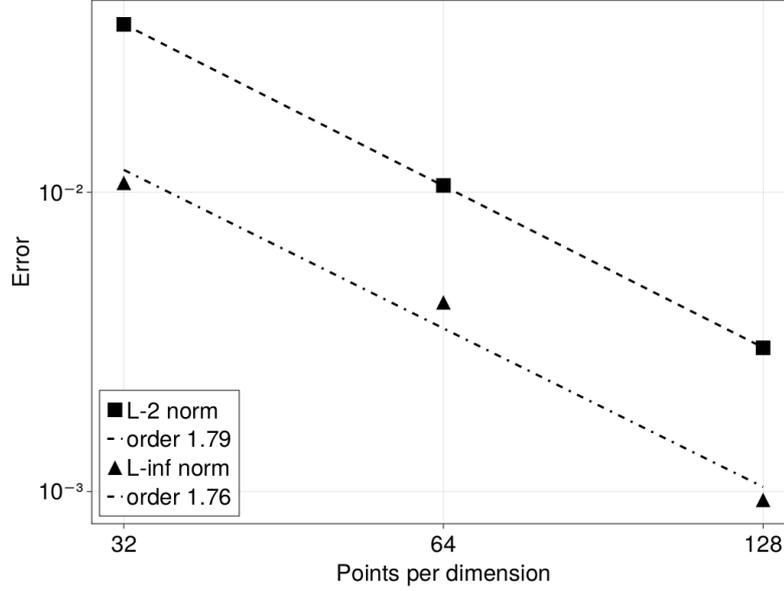
**Figure 4.1:** Schematic of gradient calculation. The Dirichlet value  $T_d$  is imposed at the interface centroid in the partial cell and the temperatures  $T_A$  and  $T_B$  are determined via a quadratic interpolation from the neighboring 3 points in the vertical direction (dotted lines).

The discrete velocities at the front are computed for different resolutions  $N = 32, 64, 128$ , and the resolution  $256 \times 256$  is used as the reference solution for the convergence study. Figure 4.2 shows that the gradient extraction algorithm coupled with the interface location method results in close to second-order accuracy in both  $L_2$  and  $L_\infty$  norms.

#### 4.2.2 Normal extension of the velocity field

As highlighted in Section 1.2, the front velocity needs to be extended away from the interface. The most natural algorithm is to let  $v_S$  be a constant along the lines normal to  $\Gamma$ . To achieve this, the method described in [Peng *et al.* 1999] is adopted here. Using this approach, the velocity is extended in the normal direction by solving the following hyperbolic partial differential equation

$$\begin{cases} \frac{\partial F}{\partial t^*} + S(\phi) \frac{\nabla \phi}{|\nabla \phi|} \cdot \nabla F = 0 & \text{in } \Omega \\ F(x, 0) = v_S & \text{on } \Gamma \end{cases} \quad (4.4)$$



**Figure 4.2:**  $L_2$  and  $L_\infty$  norms of the error in the gradient computation with the Johansen-Colella method.

where  $F$  is the extended velocity field equal to  $v_S$  at the front,  $t^*$  denotes a pseudo-time and  $S(\phi)$  is the signature function

$$S(\phi) = \begin{cases} -1 & \text{if } \phi < 0 \\ 0 & \text{if } \phi = 0 \\ +1 & \text{if } \phi > 0 \end{cases} \quad (4.5)$$

Equation 4.4 is then discretized using a first order upwind scheme and integrated in time by a forward Euler method until steady state. Taking  $\mathbf{n}$  as the normal vector defined as

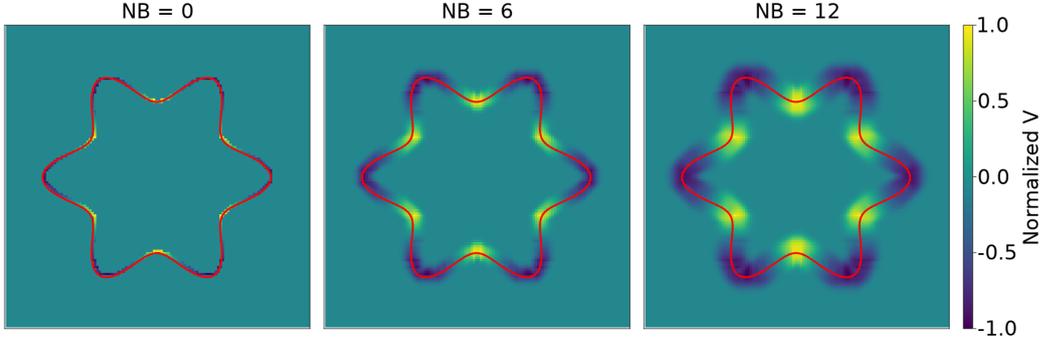
$$\mathbf{n} = (n_x, n_y) = \left( \phi_x / \sqrt{(\phi_x^2 + \phi_y^2)}, \phi_y / \sqrt{(\phi_x^2 + \phi_y^2)} \right), \quad (4.6)$$

the discretisation leads to

$$F_{ij}^{n+1} = F_{ij}^n - \tau^* \left( (S_{ij} n_{ij}^x)^+ \frac{F_{ij} - F_{i-1j}}{\Delta} + (S_{ij} n_{ij}^x)^- \frac{F_{i+1j} - F_{ij}}{\Delta} \right. \\ \left. + (S_{ij} n_{ij}^y)^+ \frac{F_{ij} - F_{ij-1}}{\Delta} + (S_{ij} n_{ij}^y)^- \frac{F_{ij+1} - F_{ij}}{\Delta} \right) \quad (4.7)$$

where  $\Delta$  is the uniform grid spacing,  $(x)^+ = \max(0, x)$  and  $(x)^- = \min(0, x)$ , and the time step  $\tau^*$  is chosen so that  $\tau^*/\Delta^2 = 0.45$ . The pseudo-time span in the velocity extension algorithm is purely fictitious and the number of iterations in Equation 4.7 corresponds to the width of the narrow-band (NB) around the 0-level set where the velocities are initialized. Figure 4.3 shows an example of an initial velocity field for different narrow-band widths, after one iteration of the heat

equation, where the temperature at the interface  $T_D = \varepsilon_\kappa \kappa$  depends only on the curvature. The computed discrete velocities are positive in the kinks and negative in the tips driving the initial crystal towards a circular shape.



**Figure 4.3:** Velocity fields of a crystal-shape geometry. The red curves represent the interface location and the velocity fields correspond, from left to right, to narrow bands widths of 0, 6 and 12.

## 4.3 The advection equation

### 4.3.1 Forward-Backward diffusion scheme

A semi-implicit scheme described in [Mikula *et al.* 2014, Mikula & Ohlberger 2010] is used here to solve the level set advection equation (Equation 1.9). Using this method, the equation is written in an alternative form

$$\frac{\partial \phi}{\partial t} + F \frac{\nabla \phi}{|\nabla \phi|} \cdot \nabla \phi = 0, \quad (4.8)$$

which is then divided into conservative and non-conservative terms

$$\frac{\partial \phi}{\partial t} + \underbrace{\nabla \cdot \left( F \phi \frac{\nabla \phi}{|\nabla \phi|} \right)}_a - \underbrace{\phi \nabla \cdot \left( F \frac{\nabla \phi}{|\nabla \phi|} \right)}_b = 0, \quad (4.9)$$

resulting in a second order partial differential equation akin to a weighted diffusion equation. The first term ( $a$ ) has a diffusion coefficient  $F\phi$  that depends on the solution and represents a nonlinear curvature flow whereas in the second term ( $b$ ) the solution is multiplied by the curvature of its level-sets. The main idea behind this scheme is to distinguish two cases: if the product  $F\phi$  is negative (positive, respectively) then  $a$  represents a forward (backward, respectively) diffusion and  $b$  represents a backward (forward, respectively) diffusion. The forward diffusion is treated implicitly while the backward diffusion is treated explicitly leading to a semi-implicit scheme with a diffusive CFL number.

In order to discretize Equation 4.9, we use the same notation as

in [Mikula *et al.* 2014]. Consider  $p$  to be a finite volume of a cell and  $e_{pq}$  the edge between  $p$  and  $q$ ,  $q \in \mathcal{N}(p)$  where  $\mathcal{N}(p)$  is the set of neighboring finite volumes. The length of  $e_{pq}$  is normalized to 1. Let  $\mathbf{n}_{pq}$  be the outer normal vector to  $e_{pq}$  with respect to  $p$ . Finally, let us denote  $\bar{\phi}_p$  the constant reconstruction of  $\phi_p$  in the finite volume  $p$  and  $\bar{\phi}_{pq}$  the constant reconstruction of  $\phi$  on  $e_{pq}$ . Integrating volume  $p$ , Equation 4.9 yields

$$\int_p \frac{\partial \phi}{\partial t} dx + \int_p \nabla \cdot \left( F \phi \frac{\nabla \phi}{|\nabla \phi|} \right) dx - \int_p \phi \nabla \cdot \left( F \frac{\nabla \phi}{|\nabla \phi|} \right) dx = 0. \quad (4.10)$$

Applying Stokes' theorem and using the constant reconstructions of  $\phi$ , we obtain

$$\begin{aligned} \int_p \frac{\partial \phi}{\partial t} dx + \sum_{q \in \mathcal{N}(p)} \bar{\phi}_{pq} \int_{e_{pq}} F \frac{1}{|\nabla \phi|} \nabla \phi \cdot \mathbf{n}_{pq} ds \\ - \sum_{q \in \mathcal{N}(p)} \bar{\phi}_p \int_{e_{pq}} F \phi \frac{1}{|\nabla \phi|} \nabla \phi \cdot \mathbf{n}_{pq} ds = 0. \end{aligned}$$

Let us denote  $|\nabla \phi_{pq}|$  the reconstructed Hamiltonian  $|\nabla \phi|$  on the edge  $e_{pq}$  and  $(\phi_q - \phi_p)/1$  the normal derivative  $\nabla \phi \cdot \mathbf{n}_{pq}$  on the same edge. We obtain the following expression

$$\begin{aligned} \int_p \frac{\partial \phi}{\partial t} dx + \sum_{q \in \mathcal{N}(p)} \frac{F \bar{\phi}_{pq}}{|\nabla \phi_{pq}|} (\phi_q - \phi_p) \\ - \sum_{q \in \mathcal{N}(p)} \frac{F \bar{\phi}_p}{|\nabla \phi_{pq}|} (\phi_q - \phi_p) = 0, \end{aligned}$$

leading to

$$\int_p \frac{\partial \phi}{\partial t} dx + \sum_{q \in \mathcal{N}(p)} \frac{F(\bar{\phi}_p - \bar{\phi}_{pq})}{|\nabla \phi_{pq}|} (\phi_p - \phi_q) = 0. \quad (4.11)$$

Looking at the term  $F(\bar{\phi}_p - \bar{\phi}_{pq})$ , we can distinguish two cases:

- If the term is positive, we have a 'forward diffusion' or inflow towards the cell.
- If the term is negative, we have a 'backward diffusion' or outflow from the cell.

We therefore define the diffusion coefficient  $a_{pq}$  as

$$a_{pq} = \frac{F(\bar{\phi}_p - \bar{\phi}_{pq})}{|\nabla \phi_{pq}|}, \quad (4.12)$$

and the related dominant forward and backward diffusion parts as

$$a_{pq}^f = \max(a_{pq}, 0), \quad a_{pq}^b = \min(a_{pq}, 0). \quad (4.13)$$

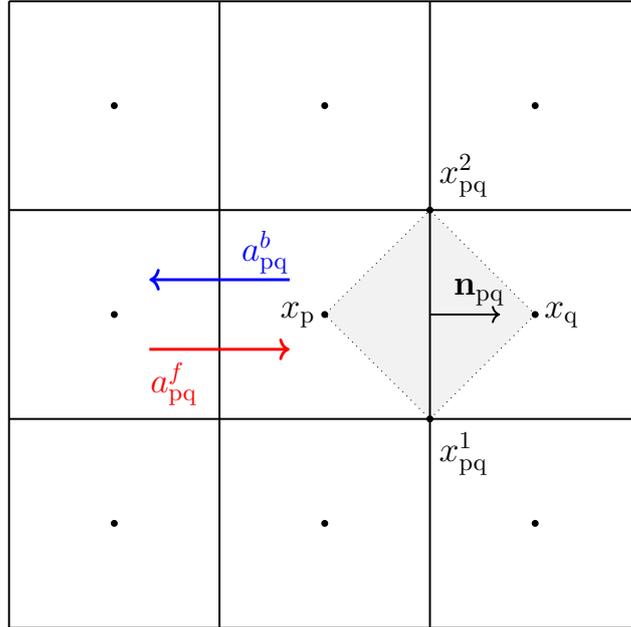
Using a backward Euler time discretization, taking the forward contribution explicitly and the backward contribution implicitly, Equation 4.11 gives the following

linear system

$$\underbrace{\phi_p^n + \frac{\tau}{\Delta^2} \sum_{q \in \mathcal{N}(p)} a_{pq}^f (\phi_p^n - \phi_q^n)}_{\text{implicit}} = \phi_p^{n-1} + \underbrace{\frac{\tau}{\Delta^2} \sum_{q \in \mathcal{N}(p)} a_{pq}^b (\phi_p^{n-1} - \phi_q^{n-1})}_{\text{explicit}} \quad (4.14)$$

where  $\tau$  is the time step,  $\Delta$  the uniform grid spacing and  $n$  a given time step. The question remaining is how to tackle the reconstruction of  $\bar{\phi}_p$ ,  $\bar{\phi}_{pq}$  and  $|\nabla \phi_{pq}|$ . The latter is constructed using the diamond-cell strategy described in [Mikula & Ohlberger 2010]. Consider  $\phi_{pq}^1$  and  $\phi_{pq}^2$  the values of  $\phi$  located at the end points  $x_{pq}^1$  and  $x_{pq}^2$  of the edge  $e_{pq}$  as shown in Figure 4.4.  $\phi_{pq}^1$  and  $\phi_{pq}^2$  are defined as the average of the 4 adjacent values of  $\phi$ . Bearing this in mind, we can define the reconstructed Hamiltonian as

$$|\nabla \phi_{pq}| = \left[ \left( \frac{\phi_p - \phi_q}{\Delta} \right)^2 + \left( \frac{\phi_{pq}^1 - \phi_{pq}^2}{\Delta} \right)^2 \right]^{1/2} \quad (4.15)$$



**Figure 4.4:** Schematic of the diamond-cell strategy for the calculation of the Hamiltonian. The red arrow corresponds to the forward coefficient and the blue arrow to the backward coefficient.

The remaining reconstructions are carried out as follows:

1. As a first step we consider the simplest reconstruction

$$\bar{\phi}_p = \phi_p, \quad \bar{\phi}_{pq} = \frac{1}{2}(\phi_p + \phi_q). \quad (4.16)$$

2. For every finite volume  $p$  we determine the characteristic of the diffusion coefficient by computing the sum of the forward and backward contributions

$$S_p^f = \sum_{q \in \mathcal{N}(p)} a_{pq}^f, \quad S_p^b = \sum_{q \in \mathcal{N}(p)} a_{pq}^b. \quad (4.17)$$

3. If the forward diffusion is dominant (ie.  $S_p^f > -S_p^b$ ) then no further steps are needed as the discretization will lean towards the implicit part.
4. On the other hand, if the backward diffusion is dominant (ie.  $S_p^f < -S_p^b$ ), we need to smooth the reconstructed solution for stability, using the following formula

$$\bar{\phi}_p = \frac{1}{4} \sum_{q \in \mathcal{N}(p)} (\phi_p + \phi_q + \phi_{pq}^1 + \phi_{pq}^2), \quad \bar{\phi}_{pq} = \frac{1}{4} \sum_{q \in \mathcal{N}(p)} \phi_{pq}. \quad (4.18)$$

In case of step 4, the diffusion coefficients are recomputed using the new reconstructions. Algorithm 5 summarizes the steps to construct the linear system at each time step. The resulting matrix is a pentadiagonal matrix as in the Cut Cell method coupled with a Crank-Nicolson scheme (see Section 3.4).

### 4.3.2 Numerical validation

This scheme allows us the relaxation of the CFL condition, usually present in most of level set methods. To check the robustness of the method for a  $\tau/\Delta^2$  ratio exceeding the usual CFL condition, we consider an initial level set function  $\phi_0$  in a  $2 \times 2$  domain, given by

$$\phi_0(x, y) = \sqrt{x^2 + y^2} - R, \quad (4.19)$$

with  $R = 0.8$  the radius of the 0-level set. The velocity field is set to  $F = -1$  in the whole domain and we run the simulation until the final time  $t_f = 0.3625$  for different resolutions  $N = 16, 32, 64$  and different CFL numbers ranging from 1 to 16. The  $L_2$  norm of the error with respect to the analytical solution is computed in the whole domain. Results in Figure 4.5 show a second-order accuracy for any given CFL number for the retracting circle case.

**Algorithm 5:** Forward-Backward diffusion scheme**input :**  $\phi^{n-1}, F$ **output:**  $\phi^n$ Set the boundary conditions for  $\phi^{n-1}$ **foreach** *element*  $x_p \in \Omega \setminus \partial\Omega$  **do**

Use the reconstructions

$$\bar{\phi}_p = \phi_p, \quad \bar{\phi}_{pq} = \frac{1}{2}(\phi_p + \phi_q)$$

Compute the Hamiltonian using the diamond-cell strategy

$$|\nabla\phi_{pq}| = \left[ \left( \frac{\phi_p - \phi_q}{\Delta} \right)^2 + \left( \frac{\phi_{pq}^1 - \phi_{pq}^2}{\Delta} \right)^2 \right]^{1/2}$$

Compute the forward and backward diffusion coefficients

$$a_{pq} = \frac{F(\bar{\phi}_p - \bar{\phi}_{pq})}{|\nabla\phi_{pq}|}$$

$$a_{pq}^f = \max(a_{pq}, 0), \quad a_{pq}^b = \min(a_{pq}, 0)$$

Compute the dominant contribution

$$S_p^f = \sum_{q \in \mathcal{N}(p)} a_{pq}^f, \quad S_p^b = \sum_{q \in \mathcal{N}(p)} a_{pq}^b$$

**if**  $S_p^f < -S_p^b$  **then**

Use the reconstructions

$$\bar{\phi}_p = \frac{1}{4} \sum_{q \in \mathcal{N}(p)} (\phi_p + \phi_q + \phi_{pq}^1 + \phi_{pq}^2), \quad \bar{\phi}_{pq} = \frac{1}{4} \sum_{q \in \mathcal{N}(p)} \phi_{pq}$$

Recompute the forward and backward diffusion coefficients

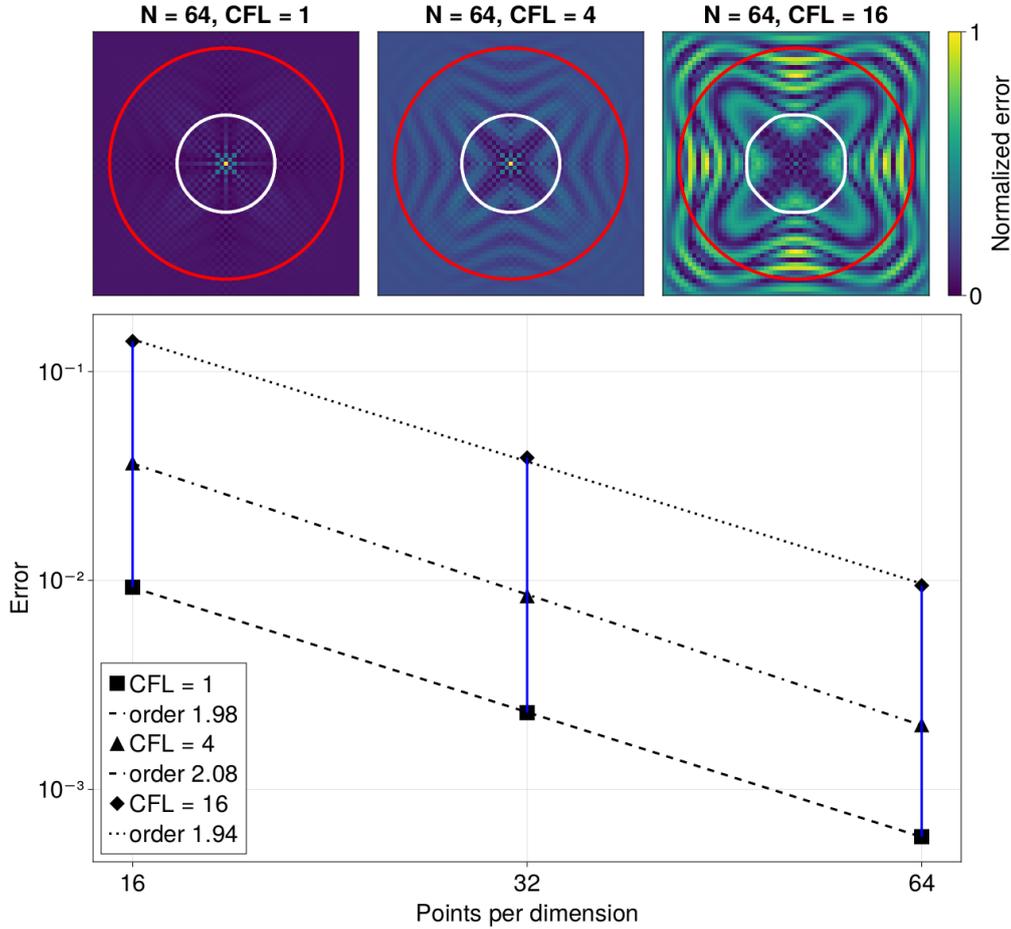
$$a_{pq} = \frac{F(\bar{\phi}_p - \bar{\phi}_{pq})}{|\nabla\phi_{pq}|}$$

$$a_{pq}^f = \max(a_{pq}, 0), \quad a_{pq}^b = \min(a_{pq}, 0)$$

**end****end**

Solve the linear system

$$\phi_p^n + \frac{\tau}{\Delta^2} \sum_{q \in \mathcal{N}(p)} a_{pq}^f (\phi_p^n - \phi_q^n) = \phi_p^{n-1} + \frac{\tau}{\Delta^2} \sum_{q \in \mathcal{N}(p)} a_{pq}^b (\phi_p^{n-1} - \phi_q^{n-1})$$



**Figure 4.5:** The top figures represent the normalized error field for different CFL = 1, 4, 16 and fixed  $N = 64$ . The red (white, respectively) curve represents the initial (final, respectively) 0-level set. In the bottom figure the error in  $L_2$  norm is plotted for different resolutions and CFL numbers. The vertical blue lines correspond to a fixed number of points for varying CFL numbers.

## 4.4 Reinitialization and initialization steps

### 4.4.1 Reinitialization algorithm

Depending on the case, usually after one or more iterations of the time advancement scheme, the level set function will cease to be an exact signed distance function, necessitating a reinitialization step to enforce this criterion at the front  $\Gamma(t)$ . Given a function  $\phi_0$ , which is not a signed distance function, it can be evolved into a function  $\phi$  by solving the Eikonal equation

$$\begin{cases} \frac{\partial \phi}{\partial t^*} = S(\phi_0)(1 - |\nabla \phi|) & \text{in } \Omega, \\ \phi(x, \tau) = 0 & \text{on } \Gamma(t), \end{cases} \quad (4.20)$$

where  $t^*$  again denotes a pseudo-time,  $S(\phi_0)$  is the signature function (Equation 4.5), and  $\phi$  the signed distance function (once steady-state is reached). There exist many numerical methods for solving the Eikonal equation and, here we have adopted that of Min [Min 2010], also recently used in the work of Limare [Limare *et al.* 2023]. This approach relies on a second order ENO spatial discretization with sub-cell resolution near the interface. The one-sided ENO finite difference (in the  $x$  direction only) yields

$$\begin{aligned} D_x^+ \phi_{ij} &= \frac{\phi_{i+1,j} - \phi_{ij}}{\Delta} - \frac{\Delta}{2} \min\text{mod}(D_{xx}\phi_{ij}, D_{xx}\phi_{i+1,j}), \\ D_x^- \phi_{ij} &= \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta} + \frac{\Delta}{2} \min\text{mod}(D_{xx}\phi_{ij}, D_{xx}\phi_{i-1,j}), \end{aligned} \quad (4.21)$$

where  $D_{xx}\phi_{ij} = (\phi_{i-1,j} - 2\phi_{ij} + \phi_{i+1,j})/\Delta^2$  is the second order derivative of  $\phi_{ij}$  and the ‘minmod’ limiter is zero when both arguments have opposite signs, and takes the argument of smallest absolute value otherwise. The numerical Hamiltonian  $|\phi|$  is computed as follows

$$|\nabla\phi|_{ij} \simeq H_G(D_x^+ \phi_{ij}, D_x^- \phi_{ij}, D_y^+ \phi_{ij}, D_y^- \phi_{ij}), \quad (4.22)$$

where function  $H_g$  is given by

$$H_G(a, b, c, d) = \begin{cases} \sqrt{\max((a^-)^2, (b^+)^2) + \max((c^-)^2, (d^+)^2)} & \text{when } S(\phi^0) \geq 0, \\ \sqrt{\max((a^+)^2, (b^-)^2) + \max((c^+)^2, (d^-)^2)} & \text{when } S(\phi^0) < 0. \end{cases} \quad (4.23)$$

Near the interface, the finite differences need to be modified in order to impose  $\phi = 0$  where  $\phi_0 = 0$ . To this end, a quadratic ENO polynomial interpolation gives

$$D_x^+ \phi_{ij} = \frac{0 - \phi_{ij}}{\Delta^+} - \frac{\Delta^+}{2} \min\text{mod}(D_{xx}\phi_{ij}, D_{xx}\phi_{i+1,j}), \quad (4.24)$$

where

$$\Delta^+ = \begin{cases} \Delta \left( \frac{1}{2} + \frac{\phi_{ij}^0 - \phi_{i+1,j}^0 - S(\phi_{ij}^0 - \phi_{i+1,j}^0) \sqrt{D}}{\phi_{xx}^0} \right) & \text{where } |\phi_{xx}^0| > \varepsilon, \\ \Delta \frac{\phi_{ij}^0}{\phi_{ij}^0 - \phi_{i+1,j}^0} & \text{elsewhere,} \end{cases} \quad (4.25)$$

with

$$\phi_{xx}^0 = \min\text{mod}(\phi_{i-1,j}^0 - 2\phi_{ij}^0 + \phi_{i+1,j}^0, \phi_{ij}^0 - 2\phi_{i-1,j}^0 + \phi_{i-2,j}^0),$$

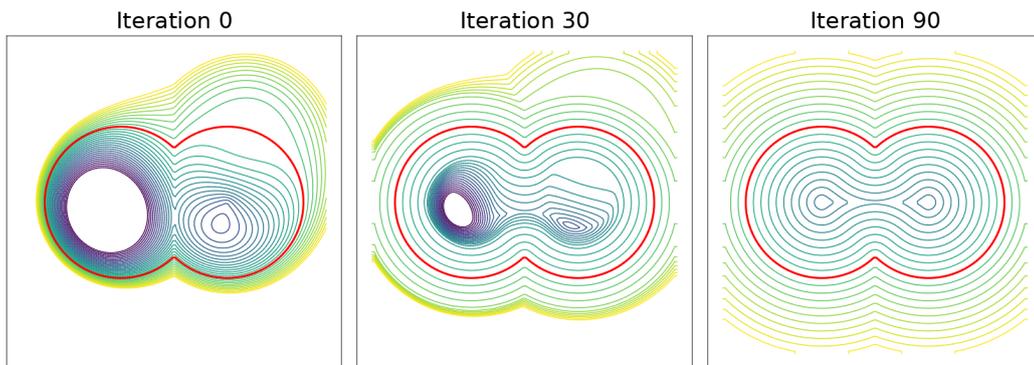
and

$$D = (\phi_{xx}^0/2 - \phi_{ij}^0 - \phi_{i-1,j}^0)^2 - 4\phi_{ij}^0\phi_{i-1,j}^0.$$

The negative one-sided ENO difference  $D_x^-$  is obtained similarly. A forward Euler scheme is then used for time advancement. This Hamiltonian extraction will also be used for check-pointing in the adjoint problem (Equations AP.e and AP.f) when needed. We validate the method by initializing a perturbed solution, similar to the test case found in [Min 2010], in a  $4 \times 4$  domain

$$\phi_0(x, y) = ((x - 1)^2 + (y - 1)^2 + 0.1) \times (\sqrt{(x \pm a)^2 + y^2} - R), \quad (4.26)$$

where  $R = 1$  and  $a = 0.7$ . Figure 4.6 shows the initial perturbed level set function in a  $128 \times 128$  grid converging towards a signed distance function with increasing number of iterations in pseudo-time.



**Figure 4.6:** The -2 to 1 level sets of the function  $\phi$  are shown with a 0.1 step. The red curve represents the 0-level set. After 90 iterations the level set function is a true signed distance function.

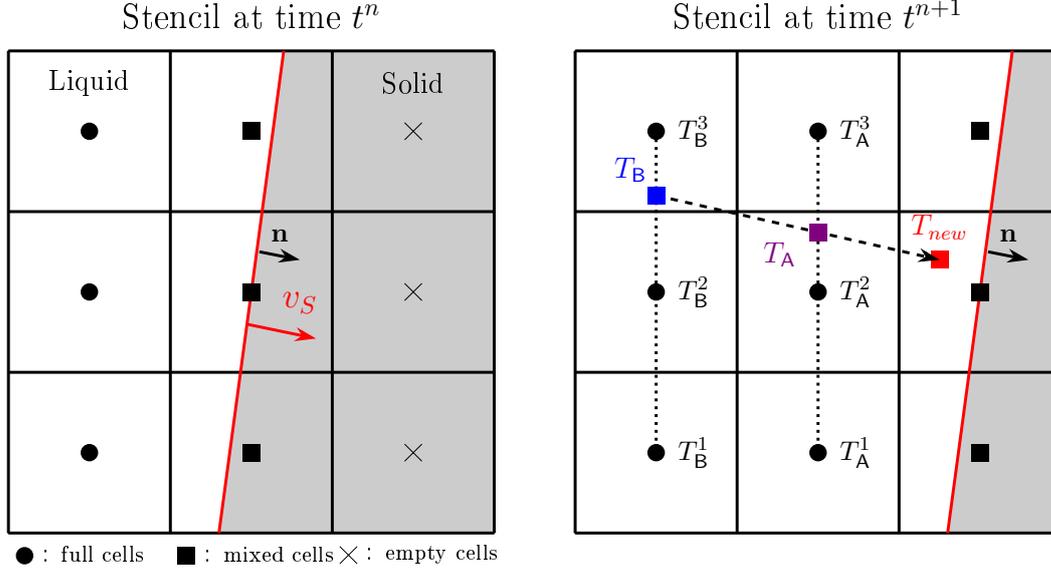
#### 4.4.2 Fresh and dead cells

The last step of the method presented in this study is the treatment of fresh and dead cells. As the interface moves through the Cartesian grid, the temperature field needs to be initialized according to the front position. Two cases can be distinguished

- A full or partial cell becomes an empty cell.
- An empty cell becomes a partial cell.

In the first case, the previous temperature value is simply set to 0. In the second case, the temperature previously non-existing needs to be initialized. The algorithm is similar to that of the Stefan condition (Section 4.2.1)

1. A shifted  $3 \times 3$  stencil is chosen, as shown in Figure 4.7.
2. A line from the interface centroid is cast in the opposite normal direction  $-\mathbf{n}$ .
3. The crossing points **A** and **B** of this line and the vertical (or horizontal depending on the normal orientation) segments of the neighboring 3 points, are identified.



**Figure 4.7:** Example of empty cells becoming partial cells from the point of view of the solid phase. The interface at time  $t^n$  moves in the normal direction with speed  $V$ . At time  $t^{n+1}$ , the newly initialized value  $T_{new}$  located at the partial cell centroid is extrapolated from  $T_A$  and  $T_B$ .

4. The values  $T_A^*$  and  $T_B^*$  are interpolated using  $T_A^1, T_A^2, T_A^3$  and  $T_B^1, T_B^2, T_B^3$  respectively.
5. The coordinates  $\mathbf{o} = (x_{new}, y_{new})$  of the barycenter of the cell to initialize is located.

The last step is done by using the discrete face capacities defined in Section 3.3. An orthonormal coordinate system  $\mathcal{R} = (\mathbf{o}, (x', \mathbf{n}))$  similar to the parabola-fitted curvature found in [Popinet 2009] is then defined. Finally, the new temperature value,  $T_{new}$ , is linearly extrapolated by solving for  $a_0$  and  $a_1$  in the resulting system of equations

$$\begin{cases} T_A^* = a_0 x'_A + a_1, \\ T_B^* = a_0 x'_B + a_1, \end{cases} \quad (4.27)$$

where  $a_1 = T_{new}$ .

## 4.5 Algorithm for solving the adjoint Stefan problem

We present here the algorithm for solving the adjoint problem (Equation AP). The adjoint variables  $\Theta$  and  $\psi$  are solved backward in time in order to compute the gradient equation (Equation AP.h). These variables are initialized with the forward variables at final time. As mentioned in Chapter 2, the adjoint level set  $\psi$  is no longer a signed distance function but an auxiliary variable that enters in the Dirichlet

boundary condition of the adjoint temperature  $\Theta$  at the interface. The location of the interface at a given time step  $n$  is known thanks to  $\phi^n$ , an information that is stored during the forward resolution of the problem (Algorithm 4). Moreover, the constructed Laplacian at the same time step  $n$ , computed thanks to Algorithm 3, is re-used to solve the adjoint heat equations. The steps required to solve the adjoint problem are described in Algorithm 6.

---

**Algorithm 6:** Algorithm for the solution of the adjoint problem
 

---

**input** :  $T, \phi, \varepsilon_V, t_f$

**output:**  $\Theta^0$

Initialize the adjoint temperature field  $\Theta^f$  (Equation AP.c)

Initialize the adjoint level set function  $\psi^f$  (Equation AP.g)

Set the boundary conditions for  $\Theta$  (Equation AP.d)

**repeat**

1. Checkpoint the faces, volumes and centroids with  $\phi^n$

2. Compute the adjoint Stefan condition  $[\nabla\tilde{\Theta}^n]_2^1 \leftarrow \Theta^n$

3. Solve the adjoint advection equation

$\psi^{n-1} \leftarrow \psi^n, T^n, \phi^n, v_s^n, [\nabla\tilde{\Theta}^n]_2^1, \varepsilon_V$  (Equation AP.f)

4. Solve the adjoint heat equations  $\tilde{\Theta}^{n-1} \leftarrow \tilde{\Theta}^n, \psi^{n-1}, \phi^n$

5. Clean or Initialize dead or fresh cells  $\Theta^{n-1} \leftarrow \tilde{\Theta}^{n-1}$

**until** *initial time*  $0 \leftarrow t_f$ ;

---

Finally, we focus on the solution of the first-order conservation law (Equation AP.f) on  $\Gamma$ . Solving this first-order equation on a time time moving surface can be a numerical challenge. Fortunately, all the derivatives can be extended on all of  $\Omega$ . Leading to the following equation,

$$\frac{\partial\psi}{\partial t} + \text{div}(\psi\vec{v}_S) = \text{rhs} \quad \text{in } \Omega, \quad (4.28)$$

where the right hand side

$$\text{rhs} = \frac{1}{|\nabla\phi|} \left( \frac{\partial T}{\partial n} - \varepsilon_V \frac{\partial v_S}{\partial n} \right) [\nabla\Theta_i]_2^1 \cdot n,$$

is computed beforehand. Numerically, Equation 4.28 is solved using the same Forward-Backward diffusion scheme presented in Section 4.3.1.

With this, we close the Chapter on numerical methods for both the forward and adjoint Stefan problems. In the next Chapter, we present numerical results of the forward problem as well as validations on analytical solutions of the two-phase Stefan problem.



# Numerical simulations of two-phase Stefan problems

---

## Contents

---

<b>5.1</b>	<b>Flower.jl an in-house Julia package . . . . .</b>	<b>67</b>
<b>5.2</b>	<b>Analytical solutions of two-dimensional problems . . . . .</b>	<b>68</b>
5.2.1	Planar interface . . . . .	68
5.2.2	Growing Frank's spheres . . . . .	69
<b>5.3</b>	<b>Crystal growth in an under-cooled liquid bath . . . . .</b>	<b>72</b>
5.3.1	Surface tension effects . . . . .	72
5.3.2	Grid refinement effects . . . . .	76
5.3.3	Effect of the molecular kinetic coefficient . . . . .	77
5.3.4	Anisotropy effects . . . . .	78
<b>5.4</b>	<b>Rayleigh-Bénard convection with a melting boundary . . . . .</b>	<b>81</b>

---

## 5.1 Flower.jl an in-house Julia package

In this Chapter, we validate our numerical scheme with respect to analytical solutions of the two-phase Stefan problem. Moreover, we present numerical results of growing crystal in an under-cooled liquid bath and Rayleigh-Bénard convection with a melting boundary.

The code is written in Julia [Bezanson *et al.* 2017] a high level scientific programming language. The key features of the Julia language are:

- *Just-In-Time compilation*: the code is compiled during execution of a program, at run time.
- *Multiple dispatch*: dispatch process to choose which of a function's methods to call based on the number of arguments given, and on the types of all of the function's arguments.
- *Broadcasting*: applies a given function element-wise on a multi-dimensional array without using extra memory.
- *Shared-memory parallelization*: easy implementation multi-threading in the main loops.

The numerical schemes presented in Chapter 4 are coded in an in-house package *Flower.jl* (see Appendix C for more details). Each time step of the forward or adjoint two-phase Stefan problem requires the solution of  $2 + 1$  linear systems:

1. The two heat equations (one for each phase) using the Cut Cell method coupled to the semi-implicit Crank-Nicolson scheme (Equation 3.15).
2. The level set advection equation using the Forward-Backward diffusion scheme, either in the forward problem (Equation 4.14) or in the adjoint case (Equation 4.28).

The linear systems are solved using the Conjugate Gradient iterative method implemented in the *Iterative.jl* package. In the case of fluid flow, the viscous term in the Navier Stokes equations (see Appendix B) is solved using PETSc (Portable, Extensible Toolkit for Scientific Computation) [Balay *et al.* 1997]. This C library is wrapped in the Julia language in *PETSc.jl*.

The optimization cases presented in Chapter 6 are carried out using the L-BFGS method (adjoint-based) or the Particule Swarm algorithm (derivative-free) implemented in the *Optim.jl* package [Mogensen & Riseth 2018].

## 5.2 Analytical solutions of two-dimensional problems

### 5.2.1 Planar interface

We validate our numerical methods for Stefan problems against analytical solutions. As a first benchmark, we consider a moving planar interface bounded by two walls. This case is taken from [Limare *et al.* 2023], where they benchmark their hybrid embedded-level set boundary method, and from [Crank 1987] (original test case). An initial ice layer is melting from the top where we impose a positive temperature  $T_{\text{wall}}^1$ . The melting temperature is set to  $T_M = 0$  and we only solve for the liquid phase. The Stefan number is set to

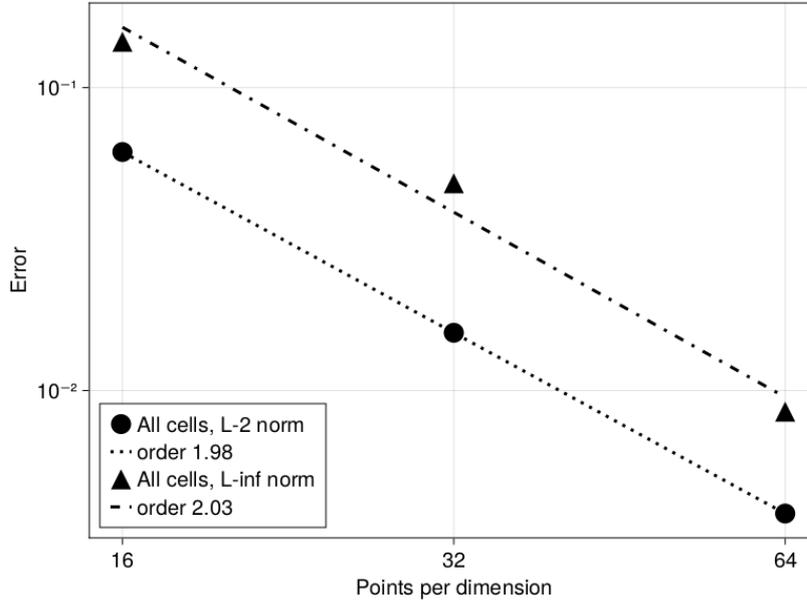
$$\text{St} = \frac{c_L(T_{\text{wall}}^1 - T_M)}{L_H} = 2.85,$$

where  $c_L$  is the liquid heat capacity and  $L_H$  the latent heat. The initial temperature in the liquid phase, only depending in the  $y$  coordinate, is set to

$$T_L(y, t_0) = \frac{\text{erf}\left(\frac{1-y}{2\sqrt{t_0}}\right)}{\text{erf}(l)}, \quad (5.1)$$

where  $t_0$  is the initial time at which the simulation is started and  $l = 0.9$ . The analytical position of the interface is given by the following formula

$$y = 1 - 2l\sqrt{t}. \quad (5.2)$$



**Figure 5.1:** Convergence of the temperature field in the planar interface case.  $L_2$  and  $L_\infty$  norms of the error in temperature in all cells as a function of the number of points per dimension.

The simulation is started at  $t_0 = 0.03$  such that there are enough cells to compute the Stefan condition (Equation 4.1). The simulation is run until a final time  $t_f = 0.1$  for different number of points per dimension  $N = 16, 32$  and  $128$ . The ratio  $\tau/\Delta^2 = 0.5$  is kept constant as we increase the spatial resolution. In Figure 5.1, we show the  $L_2$  and  $L_\infty$  error norm of the temperature field with respect to the analytical solution (Equation 5.1). The method shows a second order convergence in both norms. This case validates the initialization of fresh cells since, as the ice layer melts, it unveils previously non-existing cells in the liquid phase which need to be properly initialized (fresh-cell problem).

### 5.2.2 Growing Frank's spheres

The growth of an ice sphere surrounded by an under-cooled liquid was initially studied by Frank [Frank 1950] where he showed that the initial sphere radius evolved in a self-similar manner as the square-root of time. This problem can be used to validate the accuracy of the numerical scheme by comparing the results with an analytical solution. In this configuration, the temperature field is given by

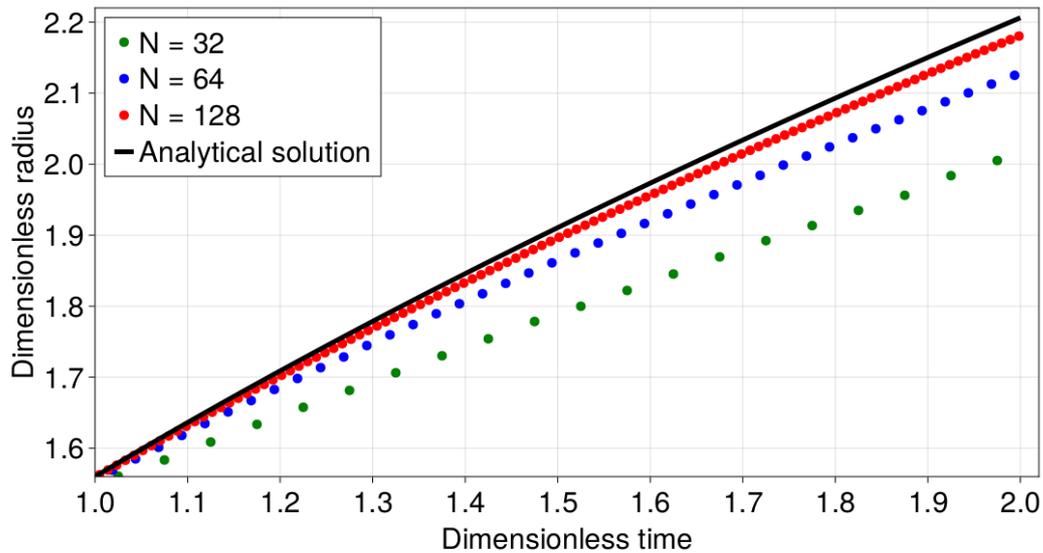
$$T(r, t) = T(s) = \begin{cases} T_\infty \left(1 - \frac{F(s)}{F(S)}\right), & s > S, \\ 0, & s < S, \end{cases} \quad (5.3)$$

where  $r = \sqrt{x^2 + y^2}$  and  $s = r/t^{1/2}$ .  $T_\infty$  is a given under-cooling temperature, and  $F(s)$  denotes the similarity solution of the heat equation

$$F(s) = E_1(1/4 s^2).$$

As the sphere expands, numerical errors may lead to an unwanted alteration of the initial shape due to the Mullins-Sekerka instability [Mullins & Sekerka 1964], where a perturbed solution can lead to unstable dendritic growth in the case of zero melting temperature ( $T_M = 0$ ). We therefore test the robustness of our method using the initial parameters recommended in Almgren [Almgren 1993].

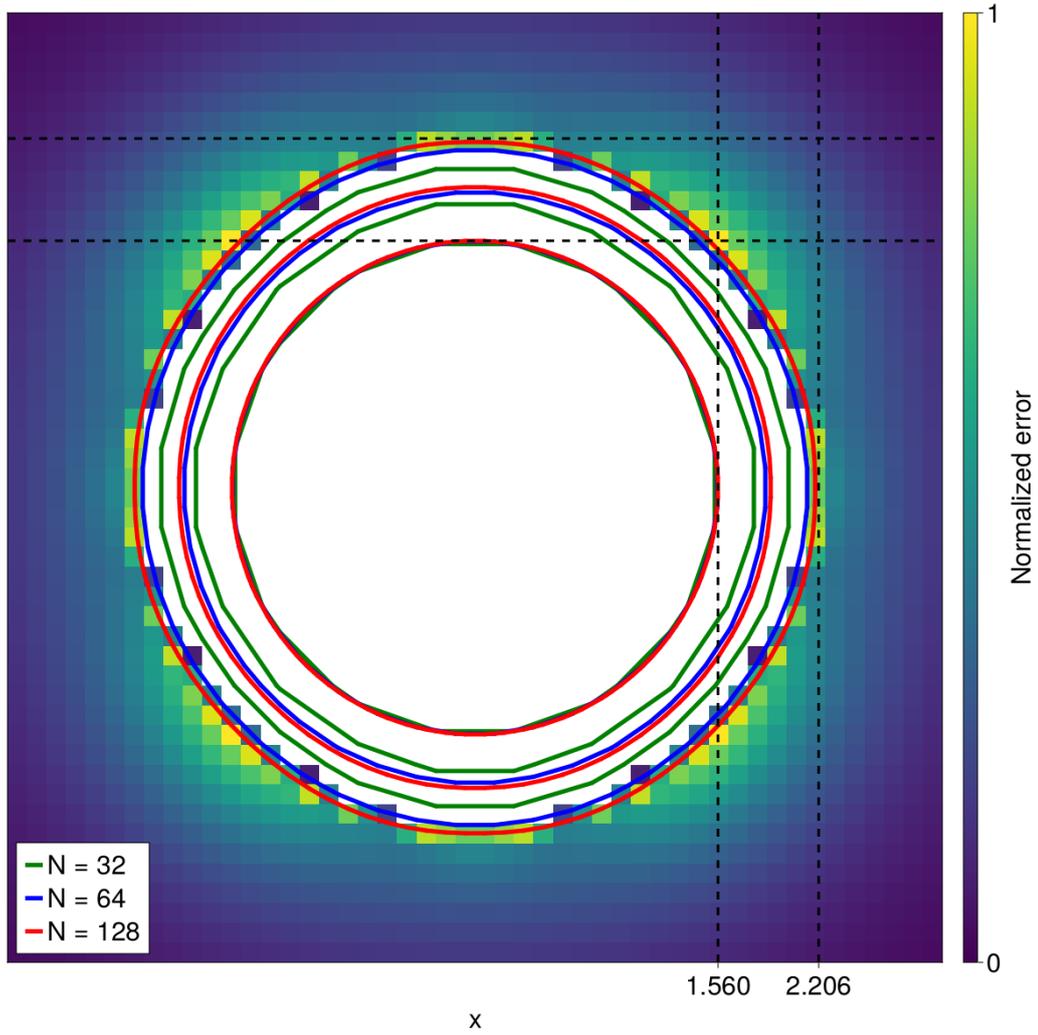
The initial level set function is a circle of radius  $R_0 = 1.56$  in a  $8 \times 8$  domain surrounded by an initial negative temperature field (Equation 5.3 with  $T_\infty = -0.5$ ). The initial time is set to 1 and the simulation is advanced until a final time  $t_f = 2$ . We carry out the simulations with different grids  $N = 32, 64$  and  $128$ . The results of this study are summarized in the following figures.



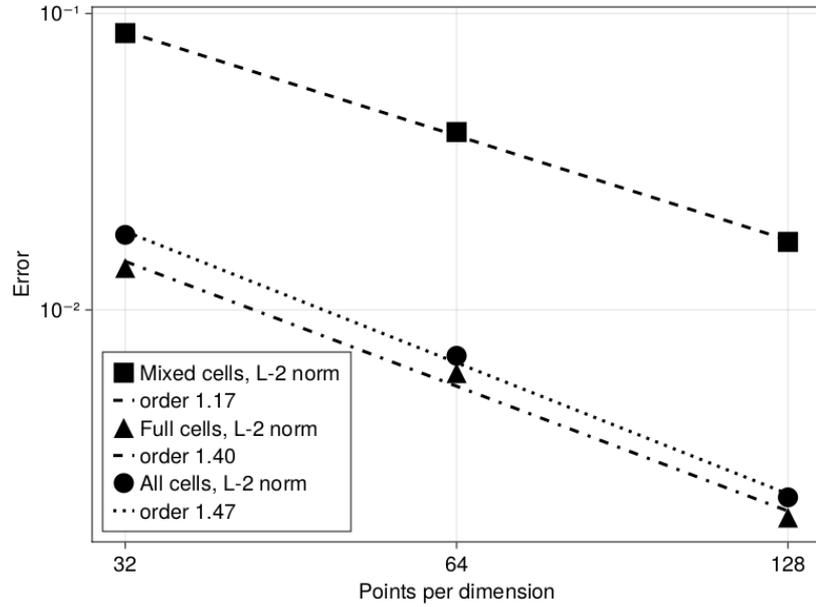
**Figure 5.2:** Convergence of the radius towards the analytical solution as a function of time for the growing Frank sphere.

Figure 5.2 shows the convergence of the radius towards the analytical solution. Figure 5.3 shows the final error map in temperature field as well as the interface positions at different times  $t = 1, 1.5, 2$ . We can clearly see that the initial sphere tends towards the analytical final shape as we increase the number of points. The intrinsic regularization of our method allows the level set function to retain its initial circular shape, avoiding spurious interface oscillations. In Figure 5.4, we present the error norm in temperature field for the different type of cells. The order of convergence is close to 1.5 for cells that are not mixed. The following test cases will

focus on the dendritic growth of an initial crystal shape in a uniform under-cooled temperature field.



**Figure 5.3:** Positions of the interface at times  $t = 1, 1.5, 2$  for the different grids. The dotted lines correspond to the initial radius  $R(t = 1) = 1.56$  and the analytical radius at final time  $R(t = 2) = 2.206$ . The heat map corresponds to the normalized error in temperature field for the highest resolution  $N = 128$ .



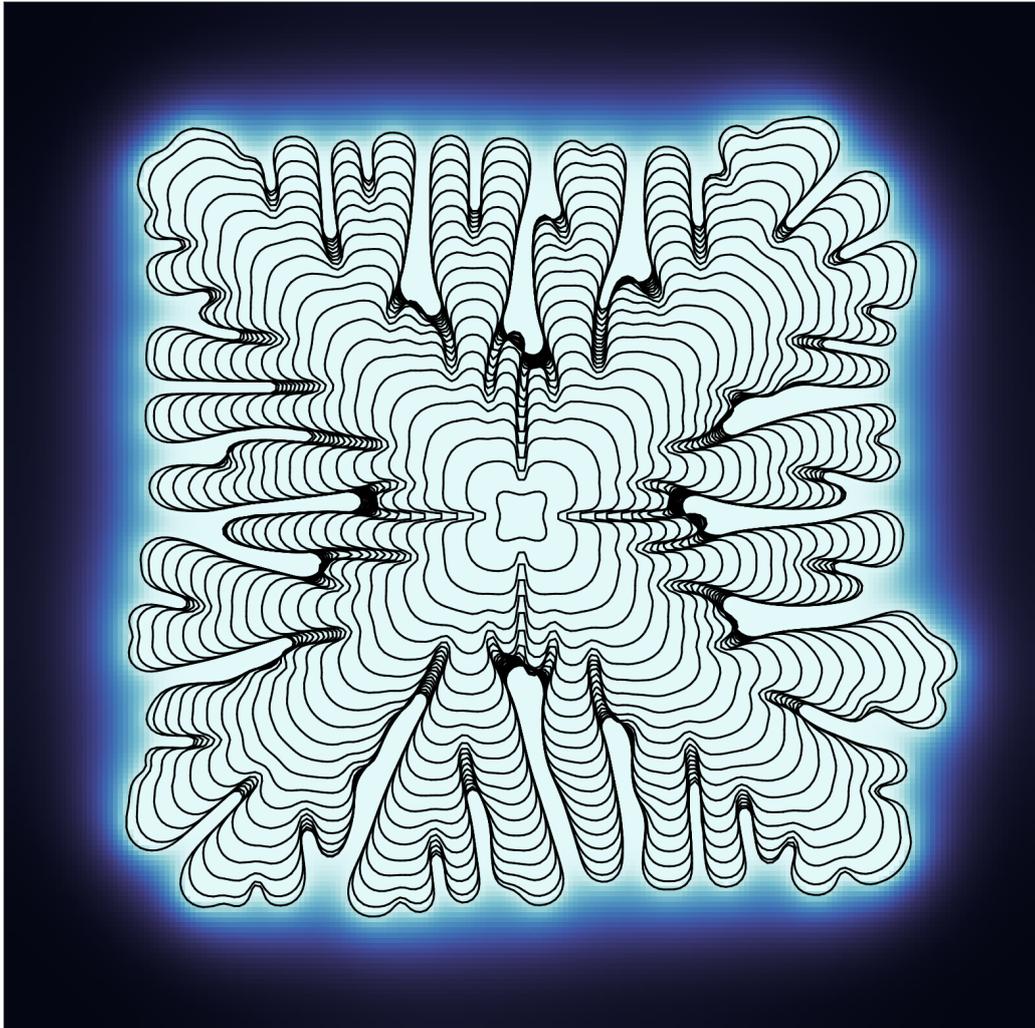
**Figure 5.4:** Convergence of the temperature field in the growing Frank sphere case.  $L_2$  norm of the error in temperature for mixed, full and all cells as a function of the number of points per dimension.

## 5.3 Crystal growth in an under-cooled liquid bath

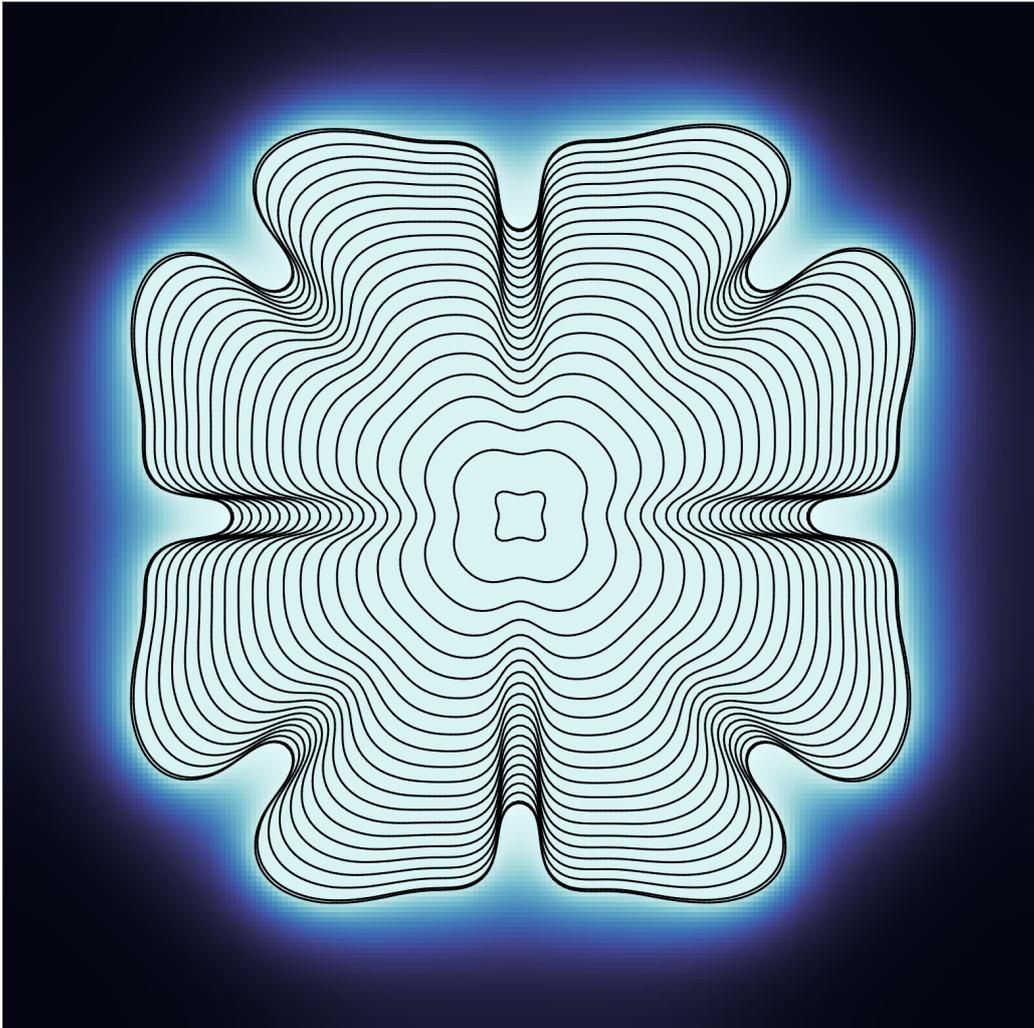
### 5.3.1 Surface tension effects

Crystal growth is an unstable phenomenon that occurs spontaneously in nature. Its appearance is the result of a competition between the natural growth due to the Stefan condition (Equation 1.2), which tends to stretch the tips, and the surface tension effect present in the Gibbs-Thomson relation (Equation 1.3), which tends to restore the flatness of the interface (see [Langer 1980] for further details).

Numerical reproduction of such patterns is a challenge, since the dendritic growth depends on the smallest resolved length scale. Moreover, in the absence of surface tension, the numerical noise and the Mullins-Sekerka instability will lead to an unphysical growth of the initial crystal. An example of crystal growth without surface tension effects is shown in Figure 5.5. The initial 4-folded crystal expands in an unordered fashion with numerically produced dendrites. On the other hand, when taking into account the surface tension effects, the solution is regularized (Figure 5.6).



**Figure 5.5:** Example of numerical instabilities in crystal growth due to the absence of surface tension effects. The initial condition is a solid crystal at temperature  $T = 0$  surrounded by an under-cooled temperature field with  $T_\infty = -0.5$ . The color map represents the final temperature field.

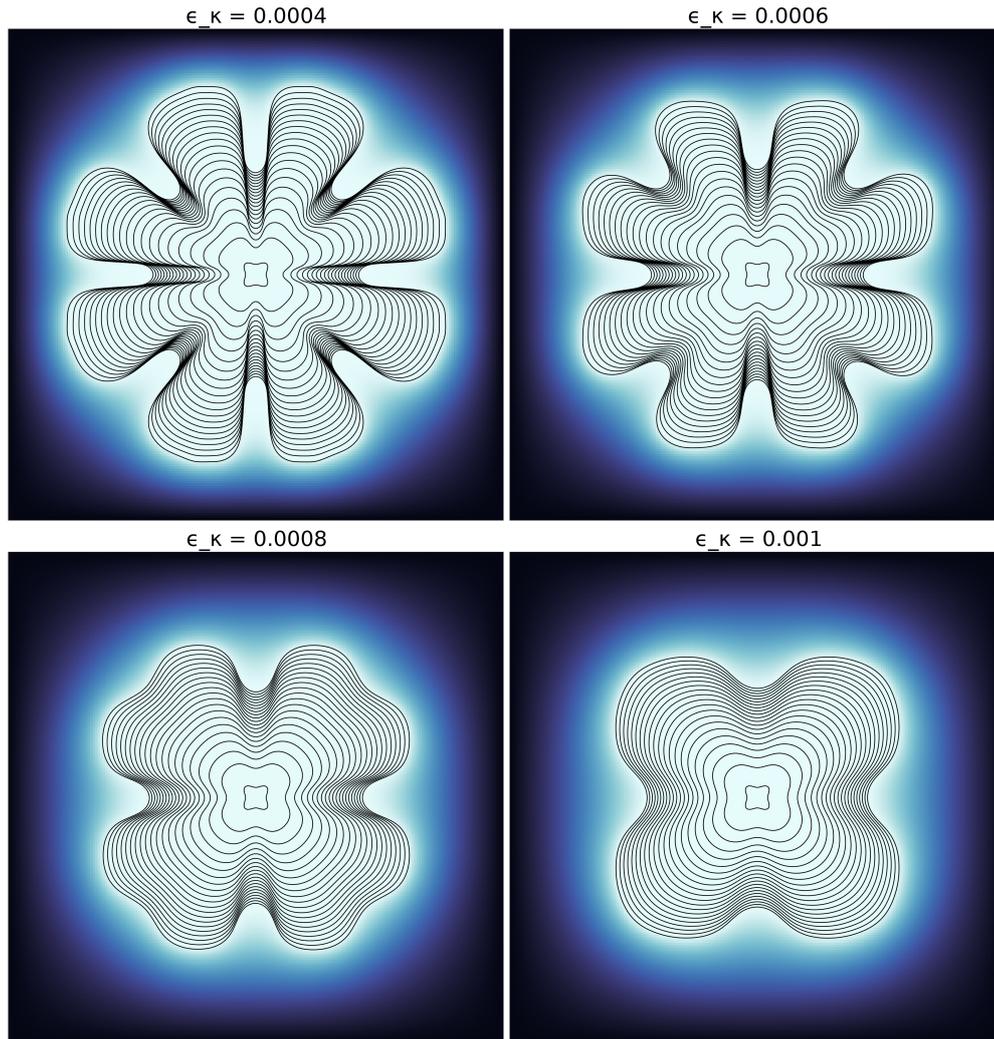


**Figure 5.6:** Example of the regularization introduced by the surface tension effects in crystal growth. The initial solid crystal at a temperature  $T = 0$  is surrounded by an undercooled temperature field with  $T_\infty = -0.5$ . The color map represents the final temperature field.

In order to validate our method and to qualitatively describe the effect of the Gibbs-Thomson relation, a case similar to that of Chen [Chen *et al.* 1997] is considered, where an initial solid crystal is surrounded by an under-cooled liquid. The level set function is initialized in a  $2 \times 2$  domain as

$$\phi_0(x, y) = \sqrt{x^2 + y^2} \times (0.1 + 0.02 \cos(4\alpha) - 0.01), \quad (5.4)$$

where  $\alpha$  is the angle of the interface with respect to the  $x$  axis, and  $T_\infty = -0.5$ .



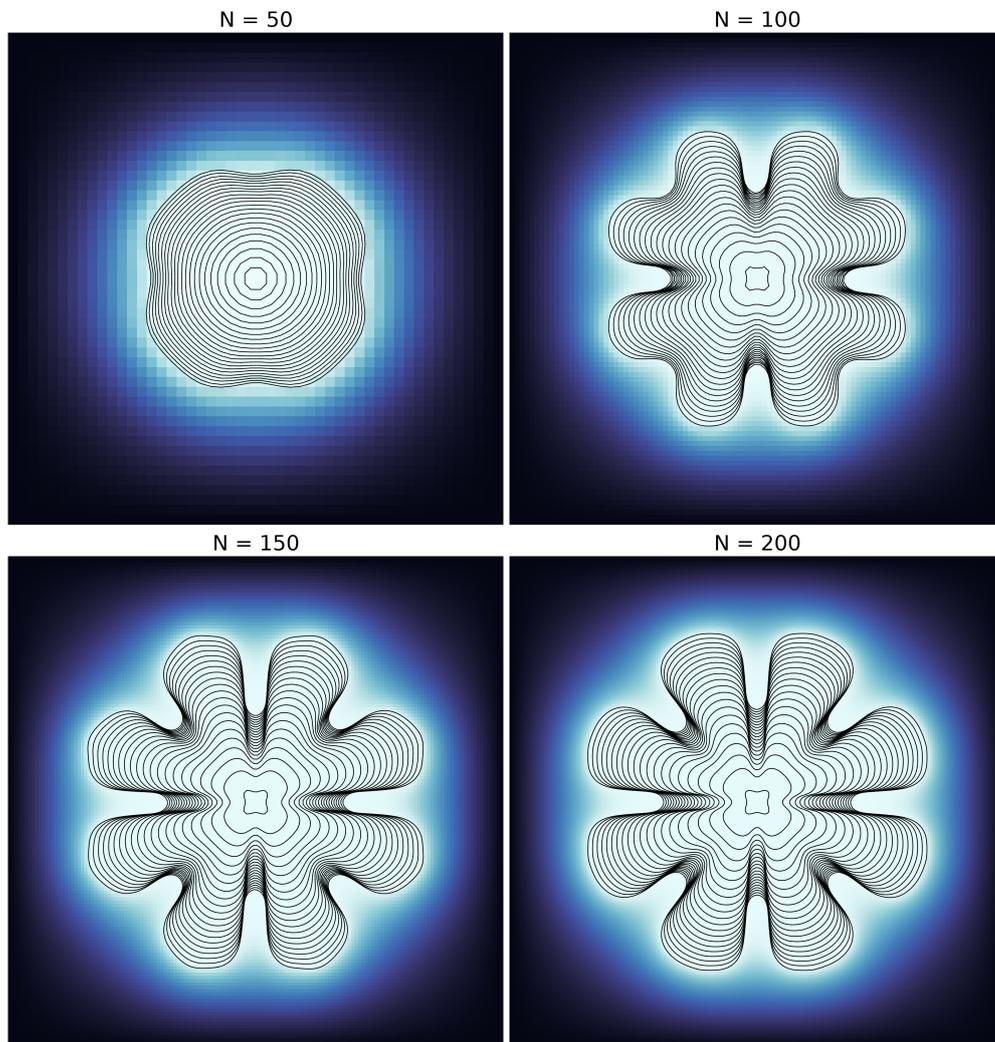
**Figure 5.7:** Effect of the surface tension coefficient on the dendrite growing tips for the indicated coefficients. The initial solid crystal at a temperature  $T = 0$  is surrounded by an under-cooled temperature field with  $T_\infty = -0.5$ . The color map represents the final temperature field.

The surface tension coefficient is varied from 0.0004 to 0.001 for a fixed  $N = 200$  with the kinetic coefficient set to  $\varepsilon_V = 0$ . Figure 5.7 clearly demonstrates the stabilizing effect of the Gibbs-Thomson relation. The tip-splitting disappears as the

surface tension increases. This behavior is explained by a stronger regularization due to higher surface tension coefficient, reducing the growth rate of the instability, and is in agreement with the observations in similar regimes [Chen *et al.* 1997, Juric & Tryggvason 1996].

### 5.3.2 Grid refinement effects

The simulation presented in the previous section were carried out with a relatively high number of points per dimension ( $N = 200$ ). As a second validation, the effect of the grid resolution on the solution (Figure 5.8) is assessed.

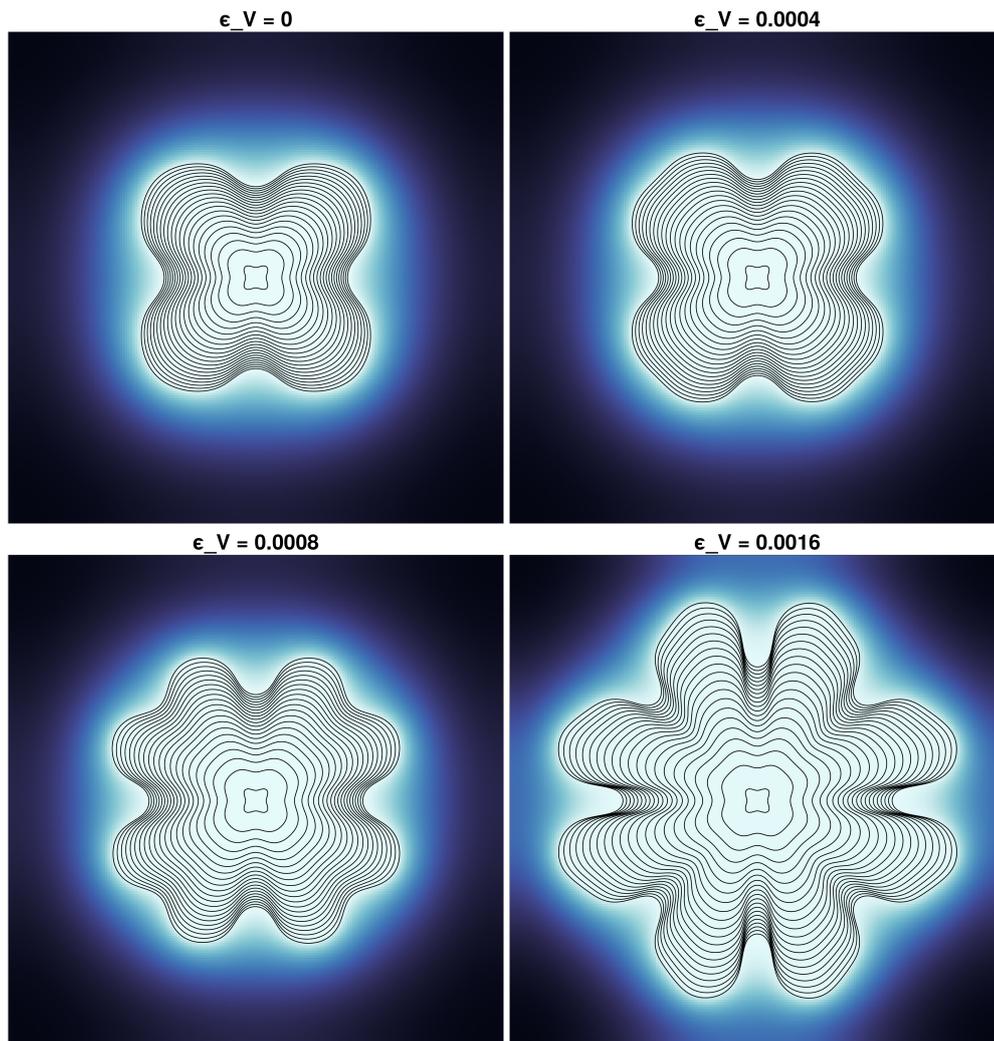


**Figure 5.8:** Effect of grid refinement on the dendrite growing tips. The initial solid crystal at a temperature  $T = 0$  is surrounded by an under-cooled temperature field with  $T_\infty = -0.5$ . The color map represents the final temperature field. The final time is  $t_f = 0.5$  and the interface is plotted with a time step of 0.025.

The surface tension coefficient is fixed at  $\varepsilon_\kappa = 0.0004$ , and kinetic coefficient is set

to  $\varepsilon_V = 0$ . The simulations are advanced with insulated boundary conditions until a final time  $t_f = 0.5$  for different grid resolutions  $N = 50, 100, 150, 200$ . Figure 5.8 shows that as the grid size decreases, the tip-splitting appears earlier and the length of the dendrites increases. The final shape converges towards an 8-fold symmetric crystal shape.

### 5.3.3 Effect of the molecular kinetic coefficient



**Figure 5.9:** Effect of the molecular kinematic coefficient on the dendrite growing tips. The initial solid crystal at a temperature  $T = 0$  is surrounded by an under-cooled temperature field with  $T_\infty = -0.5$ . The color map represents the final temperature field. The final time is  $t_f = 0.4$  and the interface is plotted with a time step of 0.02.

In this test case, we assess the effect of the molecular kinetic coefficient. We take the same initial condition than in Section 5.3.1, we fix the surface tension  $\varepsilon_\kappa = 0.001$  and we run the simulation until a final time  $t_f = 0.4$ . Figure 5.9 shows the effect of the

kinematic coefficient as we vary  $\varepsilon_V$  from 0 to 0.0016. As the coefficient is increased, the tip of the dendrites move faster from the initial shape. At the tip, where the initial velocity is maximal, due to the Stefan condition, the melting temperature is reduced by an  $\varepsilon_V v_S$  factor thus further entraining the growing motion.

### 5.3.4 Anisotropy effects

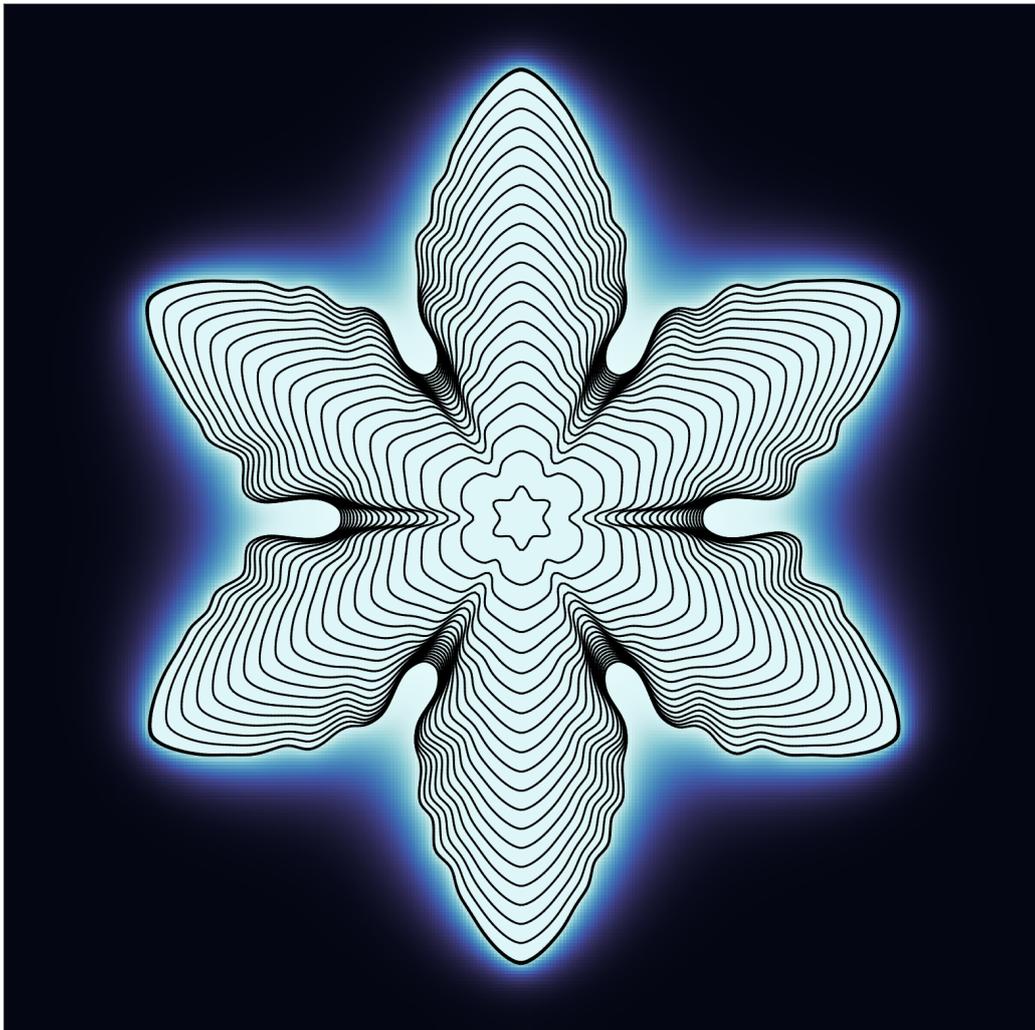
In order to accurately reproduce the crystal shapes produced by dendritic growth, we introduce a variable surface tension coefficient, similar to [Tan & Zabaras 2006] and [Limare *et al.* 2023], in the Gibbs-Thomson relation (Equation 1.3) to account for anisotropy effects

$$\bar{\varepsilon}_\kappa(\alpha) = \varepsilon_\kappa \left( 1 + A \left[ \frac{8}{3} \sin^4 \left( \frac{1}{2} M (\alpha - \alpha_0) \right) - 1 \right] \right), \quad (5.5)$$

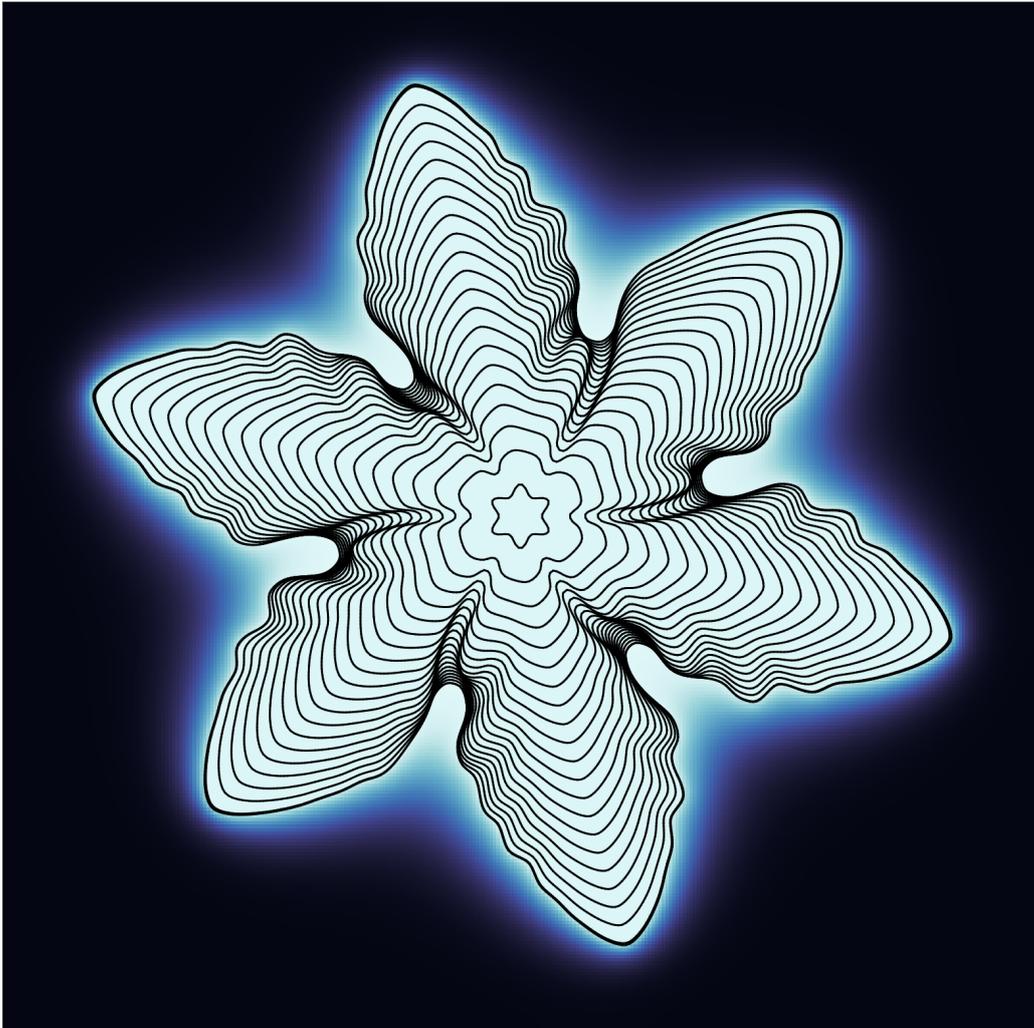
where  $\varepsilon_\kappa$  is the surface tension coefficient,  $A$  represents the weight of the anisotropy effect,  $M$  the mode number,  $\alpha$  the angle of the interface with respect to the x-axis and  $\alpha_0$  the prescribed angle of symmetry on which the dendrites will grow.

To validate the anisotropy effects, we initialize a six-fold crystal in an undercooled liquid with  $T_\infty = -0.8$  and a fixed velocity coefficient  $\varepsilon_V = 0$ . The anisotropic weight is fixed at  $A = 0.4$  and mode number  $M = 6$ . The simulations are carried out for two different prescribed angles  $\alpha_0 = \pi/2, \pi/4$  until a final time  $t_f = 0.09$  in a  $2 \times 2$  domain with  $N = 300$ .

Figure 5.10 shows the crystal growing in the direction of the prescribed angle  $\alpha_0 = \pi/2$ , when the angle of symmetry is aligned with the initial branches. The minimal surface tension effect along the symmetry lines allows the the primary branches to grow further away from the initial crystal. We can also observe secondary branches appearing and propagating. When  $\alpha_0 = \pi/4$  (different from the initial orientation of the branches) the crystal rotates towards the prescribed direction (Figure 5.11). These results validate our implementation of the anisotropy effects.



**Figure 5.10:** Anisotropy effects on the crystal growth for a prescribed angle of symmetry  $\alpha_0 = \pi/2$ . The initial condition is a six-fold solid at temperature  $T = 0$  surrounded by an under-cooled temperature field with  $T_\infty = -0.8$ . The color map represent the final temperature field. The final time is  $t_f = 0.09$  and the interface is plotted with a time step of 0.0045.



**Figure 5.11:** Anisotropy effects on the crystal growth for a prescribed angle of symmetry  $\alpha_0 = \pi/4$ . The initial condition is a six-fold solid at temperature  $T = 0$  surrounded by an under-cooled temperature field with  $T_\infty = -0.8$ . The color map represent the final temperature field. The final time is  $t_f = 0.09$  and the interface is plotted with a time step of 0.0045.

## 5.4 Rayleigh-Bénard convection with a melting boundary

In this section, we investigate the formation of convection cells when considering a fluid flow in the two-phase Stefan problem. The global Rayleigh number (Equation 1.13) will control the onset of the Rayleigh-Bénard instabilities. We use the setup described in Section 1.3, where a initial fluid layer of height  $h_0$  is bounded by two walls in a  $[0 : bH] \times [0 : H]$  domain with  $b$  the aspect ratio. We consider an isothermal solid phase such that the temperature in the solid phase is set to the melting temperature  $T_M = 0$ .

We define an effective Rayleigh number similarly to Favier [Favier *et al.* 2019] and Limare [Limare *et al.* 2023] as

$$\text{Ra}_e = \text{Ra}(1 - T_M)\bar{h}^3, \quad (5.6)$$

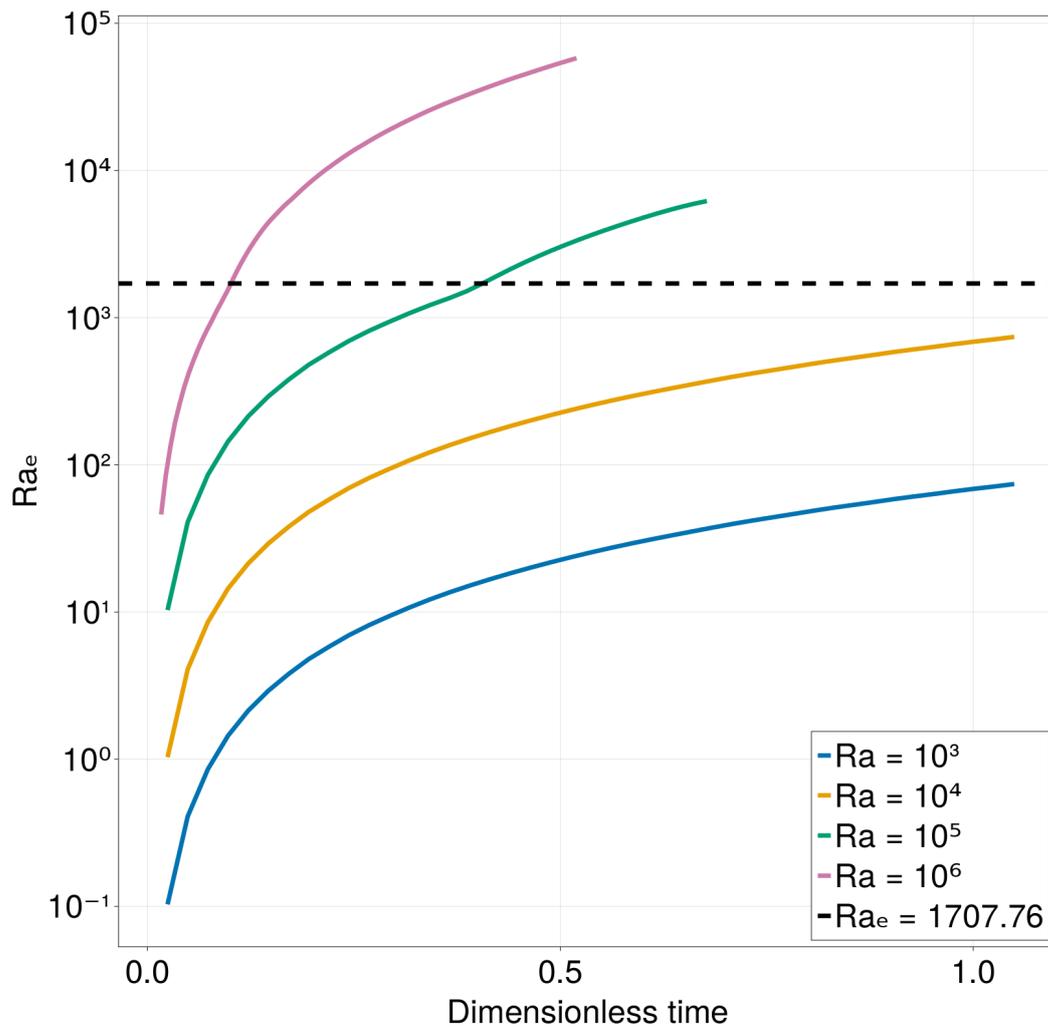
where  $\bar{h}$  is the average fluid height defined as

$$\bar{h}(t) = \frac{1}{b} \int_0^b h(x, t) dx. \quad (5.7)$$

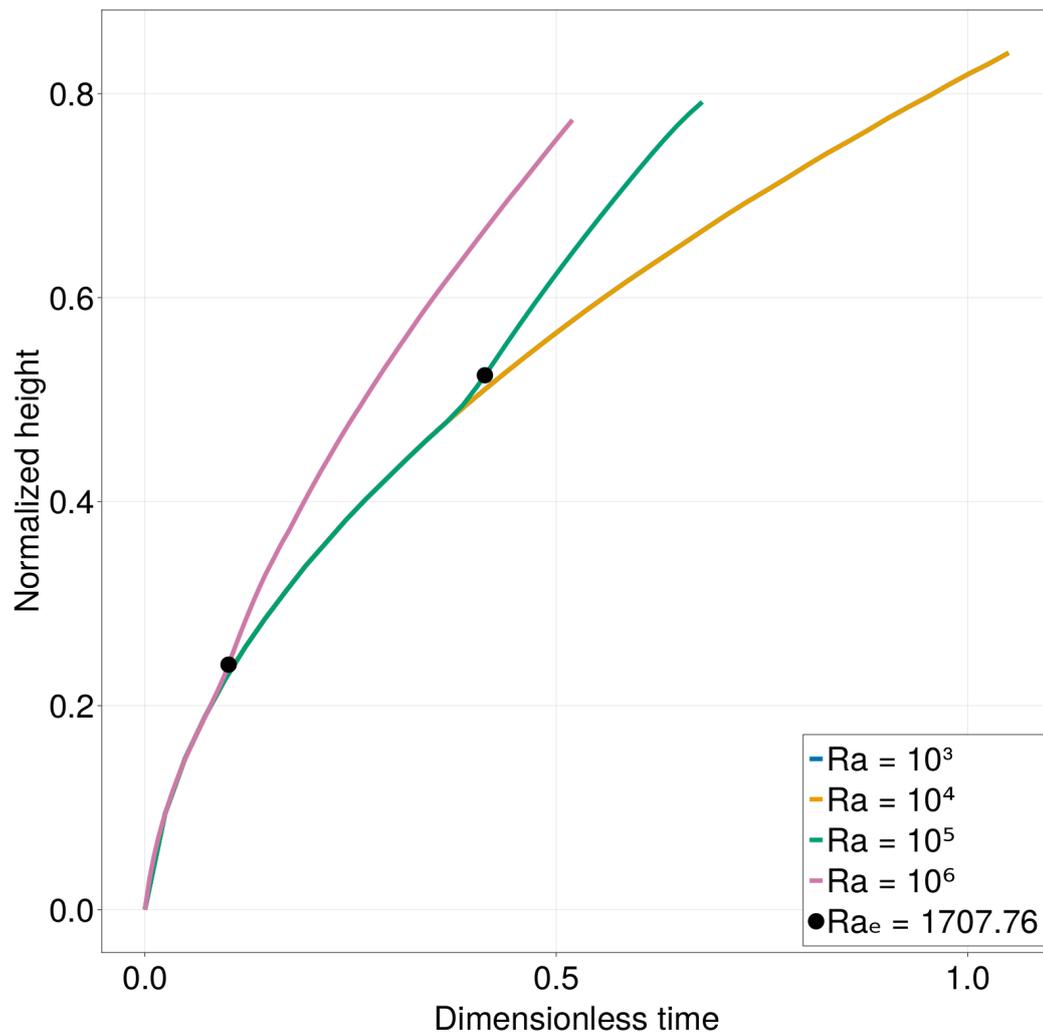
When the effective Rayleigh number reaches the well-known critical value  $\text{Ra}_c = 1707.76$ , the initial diffusion-driven motion is transformed into a convection-driven one. We carry out simulations for different  $\text{Ra} = 10^3, 10^4, 10^5$  and  $10^6$  with a Stefan number set to  $\text{St} = 0.25$ . The aspect ratio is set to  $b = 8$  and the grid used in our simulation is  $512 \times 64$ . The simulation are run until the maximum height reaches  $0.9H$ .

Figure 5.12 shows the effective Rayleigh number as a function of time for the different cases. When  $\text{Ra} = 10^3$  and  $\text{Ra} = 10^4$  the critical Rayleigh number is not reached and the motion of the fluid layer is not affected, thus remaining a diffusion-driven one, similarly to the planar motion described in Section 5.2.1. In the  $\text{Ra} = 10^5$  and  $\text{Ra} = 10^6$  cases, however, we see  $\text{Ra}_e$  crossing the threshold indicating the onset of the instability. This onset is visible in Figure 5.13, where the average height strongly increases when the convection-driven motion takes over.

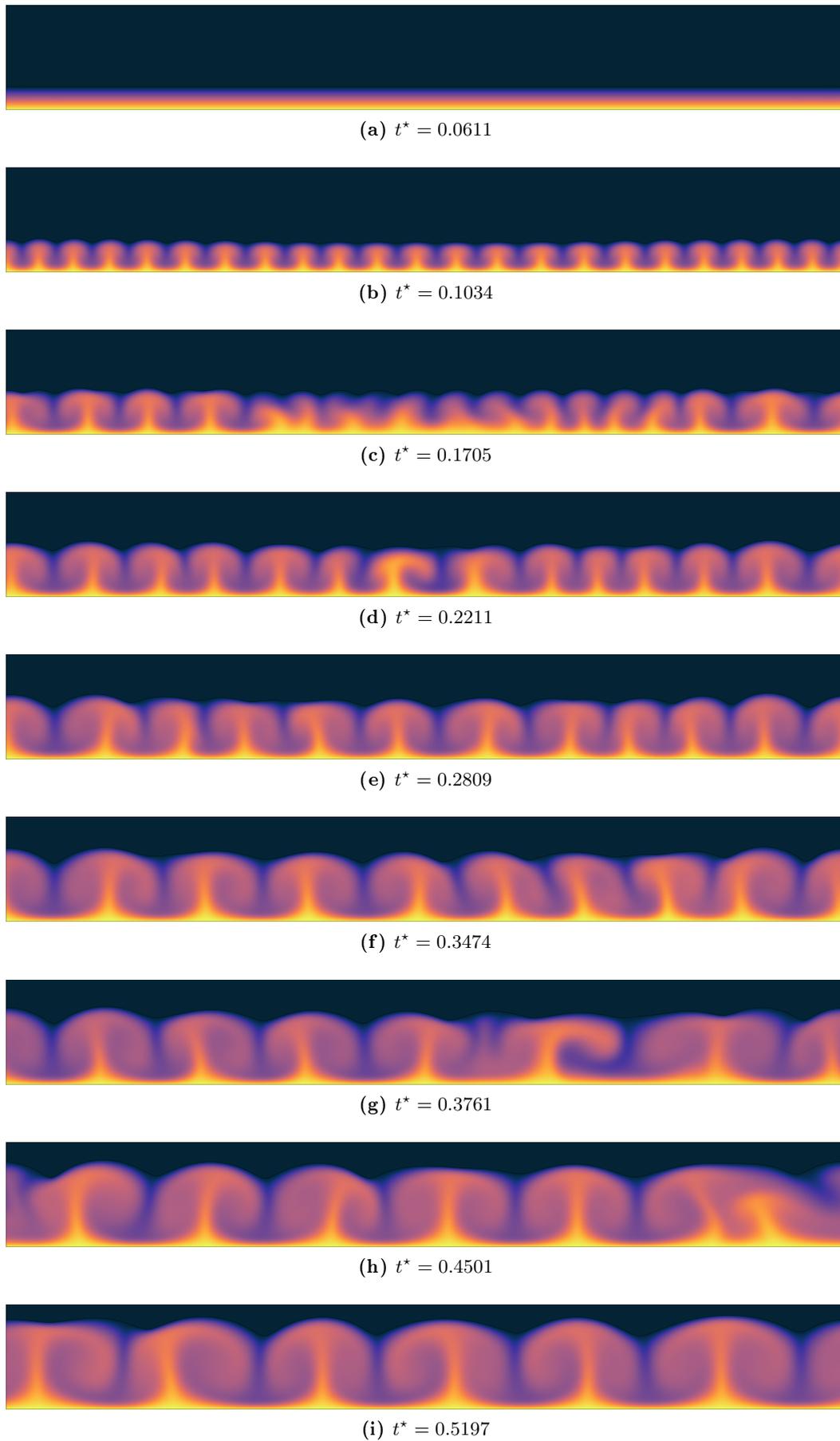
Figure 5.14 shows a time series of the temperature field and interface position for the  $\text{Ra} = 10^6$  case. In that case, the critical Rayleigh number is reached at  $t^* = 0.1034$  and the first bifurcation appears, creating the convection cells. The size of the convection cells will then vary with the secondary bifurcations mechanism. When the averaged height  $\bar{h}$  matches the characteristic wavelength of the convection rolls, the convection cells have sufficient time to merge and then stabilize. We also note that the interface is deformed according to the shape of the cells.



**Figure 5.12:** Effective Rayleigh number in logarithmic scale as a function of time for the indicated global Rayleigh numbers. The dotted black line indicates the critical Rayleigh number.



**Figure 5.13:** Average height as a function of time for the indicated global Rayleigh numbers. The black dots indicate the moment the  $Ra_c$  is reached for the  $Ra = 10^5$  and  $Ra = 10^6$  cases.



**Figure 5.14:** Times series of the temperature field and interface position in the  $Ra = 10^6$  case. The Rayleigh-Bénard convection cells appear at  $t^* = 0.1034$ .

# Shape optimization results

## Contents

<b>6.1</b>	<b>Actuation function parametrization . . . . .</b>	<b>85</b>
<b>6.2</b>	<b>Adjoint-based optimization cases . . . . .</b>	<b>86</b>
6.2.1	Asymmetric melting of a solid circle . . . . .	86
6.2.2	Controlling the Mullins-Sekerka instability . . . . .	88
6.2.3	Growing crystals with topology changes . . . . .	90
<b>6.3</b>	<b>Comparison with derivative-free methods . . . . .</b>	<b>93</b>

## 6.1 Actuation function parametrization

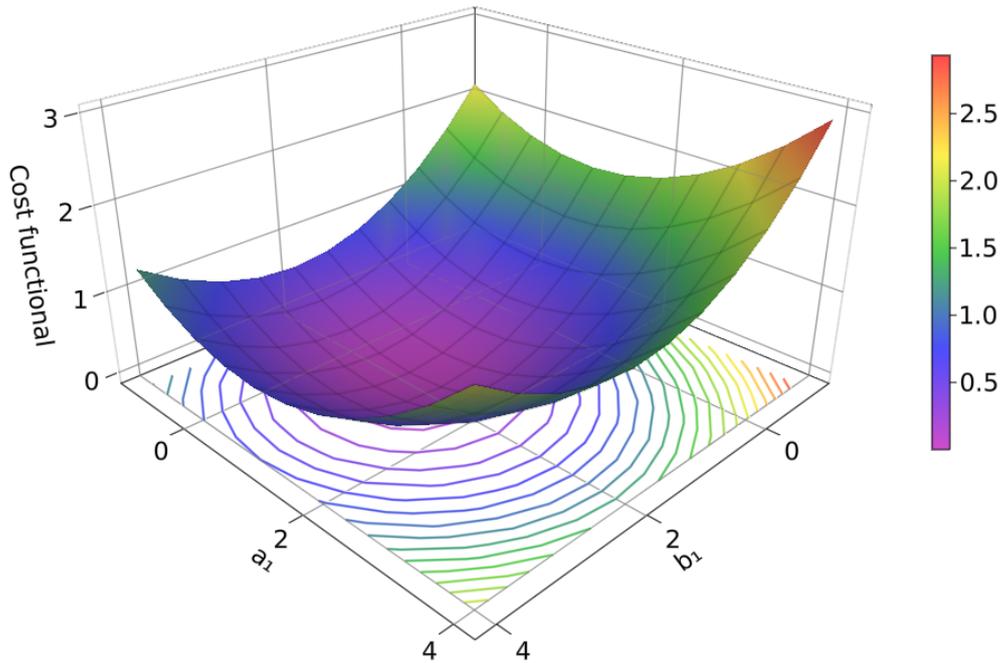
A range of optimization cases for the Stefan problem with varying complexity are presented here. The results and performance of the adjoint-based algorithm (Algorithm 2) are analyzed and compared to a derivative-free algorithm (with no gradient requirement) for these different setups.

In all of the cases the desired temperature field,  $T^f$ , and desired level set,  $\phi^f$ , are computed beforehand, and are used to drive the control variable  $w$  that acts as a Neumann (or Dirichlet) boundary condition on the domain boundary. The actuator  $w$  is parametrized with a Fourier basis

$$w = \sum_{p=1}^n a_p \cos(p\pi x) + \sum_{p=1}^n b_p \sin(p\pi x), \quad (6.1)$$

where  $x$  corresponds to the bounds of the domain and  $a$  and  $b$  to the basis coefficients. The number of coefficients to fit is  $2n$ , with  $n$  being the order of each basis function. Through the optimization process, the amplitude of each basis function is determined using the gradient equation (Equation AP.h).

By opting for a parameterized distribution we ensure the smoothness of the actuation function. Due to the high sensitivity of the cost functional with respect to the basis considered – too many parameters will create multiple local minima – the number of parameters are kept at a low enough value to ensure the convexity of the problem while allowing spatial variation of the actuation function.



**Figure 6.1:** Example of convex cost functional as a function of two parameters  $a_1$  and  $b_1$ .

An example of cost functional  $\mathcal{J}$  as a function of the two coefficients that parametrize the actuator is shown in Figure 6.1. In that simple case, the convexity of the cost functional is clear and the gradient equation will drive  $\mathcal{J}$  towards the global minimum. Moreover, in all of the optimization cases considered, the initial guess – the starting point in our cost functional space – is set to  $w = 0$  in order not to add a bias regarding the initial descent.

## 6.2 Adjoint-based optimization cases

### 6.2.1 Asymmetric melting of a solid circle

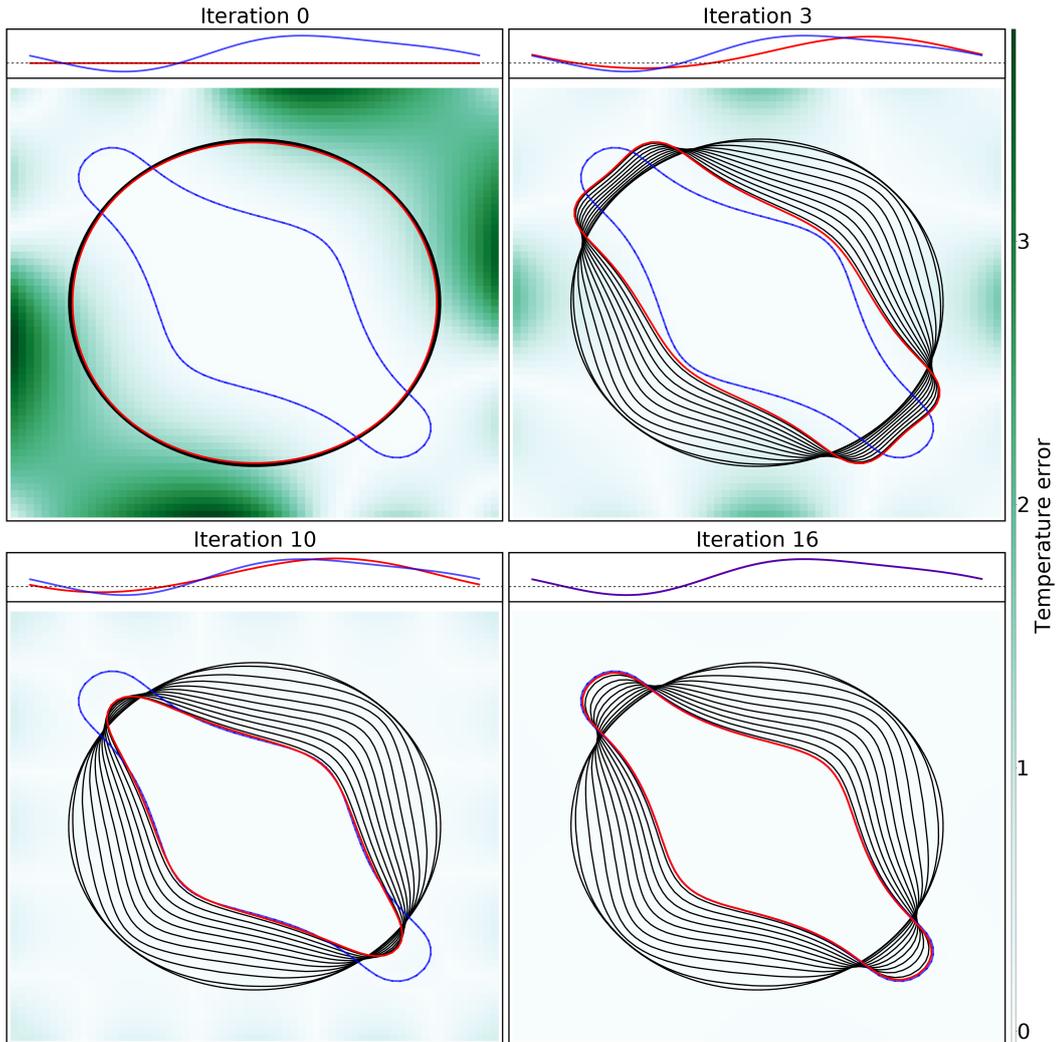
In the first case, we consider an asymmetric expansion and retraction of an initial ice circle of radius  $R = 0.75$  in a  $2 \times 2$  domain. The surface tension coefficient is constant and set to  $\varepsilon_\kappa = 0.002$  while  $\varepsilon_V = 0$ . The non-zero surface tension coefficient is added to further regularize the level set function. The number of points per dimension is  $N = 64$  and the final time is set to  $t_f = 0.1$ .

The control  $w$  acts on the whole domain boundary and is of Dirichlet type. The adjoint boundary condition, on the other hand, remains a homogeneous Neumann boundary condition. The actuator  $w$  is parameterized as

$$u = \sum_{p=1}^2 a_p \cos(p\pi x) + \sum_{p=1}^2 b_p \sin(p\pi x). \quad (6.2)$$

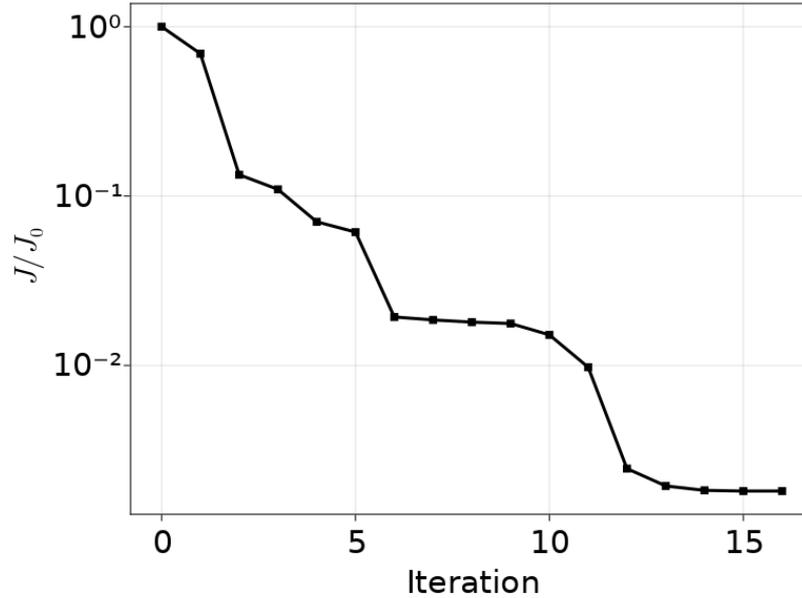
The coefficients in the cost functional (Equation 2.1) are set to  $\beta_1 = 1$ ,  $\beta_2 = 10$  and  $\beta_4 = 10^{-3}$ . In this case, the term controlling the interface length  $\beta_3$  is set to zero.

Figure 6.2 shows the level set of the forward problem evolving towards the desired shape after 16 iterations corresponding to 66 solutions of the forward problem (FP) and 16 solutions of the adjoint problem (AP). The algorithm is able to recover both the retraction and the expansion of the initial circle.



**Figure 6.2:** Iterations 0, 3, 10 and 16 of the optimization procedure for the circle test case. The blue curve represents the desired shape and the red one the final position of the interface at a given iteration. The final time is  $t_f = 0.1$  and the interface is plotted with a time step of 0.01. The color map corresponds to the error in temperature field. The inset shows the actuator  $w$  at a given iteration (red) and the desired one (blue).

Figure 6.3 shows the evolution of the normalized cost functional  $\mathcal{J}/\mathcal{J}_0$  as a function of the iterations. At iteration 6, we see the descent slowing down and almost stopping, indicating the presence of a local minimum. Nevertheless, the algorithm manages to overcome it and continue its descent.



**Figure 6.3:** Normalized cost functional as a function of the iterations of the L-BFGS optimization procedure for the melting circle case.

## 6.2.2 Controlling the Mullins-Sekerka instability

In this second case, we consider a perturbed initial planar shape in a  $2 \times 2$  domain such that the initial level set function is given by,

$$\phi_0(x, y) = y + 0.6 + A \cos(2\pi x), \quad (6.3)$$

and the temperature field by,

$$T_0(x, y) = \begin{cases} -1 + e^{-T_\infty \phi_0(x, y)}, & \phi_0 > 0, \\ 0, & \phi_0 < 0, \end{cases} \quad (6.4)$$

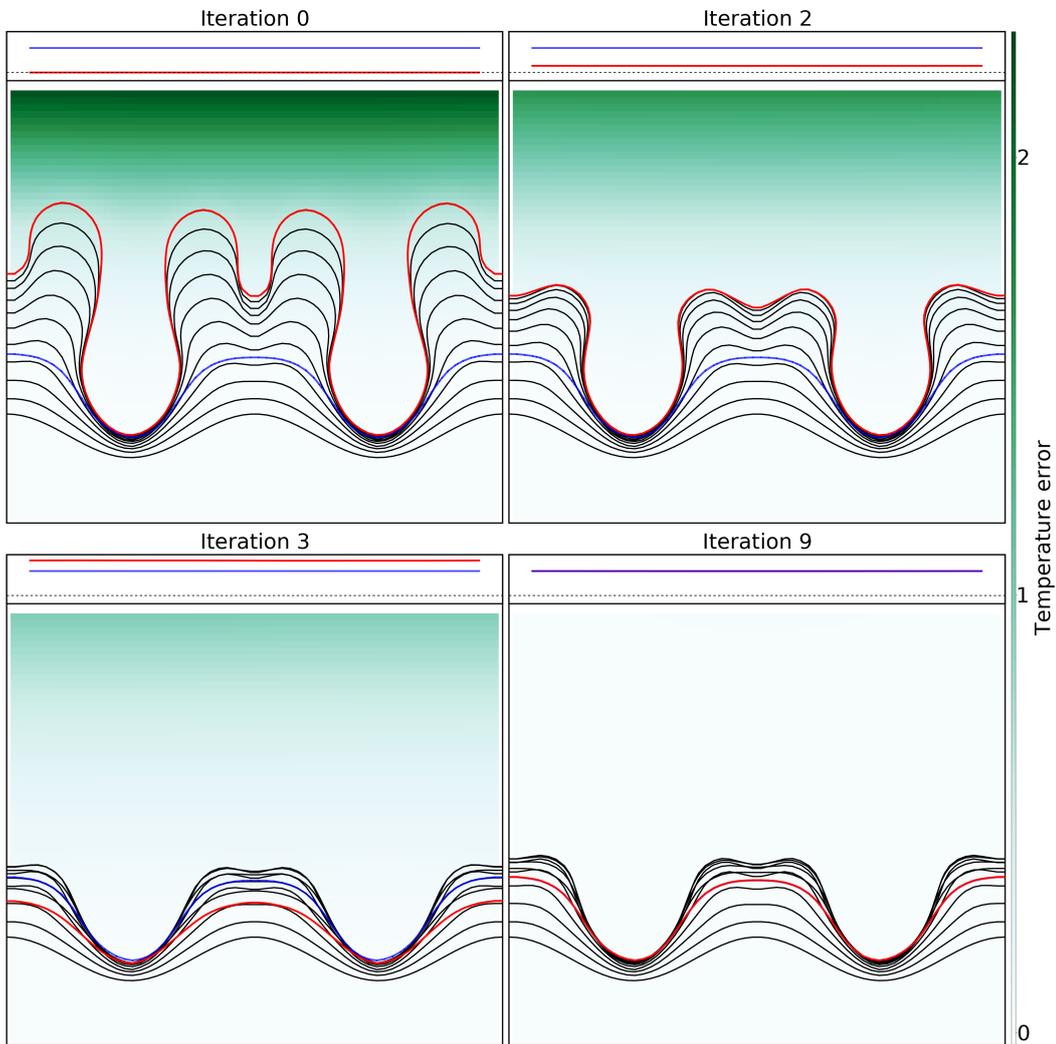
with an amplitude of the perturbation  $A = 0.05$  and an under-cooling temperature  $T_\infty = 1.2$ . The surface tension and velocity coefficients are set to zero in this case.

As described in Chapter 5, an initial perturbation such as that prescribed in this problem leads to a Mullins-Sekerka type instability [Mullins & Sekerka 1964, Chen *et al.* 1997] characterized by unstable dendritic growth. The purpose of this optimization test case is to reduce the instability by imposing an optimal actuation on the top boundary counter-acting the motion of the tips. The control variable  $w$

is of Neumann type and we set the final time  $t_f = 0.5$  with  $N = 64$ . Similar to the previous case the actuation is parametrized as

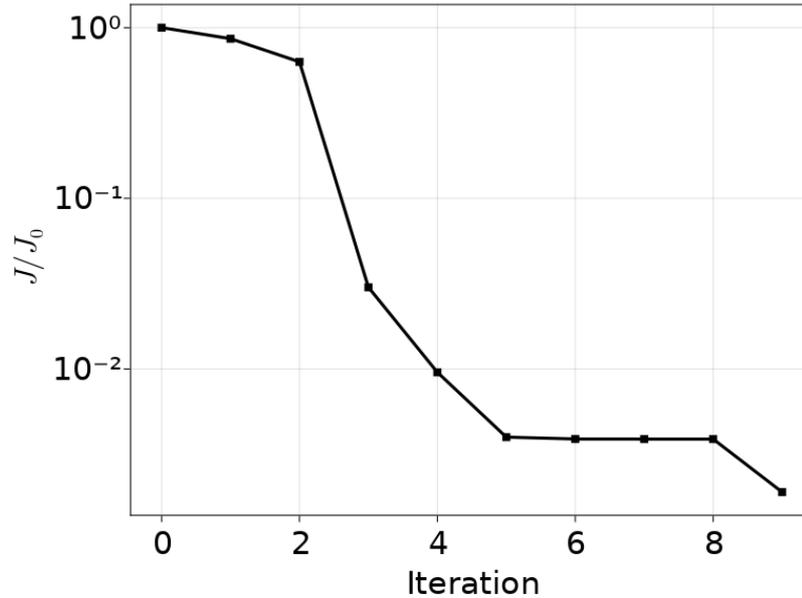
$$w = \sum_{p=1}^4 a_p \cos(p\pi x) + \sum_{p=1}^4 b_p \sin(p\pi x). \quad (6.5)$$

In this case, we can add four extra basis function in the actuator parametrization and still recover the global minimum. The extra term in the cost functional which controls the length of the interface is also included and set to  $\beta_3 = 0.1$  while the other coefficients are  $\beta_1 = 1$ ,  $\beta_2 = 10$  and  $\beta_4 = 10^{-4}$ .



**Figure 6.4:** Iterations 0, 2, 3 and 9 of the optimization procedure for the Mullins-Sekerka test case. The blue curve represents the desired shape and the red one the final position of the interface at a given iteration. The final time is  $t_f = 0.5$  and the interface is plotted with a time step of 0.05. The color map corresponds to the error in temperature field. The inset shows the actuator  $w$  at a given iteration (red) and the desired one (blue).

In Figure 6.4, we can observe how the initial tip splitting is reduced as we go through the optimization procedure. The minimum is attained after 10 iterations of the optimization algorithm corresponding to 54 solutions of the forward problem (FP) and 10 solutions of the adjoint problem (AP).



**Figure 6.5:** Normalized cost functional as a function of the iterations of the L-BFGS optimization procedure for the Mullins-Sekerka case.

The variation of the cost functional (Figure 6.5) is similar to the previous one, with an initial step descent after a few iterations that eventually plateaus. In this case, the optimization algorithm is stopped by the criterion on the gradient  $|\nabla \mathcal{J}| < 10^{-6}$ .

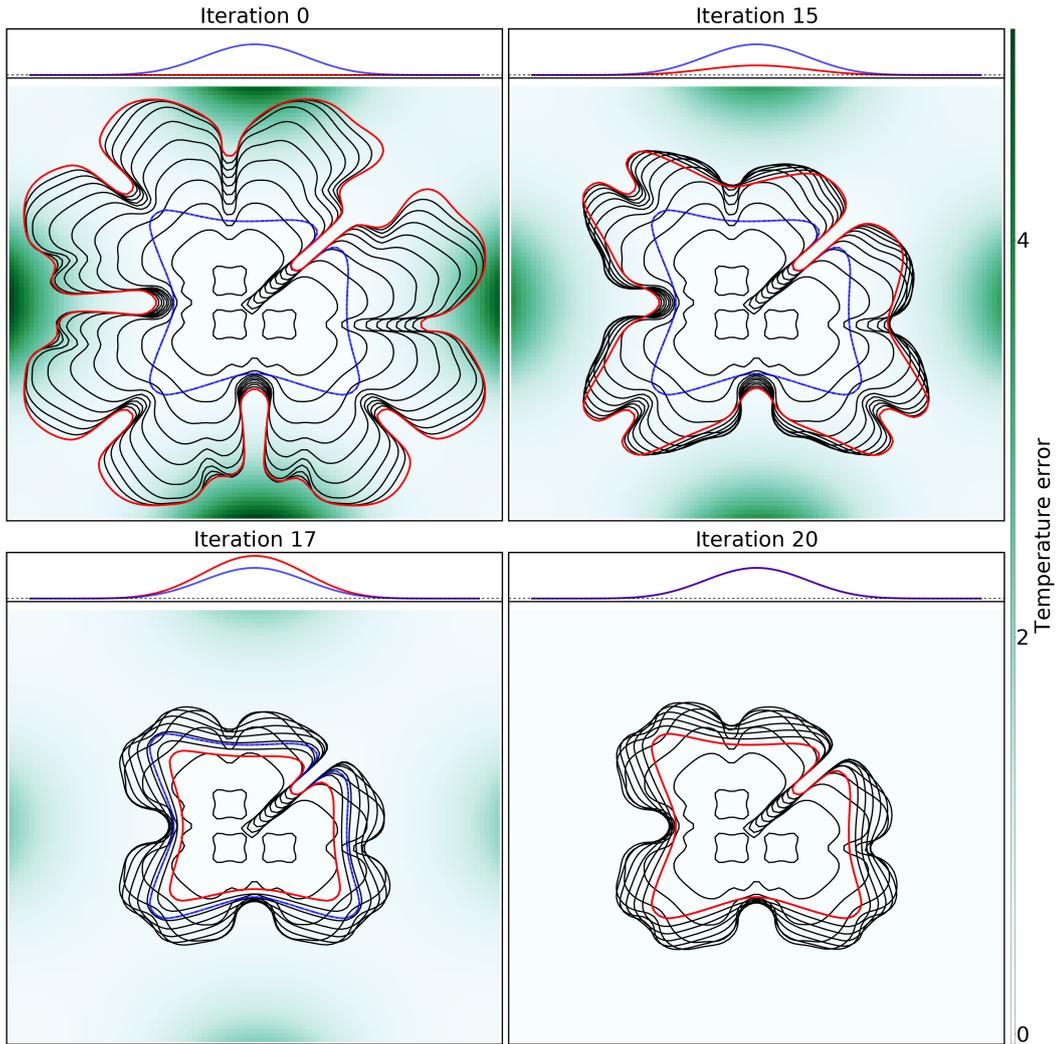
### 6.2.3 Growing crystals with topology changes

In this last case, we consider three crystals asymmetrically disposed in an undercooled liquid. The crystals will grow and eventually merge. The objective of this optimization is to drive the final shape towards the desired one by acting on the boundaries of the whole domain, and thereby to suppress the anisotropy effects. The surface tension coefficient is set to  $\varepsilon_\kappa = 0.0005$  and the velocity coefficient to  $\varepsilon_V = 0.002$ .

We choose a relatively small surface tension coefficient with respect to the velocity one in order to allow for strong dendritic formation and to examine the optimization algorithm in a challenging case where the topology remains complex. The anisotropy effects are added by setting  $\alpha_0 = \pi/2$  and  $M = 4$  in Equation 5.5. The heat flux induced through actuation will have to compete with these effects in order to drive the interface towards the desired one.

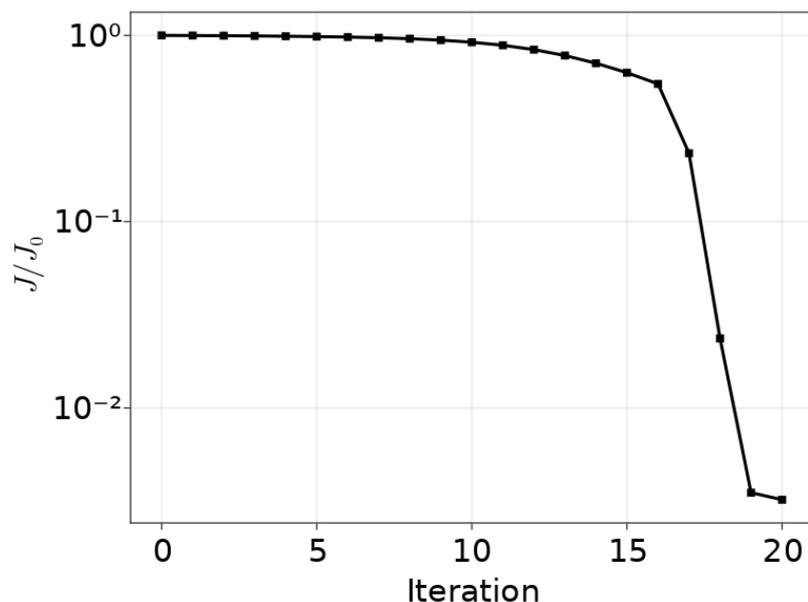
The simulations are run in a  $4 \times 4$  domain with  $N = 100$  until a final time  $t_f = 0.45$ . The under-cooling initial temperature is set to  $T_\infty = -0.6$  and the control variable  $w$  is of Neumann type. In contrary to the previous case, we need to restrict the basis used for the actuator (by fitting only two parameters) in order to descend towards the global minimum. The following parametrization is used

$$u(x, p) = p_1((1 + \cos(\pi/8x))/2)^4 + p_2((1 + \sin(\pi/8x))/2)^4. \quad (6.6)$$



**Figure 6.6:** Iterations 0, 15, 17 and 20 of the optimization procedure for the growing crystals test case. The blue curve represents the desired shape and the red one the final position of the interface at a given iteration. The final time is  $t_f = 0.45$  and the interface is plotted with a time step of 0.045. The color map corresponds to the error in the temperature field. The inset shows the actuator  $w$  at a given iteration (red) and the desired one (blue).

Figure 6.6 shows the evolution of the surface. The minimum is attained after 20 iterations of the optimization algorithm corresponding to 25 solutions of the forward problem (FP) and 20 solutions of the adjoint problem (AP). It can be seen that initially the crystal is driven towards the domain corners by the anisotropy parameters. Through the optimization procedure however, the final crystal shape tends towards the desired one. Moreover, the topology changes are implicitly taken into account thanks to the level set method.



**Figure 6.7:** Normalized cost functional as a function of the iterations of the L-BFGS optimization procedure for the growing crystals case.

Looking at the variation of the cost functional through the optimization procedure (Figure 6.7), we can notice an initial plateau until iteration 15. This is caused by the large difference in the initial shape – at iteration 0 with  $w = 0$  – and the desired one. By starting elsewhere in the parameter space of  $\mathcal{J}$ , the descent towards the minimum might occur faster. Table 6.1 summarizes the results obtained for each case.

Case	Optimization parameters				Results		
	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\mathcal{J}$ calls	$\nabla\mathcal{J}$ calls	$\mathcal{J}_{final}/\mathcal{J}_0$
Melting circle	1	10	0	$10^{-3}$	66	16	$1.81 \times 10^{-3}$
Mullins-Sekerka	1	10	0.1	$10^{-4}$	54	10	$1.89 \times 10^{-3}$
Growing crystals	1	1	0	$10^{-2}$	25	20	$3.21 \times 10^{-3}$

**Table 6.1:** Optimization parameters and final results of the different considered cases. The columns  $\mathcal{J}$  and  $\nabla\mathcal{J}$  correspond to the number of FP and AP calls respectively.

### 6.3 Comparison with derivative-free methods

In the previous section, we showed that the adjoint-based optimization procedure appears to be a robust algorithm to control the shape of a melting or solidification front, even in the presence of dendritic instabilities and anisotropic effects. We now compare the results to a derivative-free method (where no information on the gradient is required), the Particle Swarm optimization algorithm [Zhan *et al.* 2009]. This method attempts to improve global convergence by switching between four evolutionary states: exploration, exploitation, convergence, and jumping out. In the jumping out state it tries to take the best particle and move it away from its local optimum, to improve the ability to find a global one.

We test the derivative-free optimization procedure on the same three cases presented previously using the same parameters  $\beta_1$  to  $\beta_4$  for the cost functional. Table 6.2 summarizes the comparison between both methods. As expected, the derivative-free one attains a lower minimum at a cost of a much higher number of function evaluation (one order of magnitude higher).

Case	L-BFGS			Particle Swarm	
	$\mathcal{J}$ calls	$\nabla\mathcal{J}$ calls	$\mathcal{J}_{final}/\mathcal{J}_0$	$\mathcal{J}$ calls	$\mathcal{J}_{final}/\mathcal{J}_0$
Melting circle	66	16	$1.81 \times 10^{-3}$	1012	$2.04 \times 10^{-4}$
Mullins-Sekerka	54	10	$1.89 \times 10^{-3}$	933	$9.13 \times 10^{-5}$
Growing crystals	25	20	$3.21 \times 10^{-3}$	2120	$1.11 \times 10^{-4}$

**Table 6.2:** Comparison between the L-BFGS and the Particle swarm method for the three considered cases.

As a future work, one could solve optimization cases where the flow in the liquid phase is considered. The control of the onset of the Rayleigh-Bénard instability through the global Rayleigh number could be one the goals. The derivation of the continuous adjoint in that case might required some simplifications. The flow velocity  $\mathbf{u}$  would only be solved in the forward problem, leading to an incomplete adjoint derivation.

## Part II

# Physico-mathematical models for the moving contact line

# Two-phase Navier-Stokes equations and Phase-Field extensions

---

## Contents

<b>7.1</b>	<b>Two-phase flows in the one-fluid formulation . . . . .</b>	<b>95</b>
<b>7.2</b>	<b>Phase-Field extensions . . . . .</b>	<b>97</b>

---

## 7.1 Two-phase flows in the one-fluid formulation

In the first part of this work, we studied melting and solidification processes, modeled by the two-phase Stefan problem (Chapter 1), where the motion of the interface was solely a function of the heat transfer across the interface between a fluid and a solid phase. In that case, when considering the flow in the fluid phase, we solved the Navier-Stokes equations in the Boussinesq approximation. In this second part of the dissertation, we will study physical problems governed by the incompressible two-phase Navier-Stokes equations.

Flows of immiscible fluids are ubiquitous in Nature and in everyday life; for example in sea waves, rain drops or in bubbles rising in a glass of sparkling water. The air-water flows are characterized by large density and viscosity ratios as well as high surface tension. This interfacial force is an energy per area that resists the creation of new interface. These disparities in the material properties across the interface will tend to generate complex phenomena involving different spatial and temporal scales. The particular case of the motion of a fluid-fluid interface over a solid substrate [Shikhmurzaev 1994, Blake *et al.* 1999, Eggers 2004, Wilson *et al.* 2006, Fullana *et al.* 2020, Lācis *et al.* 2020, Lācis *et al.* 2022] will be the focus our study and will be detailed in Chapter 8.

In the context of numerical methods for partial differential equations, the resolution of the the incompressible two-phase Navier-Stokes equations remains a challenge. The incompressibility constraint must be taking into account at each instant and the high density and viscosity ratios in presence of distorted interfaces leads to ill-conditioned linear systems. Moreover, in order to avoid any imbalance between discrete surface tension and pressure gradient terms, an accurate representation of the interfacial force is required [Scardovelli & Zaleski 1999, Tryggvason *et al.* 2011, Popinet 2018].

The incompressible, variable-density, Navier-Stokes equations with surface tension can be written as the momentum equation augmented with the capillary effect

$$\rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot (2\mu \mathbf{D}) + \sigma \kappa \delta_s \mathbf{n} + \mathbf{g}, \quad (7.1)$$

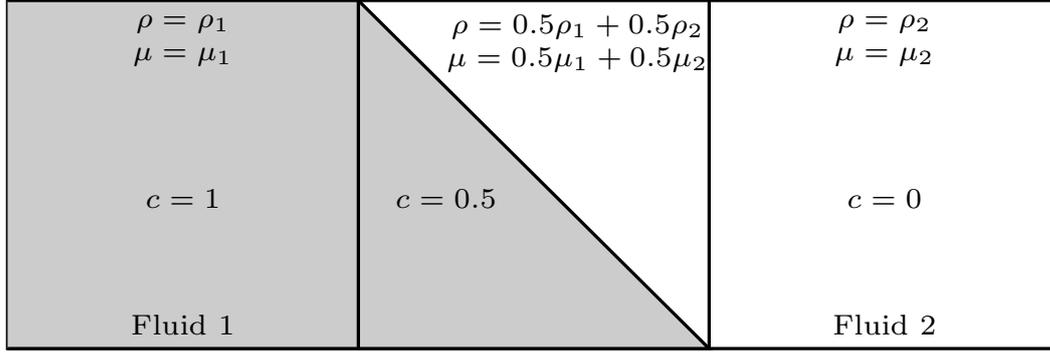
supplemented by the divergence-free condition

$$\nabla \cdot \mathbf{u} = 0, \quad (7.2)$$

and a transport equation for the density

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (7.3)$$

with  $\mathbf{u}$  the fluid velocity,  $\rho$  the fluid density,  $\mu$  the fluid viscosity and  $\mathbf{D}$  the deformation tensor defined as  $D_{ij} = (\partial_i u_j + \partial_j u_i)/2$ ,  $\sigma$  the surface tension coefficient,  $\kappa$  the curvature,  $\delta_s$  the Dirac distribution expressing the fact that the surface tension term is concentrated at the interface,  $\mathbf{n}$  the unit normal to the interface and  $\mathbf{g}$  the acceleration of gravity.



**Figure 7.1:** Example of the discrete representation of the density and viscosity in the one-fluid formulation.

In the one-fluid formulation, a color function  $c(\mathbf{x}, t)$  is used to characterize the interface location and the fluid properties (density and viscosity). The volume fraction  $c(\mathbf{x}, t)$  will be chosen as the volume fraction occupied by a reference phase. The density and viscosity are now defined by

$$\begin{aligned} \rho(c) &\equiv c\rho_1 + (1 - c)\rho_2, \\ \mu(c) &\equiv c\mu_1 + (1 - c)\mu_2, \end{aligned} \quad (7.4)$$

with  $\rho_1$ ,  $\rho_2$  and  $\mu_1$ ,  $\mu_2$  the densities and viscosities of the first and second fluids respectively. Figure 7.1 shows an example of the discrete representation of the density and viscosity. The advection of the density (Equation 7.3) is then naturally replaced by the advection equation for the color function

$$\partial_t c + \nabla \cdot (c\mathbf{u}) = 0. \quad (7.5)$$

In part 1 of this work, this equation was solved using an implicit-explicit scheme to advect a level set function (Chapter 4). In the context of the two-phase Navier-Stokes equations, when considering the sharp-interface model – as opposed to the diffuse interface model described in the following section – Equation 7.5 will be solved using a conservative, non-diffusive geometric Volume-Of-Fluid (VOF) scheme. The VOF method and the treatment of the boundary conditions will be detailed in Sections 9.1 and 9.2.

## 7.2 Phase-Field extensions

In the Phase-Field method, the governing equations are derived from the thermodynamic potentials of the system, together with the assumption of a surface energy associated with an interface [Amberg 2003]. It is, therefore, possible to consider different physical situations with relative ease. It is also straightforward to implement numerically, since interfaces are not tracked explicitly. Instead, a variable is introduced that has different constant values in the two phases with a steep transition between the two in the diffuse interface [Jacqmin 2000, Carlson 2012]. One of the major drawbacks of this method, however, is that width of the interface  $\varepsilon$  must be small to match the proper interface dynamics.

Similarly to the one-fluid formulation, the Phase-Field model introduces a phase variable  $C(\mathbf{x}, t)$  ranging from 1 to  $-1$ . This phase-field variable  $C(\mathbf{x}, t)$  is governed by a convection-diffusion equation

$$\frac{\partial C}{\partial t} = \mathbf{F}_d - \mathbf{F}_c, \quad (7.6)$$

where  $\mathbf{F}_d$  is the diffusive flux and  $\mathbf{F}_c$  is the convective flux. The latter, in the context of incompressible flows, takes the simple form

$$\mathbf{F}_c = \mathbf{u} \cdot \nabla C, \quad (7.7)$$

and the diffusive flux is

$$\mathbf{F}_d = -M \nabla \phi, \quad (7.8)$$

where  $M$  is a proportionality coefficient called the Phase-Field mobility and  $\phi$  the chemical potential defined as

$$\phi = \beta \Psi'(C) - \alpha \nabla^2 C. \quad (7.9)$$

In the chemical potential, we have two parameters  $\alpha$  and  $\beta$ , which are related to the surface tension  $\sigma$  and the characteristic thickness of the diffuse interface  $\varepsilon$  as

$$\sigma = \frac{2}{3} \sqrt{2\alpha\beta}, \quad \varepsilon = \sqrt{\frac{\alpha}{\beta}}. \quad (7.10)$$

In addition, it contains the derivative of the standard double-well potential

$$\Psi(C) = \frac{(C+1)^2(C-1)^2}{4}. \quad (7.11)$$

With the definitions above (Equations 7.6 to 7.11), the convection-diffusion equation, referred to as the Cahn-Hilliard equation [Cahn & Hilliard 1958], can be written as

$$\frac{\partial C}{\partial t} = \nabla \cdot [M \nabla (\beta \Psi'(C) - \alpha \nabla^2 C)] - \mathbf{u} \cdot \nabla C. \quad (7.12)$$

Equation 7.12 is the Phase-Field analog of Equation 7.5. The boundary conditions for this fourth-order partial differential equation will be detailed in our study of dynamic wetting by the Phase-Field method (Section 10.3.1). Taking the same notation as in Equation 7.1, the momentum equation in the Navier-Stokes equations becomes

$$\rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot (2\mu \mathbf{D}) - C \nabla \phi + \mathbf{g}, \quad (7.13)$$

where  $-C \nabla \phi$  corresponds to the surface tension force and acts over the diffuse interface region. This form of the surface tension forcing is the so called potential form [Jacqmin 1999]. Moreover, the density and viscosity are now defined through the Phase-Field variable  $C$  as

$$\begin{aligned} \rho(C) &\equiv \rho_1 \frac{C+1}{2} + \rho_2 \frac{C-1}{2}, \\ \mu(C) &\equiv \mu_1 \frac{C+1}{2} + \mu_2 \frac{C-1}{2}. \end{aligned} \quad (7.14)$$

We now state the boundary conditions in the presence of a contact line (see Chapter 8) on a solid, characterized by its equilibrium angle (the angle between the fluid-fluid interface and the wall). The convection-diffusion equation (Equation 7.12) is a fourth-order partial differential equation and requires two boundary conditions. First, we impose a non-equilibrium wetting condition [Jacqmin 2000, Qian *et al.* 2003b] on the solid wall,

$$-\mu_f \varepsilon \left( \frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C \right) = \alpha \nabla C \cdot \mathbf{n} - \sigma \cos \theta_e g'(C), \quad (7.15)$$

where  $\mu_f$  is a contact line friction parameter, having the same units as bulk dynamic viscosity. Here,  $\theta_e$  is the equilibrium contact angle and

$$g(C) = 0.5 - 0.75C + 0.25C^3, \quad (7.16)$$

is a switching function describing a smooth transition between both phases. The unit normal vector  $\mathbf{n}$  is directed from the fluid to the surrounding solid.

If one sets  $\mu_f = 0$ , the contact angle is always enforced to the equilibrium angle  $\theta_e$ . This will be the case in the toy model with comparing PF to VOF simulations (Section 10.3). Non-zero contact line friction allows the dynamic contact angle to

---

evolve naturally as a function of contact line speed. The second boundary condition for the phase function is a zero diffusive flux of chemical potential through the boundaries

$$\nabla\phi \cdot \mathbf{n} = 0. \tag{7.17}$$

In the following chapter, we will present the main topic of our study: the dynamic contact lines, the singularities involved and the models to accurately simulate the motion of the fluid-fluid-solid intersection point.



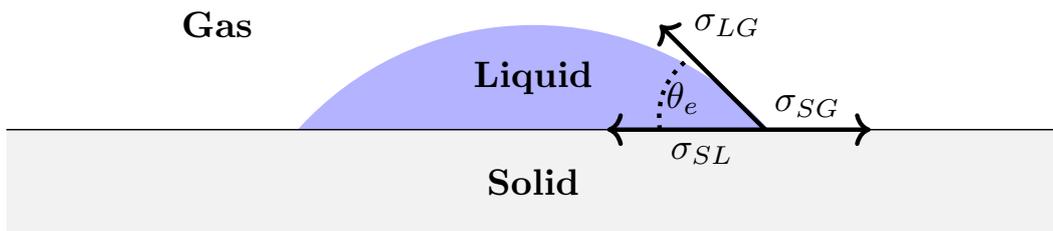
# Singularities in flows with contact lines

## Contents

8.1	The static contact line . . . . .	101
8.2	Paradoxes and singularities in the dynamic case . . . . .	102

## 8.1 The static contact line

Wetting of solids by liquids, in which a liquid displaces another fluid on a solid substrate, is an ubiquitous phenomenon with applications ranging from coating and tear films on the cornea to micro-layer formation in wall boiling and CO<sub>2</sub> sequestration. However, despite the abundance of applications, the precise mechanism of wetting is only partially understood. From the numerical modeling point of view, difficulties arise due to the highly multi-scale nature of the problem (scales extending from macroscopic to molecular lengths). One of the main challenges in continuum theory is the effective representation of molecular phenomena close to the contact line.



**Figure 8.1:** Schematic of the static contact line. The contact line formed at the intersection of a gas-liquid and a solid boundary is element of many natural and several technological processes. In the static case, the equilibrium angle is determined by Young's law.

In the static case, for a droplet at equilibrium (Figure 8.1), the equilibrium contact angle is well known to be determined by Young's law

$$\sigma_{SG} - \sigma_{SL} = \sigma_{LG} \cos(\theta_e) \quad (8.1)$$

where  $\sigma_{SG}$ ,  $\sigma_{SL}$  and  $\sigma_{LG}$  are the solid-gas, solid-liquid and liquid-gas surface tensions respectively. Numerically, while still being subject spurious currents depending on the characteristics of the system (see Appendix D), the static case does not pose any particular issues.

## 8.2 Paradoxes and singularities in the dynamic case

The difficulties arise when considering a dynamic contact line. In fact, the motion of the contact line poses, since Huh and Scriven [Huh & Scriven 1971] a remarkable problem because of the contradiction between the no-slip condition on the substrate and the motion of the contact line. Under these assumptions, one naturally ends up with an immobile contact line that is inconsistent with simple observations (for a example a water drop sliding on a leaf). This no-slip paradox induces a stress singularity at the contact line, summarized in the words of Huh and Scriven:

Not even Hercules could sink a solid [...]

This paradox does not exist in diffuse interface models, such as Phase-Field (Section 7.2), as the contact line advances through diffusion even in the no-slip scenario. However a special treatment of the boundary condition at the wall is required for sharp interface models, such as Volume-Of-Fluid (Section 7.1). One the first methods that was proposed to relieve the force singularity at the contact line was to simply relax the no-slip condition. In the slip length theory, expressed as a Navier boundary condition (NBC), a term, proportional to the shear velocity profile, is added in the boundary condition such that

$$u_x|_{y=0} - \lambda \frac{\partial u_x}{\partial y}|_{y=0} = 0, \quad (8.2)$$

where  $u_x|_{y=0}$  is the tangential velocity at the wall (located at  $y = 0$ ) and  $\lambda$  the slip length. Translating this boundary condition into a relation for the contact line speed yields

$$U_{CL} \sim \lambda \frac{\partial u}{\partial y}|_{y=0}, \quad (8.3)$$

where  $U_{CL}$  is the contact line speed. This simple slip model has proven to be sufficient to solve the contact line motion in some hydrodynamic cases [Wilson *et al.* 2006, Liu *et al.* 2016, Liu *et al.* 2018, Fullana *et al.* 2020, Lācis *et al.* 2020]. In Section 9.2.1, we will present our results on a curtain coating setup using the Navier boundary condition.

Nonetheless, the slip model still suffers from a logarithmic divergence of the capillary pressure as shown in [Devauchelle *et al.* 2007]. A further regularization of the solution is possible by considering a second-order slip (in analogy with the porous media theory)

$$U_{CL} \sim \lambda \frac{\partial u}{\partial y}|_{y=0} + \lambda^2 \frac{\partial^2 u}{\partial y^2}|_{y=0}. \quad (8.4)$$

In that case the contact line speed is also proportional to the curvature of the velocity profile in the vertical direction. In our work, this model will be referred to as the super-slip model (Section 10.1).

Another parameter to take into account is the contact angle. When the contact line enters in motion, the angle formed by the fluid-fluid interface and the solid substrate varies, becoming a dynamic contact angle, denoted  $\theta_d$ . The modeling of this angle is very controversial among experts, there is no consensual theory on whether it is an predictable quantity – a function of the contact line speed and the bulk parameters – or an output of the system [Cox 1986, Shikhmurzaev 1994, Blake *et al.* 1999, Blake & Shikhmurzaev 2002, Eggers 2004, Eggers & Evans 2005]. In the Cox-Voinov theory, the dynamic contact angle is a function of the normalized contact line  $\text{Ca}_{\text{CL}}$

$$g(\theta_d) = g(\theta_e) + \text{Ca}_{\text{CL}} \log\left(\frac{L}{l}\right), \quad (8.5)$$

where  $g$  is function describing the curved interface (defined later). The other parameters are  $L$  and  $l$  corresponding to two distinct length scales. A comparison between different options for  $L$  and  $l$  can be found in [Legendre & Maglio 2015]. They concluded that the models incorporating dynamic contact angles with a numerical cutoff length  $l$  better represent some experiments. In Section 9.3.2, we will use a dynamic angle model implemented in our VOF framework and compare it to a spreading drop experiment.

Moreover, in [Fricke *et al.* 2019] the authors demonstrated that there exists a second paradox related to that dynamic contact angle. From kinematics relations, they showed that the shear velocity at the contact line is proportional to the variation of the dynamic contact angle  $\theta_d$

$$\frac{\partial u}{\partial y}\Big|_{y=0} \sim \frac{d\theta_d}{dt}. \quad (8.6)$$

Considering now a steady state motion where the contact line speed is not null with a slip model (Equation 8.2), we have

$$U_{\text{CL}} \sim \lambda \frac{d\theta_d}{dt}. \quad (8.7)$$

In steady state, the angle is constant, implying that

$$\frac{d\theta_d}{dt} = 0, \quad (8.8)$$

which contradicts previous assumption on the contact line speed  $U_{\text{CL}} \neq 0$ . The way to relax this singularity is to allow a deviation between the dynamic contact angle and the equilibrium one ( $\theta_e$ ). In the generalized Navier boundary condition (GNBC), derived from thermodynamics considerations [Qian *et al.* 2003b, Qian *et al.* 2006],

the uncompensated Young's stress is introduced as an additional term in the NBC

$$U_{\text{CL}} \sim \lambda \frac{\partial u}{\partial y} \Big|_{y=0} + \frac{\sigma}{\mu} (\cos \theta_d - \cos \theta_e), \quad (8.9)$$

where  $\sigma$  is the surface tension and  $\mu$  the viscosity of the primary fluid. This boundary condition will be tested in a withdrawing plate setup where an accurate modeling of the dynamic contact line is crucial to predict the onset wetting failure.

In the following chapters we will present the implementation of the models and their application to physical setups. In Chapter 9, the classical slip and dynamic angle models will be described in depth and tested on different problems. In Chapter 10, the super-slip and GNBC models will be presented as well as the toy model, where we will relate the mass flux present in the Phase-Field model to a simple slip and dynamic angle relation in the VOF method.

# Numerical methods for contact lines

## Contents

<b>9.1</b>	<b>The Volume-Of-Fluid method . . . . .</b>	<b>105</b>
<b>9.2</b>	<b>Classical models for contact lines . . . . .</b>	<b>109</b>
9.2.1	The Navier boundary condition . . . . .	109
9.2.2	Cox-Voinov relation for dynamic contact angles . . . . .	110
<b>9.3</b>	<b>Application to physical systems . . . . .</b>	<b>111</b>
9.3.1	Onset of wetting failure in a curtain coating setup . . . . .	111
9.3.2	Squalane drop spreading with dynamic contact angle . . . . .	119

## 9.1 The Volume-Of-Fluid method

The VOF method for representing fluid interfaces coupled with a flow solver is well-known to be suited for solving interfacial flows [Scardovelli & Zaleski 1999, Popinet & Zaleski 1999, Tryggvason *et al.* 2011]. However, this method, such as others, is subject to spurious currents, with or without contact lines (Appendix D). These currents are lower when considering high capillary numbers, which is usually the case in the context of moving contact lines.

In our study on the moving contact line, we use the free software *Basilisk*, a platform for the solution of partial differential equations on adaptive Cartesian meshes, developed by Stéphane Popinet and co-authors [Afkhami & Bussmann 2008, Afkhami & Bussmann 2009, Popinet 2009, Popinet 2015, Afkhami *et al.* 2017, Popinet 2018].

As mentioned in Chapter 7, we consider the incompressible Navier-Stokes equations with variable density and surface tension:

$$\begin{aligned}\rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) &= -\nabla p + \nabla \cdot (2\mu \mathbf{D}) + \sigma \kappa \delta_s \mathbf{n} + \mathbf{g}, \\ \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) &= 0, \\ \nabla \cdot \mathbf{u} &= 0\end{aligned}\tag{9.1}$$

with  $\mathbf{u}$  the fluid velocity,  $\rho$  the fluid density,  $\mu$  the fluid viscosity and  $\mathbf{D}$  the deformation tensor defined as  $D_{ij} = (\partial_i u_j + \partial_j u_i)/2$ ,  $\sigma$  the surface tension coefficient,  $\kappa$  the curvature,  $\delta_s$  the Dirac distribution function used for the sharp interface model,

$\mathbf{n}$  the unit normal to the interface and  $\mathbf{g}$  the acceleration of gravity.

For a two-phase flow, the volume fraction  $c(\mathbf{x}, t)$  is defined as the integral of the first fluid's characteristic function in the control volume. The volume fraction  $c(\mathbf{x}, t)$  is used to define the density and viscosity in the control volume

$$\begin{aligned}\rho(c) &\equiv c\rho_l + (1 - c)\rho_g, \\ \mu(c) &\equiv c\mu_l + (1 - c)\mu_g,\end{aligned}\tag{9.2}$$

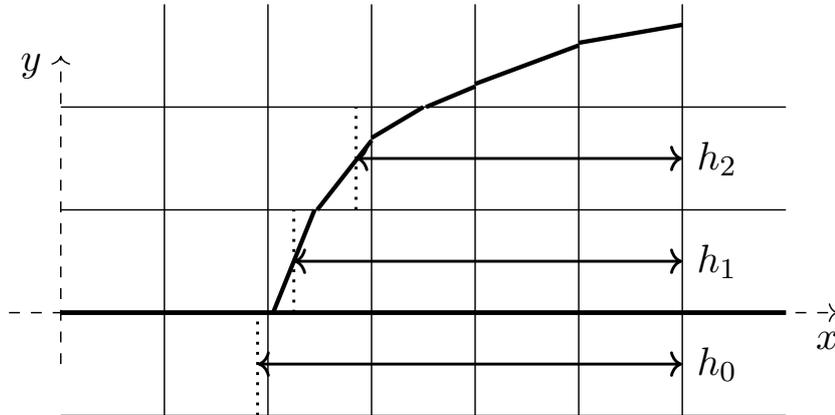
with  $\rho_l, \rho_g$  and  $\mu_l, \mu_g$  the densities and viscosities of the liquid phase and the gas phase respectively.

The advection equation for the density is then replaced by the equation for the volume fraction

$$\partial_t c + \nabla \cdot (c\mathbf{u}) = 0$$

The projection method is used to solve the incompressible Navier-Stokes equations combined with a Bell-Collella-Glaz advection scheme and a VOF method for interface tracking.

The resolution of the surface tension term is directly dependent on the accuracy of the curvature calculation. The Height-Function methodology is a VOF-based technique for calculating interface normals and curvatures [Afkhami & Bussmann 2008, Afkhami & Bussmann 2009]. About each interface cell, fluid 'heights' are calculated by summing fluid volume in the grid direction closest to the normal of the interface.



**Figure 9.1:** Construction of the 2D height-functions near the contact line [Afkhami & Bussmann 2008].

In two dimensions, a  $7 \times 3$  stencil around an interface cell is constructed and the heights are evaluated by summing volume fractions horizontally (Figure 9.1)

$$h_j = \sum_{k=i-3}^{k=i+3} c_{j,k} \Delta,$$

with  $c_{j,k}$  the volume fraction and  $\Delta$  the grid spacing. The heights are then used to

compute the the interface normal  $\mathbf{n}$  and the curvature  $\kappa$

$$\begin{aligned}\mathbf{n} &= (h_x, -1), \\ \kappa &= \frac{h_{xx}}{(1 + h_x^2)^{3/2}},\end{aligned}\tag{9.3}$$

where  $h_x$  and  $h_{xx}$  are discretized using second-order central differences.

It is important to note that a numerical specification of the contact angle affects the overall flow calculation in two ways:

- It defines the orientation of the VOF reconstruction in cells that contain the contact line.
- It influences the calculation of the surface tension term by affecting the curvature computed in cells at and near the contact line.

The orientation of the interface, characterized by the contact angle – the angle between the normal to the interface at the contact line and the normal to the solid boundary – is imposed in the contact line cell.

We now present some of the characteristics of the Volume-of-Fluid Navier-Stokes solver. For further details, we refer the reader to [Scardovelli & Zaleski 1999, Popinet & Zaleski 1999, Popinet 2009, Popinet 2015, Popinet 2018]. A staggered in time discretization of the volume-fraction/density and pressure combined with a time-splitting projection method leads to the following time discretization

$$\begin{aligned}\rho_{n+\frac{1}{2}} \left[ \frac{\mathbf{u}_* - \mathbf{u}_n}{\Delta t} + \mathbf{u}_{n+\frac{1}{2}} \cdot \nabla \mathbf{u}_{n+\frac{1}{2}} \right] &= \nabla \cdot \left[ \mu_{n+\frac{1}{2}} (D_n + D_*) \right] + (\sigma \kappa \delta_s \mathbf{n})_{n+\frac{1}{2}}, \\ \frac{c_{n+\frac{1}{2}} - c_{n-\frac{1}{2}}}{\Delta t} + \nabla \cdot (c_n \mathbf{u}_n) &= 0, \\ \mathbf{u}_{n+1} &= \mathbf{u}_* - \frac{\Delta t}{\rho_{n+\frac{1}{2}}} \nabla p_{n+\frac{1}{2}},\end{aligned}\tag{9.4}$$

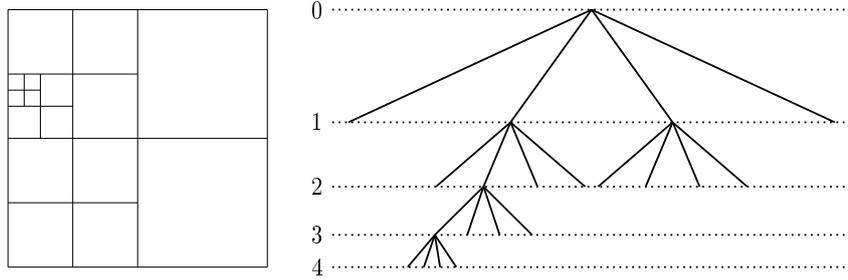
which requires the solution of the Poisson equation

$$\nabla \cdot \left[ \frac{\Delta t}{\rho_{n+\frac{1}{2}}} \nabla p_{n+\frac{1}{2}} \right] = \nabla \cdot \mathbf{u}_*.\tag{9.5}$$

The momentum equation can be rewritten as

$$\begin{aligned}\frac{\rho_{n+\frac{1}{2}}}{\Delta t} \mathbf{u}_* - \nabla \cdot \left[ \mu_{n+\frac{1}{2}} \mathbf{D}_* \right] &= \nabla \cdot \left[ \mu_{n+\frac{1}{2}} \mathbf{D}_n \right] + (\sigma \kappa \delta_s \mathbf{n})_{n+\frac{1}{2}} + \rho_{n+\frac{1}{2}} \left[ \frac{\mathbf{u}_n}{\Delta t} - \right. \\ &\left. \mathbf{u}_{n+\frac{1}{2}} \cdot \nabla \mathbf{u}_{n+\frac{1}{2}} \right],\end{aligned}\tag{9.6}$$

where the right-hand side depends only on values at time  $n$  and  $n + 1/2$ . This equation is solved using a multilevel Poisson solver. The velocity advection term  $\mathbf{u}_{n+\frac{1}{2}} \cdot$



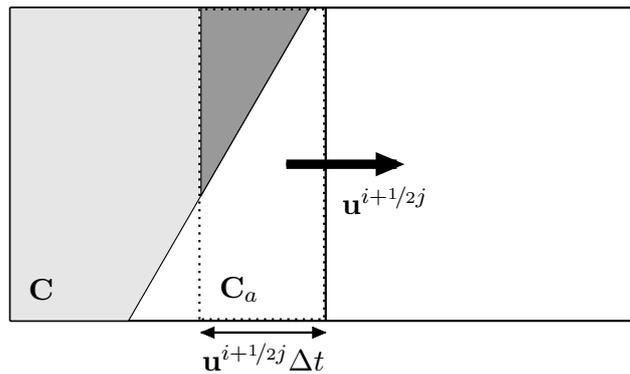
**Figure 9.2:** Example of the quadtree discretization and the corresponding tree representation.

$\nabla \mathbf{u}_{n+\frac{1}{2}}$  is estimated using the Bell-Collela-Glaz [Bell *et al.* 1989, Popinet 2003] second-order upwind scheme. Space is discretized using a quadtree partitioning in 2D (Figure 9.2).

All the variables are collocated at the center of each square discretization volume. Consistently with a finite-volume formulation, the variables are interpreted as the volume-averaged values for the corresponding discretization volume. A projection method is used for the spatial discretization of the pressure correction equation and the associated divergence in the Poisson equation.

To solve the advection equation the geometrical VOF scheme is used and proceeds in two steps:

1. Interface reconstruction.
2. Geometrical of flux estimation and interface advection.



**Figure 9.3:** Example of geometrical flux estimation.

The reconstruction is a ‘piecewise linear interface calculation’ (PLIC), followed by a Lagrangian advection. In the PLIC technique, given a volume fraction  $c(\mathbf{x}, t)$  and an approximate normal vector  $\mathbf{n}$ , a linear interface is constructed within each interface cell, which corresponds exactly to  $c(\mathbf{x}, t)$  and  $\mathbf{n}$ . In Figure 9.3, we illustrate the principle of geometrical flux estimation. The total volume which will be fluxed to

the right-hand neighbor is delimited with a dashed line. The fraction of this volume occupied by the first phase is indicated by the dark grey triangle.

## 9.2 Classical models for contact lines

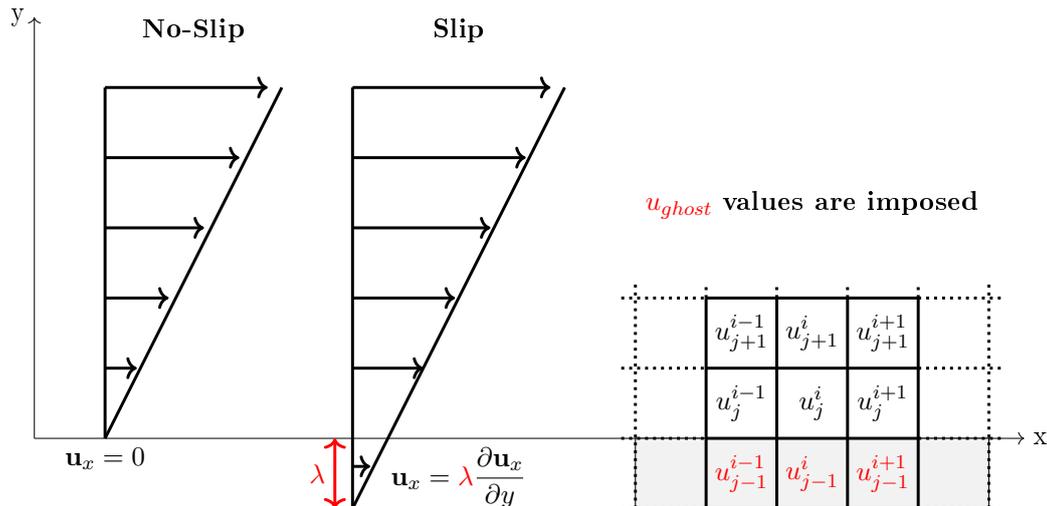
### 9.2.1 The Navier boundary condition

The dynamic contact line introduces a paradox at the triple point, where the no-slip boundary condition or Dirichlet boundary condition at the solid interface induces a non-integrability of the solution. Nevertheless, in the VOF method such a contradiction does not take place as the volume fraction is advected using the velocity half-cell away from the wall. In their paper, the authors of [Afkhani *et al.* 2017], showed that there exists a ‘numerical slip’ that is mesh dependent. In order to control such slip, it is useful to introduce a Navier boundary condition (NBC) to explicitly define a physical slip length  $\lambda$ , that will be used as a fitting parameter in our simulations [Legendre 2013, Sui *et al.* 2014, Legendre & Maglio 2015, Fullana *et al.* 2020, Lācis *et al.* 2020, Lācis *et al.* 2022].

The NBC corresponding to a slip model in a 2D configuration with the substrate at  $y = 0$  can be written as follows

$$\begin{aligned} u_x|_{y=0} - \lambda \frac{\partial u_x}{\partial y}|_{y=0} &= U, \\ u_y|_{y=0} &= 0, \end{aligned} \quad (9.7)$$

with  $u_x|_{y=0}$  and  $u_y|_{y=0}$  the  $x$  and  $y$  component of the velocity at the solid boundary,  $\lambda$  the slip length, and  $U$  the prescribed velocity of the moving substrate.



**Figure 9.4:** Left side : velocities profiles at the solid interface for the no-slip and slip boundary conditions. Right side : 3x3 stencils with the ghost boundary layer used to impose the boundary condition.

Stencils located close enough to the boundaries of the domain will extend beyond it (Figure 9.4). The stencil values outside the domain (ghost values) need to be initialized. These values that depend on the values inside the domain are set in order to provide the discrete equivalent of the NBC as follows

$$\frac{u_x[ghost] + u_x[]}{2} + \lambda \frac{u_x[ghost] - u_x[]}{\Delta} = U,$$

$$\Leftrightarrow u_x[ghost] = \frac{2\Delta}{2\lambda + \Delta} U + \frac{2\lambda - \Delta}{2\lambda + \Delta} u_x[],$$

with  $u_x[ghost]$  the tangential velocity at the ghost cell,  $u_x[]$  the tangential velocity of the cell inside the domain,  $\Delta$  the grid spacing and  $\lambda$  the slip length.

As  $\mathbf{u}$  is solution of the Poisson equation, we also need to define the discrete homogeneous counterpart of the NBC

$$u_x^h[ghost] = \frac{2\lambda - \Delta}{2\lambda + \Delta} u_x[].$$

We will show that the Navier boundary condition coupled with a constant contact angle is sufficient to model the contact line motion. In Section 9.3.1, we study the onset of wetting failure in a curtain coating system, where a liquid falls on a moving plate and starts coating the surface. We are able to reproduce the non-monotonic behaviour of the critical velocity as the liquid flow-rate increases.

Moreover, in [Lācis *et al.* 2020], we benchmarked our VOF simulations with phase-field (PF) and molecular dynamics (MD) simulations in a nanoscopic sheared droplet setup. In that case we used the slip length and the constant contact angle as fitting parameters. The slip had to be localized near the contact line point. The boundary condition for the tangential component in the NBC was modified as follows

$$u_x|_{y=0} - f\left(\frac{x}{\varepsilon}\right) \lambda \frac{\partial u_x}{\partial y}|_{y=0} = U, \quad (9.8)$$

where  $f\left(\frac{x}{\varepsilon}\right)$  is a bell function defined as

$$f\left(\frac{d}{\varepsilon}\right) = \begin{cases} \left[\frac{1 + \cos(\pi d/\varepsilon)}{2}\right]^2 & |d| < \varepsilon, \\ 0 & |d| \geq \varepsilon, \end{cases} \quad (9.9)$$

where  $d$  is the distance to the contact line point and  $\varepsilon$  the width of the bell function (related to interface width in the phase-field model).

### 9.2.2 Cox-Voinov relation for dynamic contact angles

The theory of the moving contact line at small capillary numbers was founded by Voinov [Voinov 1976] and generalized to arbitrary viscosity ratios by Cox [Cox 1986]. The latter clarified the structure of low capillary number problems in terms of the

ratio  $L/l$  between the two length scales where  $L$  numerical factor to be determined. From a general analysis, Cox obtained

$$g(\theta_d) = g(\theta_e) + \text{Ca}_{\text{CL}} \log\left(\frac{L}{l}\right), \quad (9.10)$$

where  $\theta_d$  is the dynamic numerical angle imposed at the contact line,  $\theta_e$  is a static equilibrium contact angle, a property of the substrate and the capillary number is expressed as a function of the contact line speed  $U_{\text{CL}}$

$$\text{Ca}_{\text{CL}} = \frac{\mu_l U_{\text{CL}}}{\sigma}, \quad (9.11)$$

where  $\mu_l$  is the fluid viscosity and  $\sigma$  the surface tension. When the surrounding fluid is of lower viscosity, the function  $g(\theta)$  simplifies to

$$g(\theta) = \int_0^\theta \frac{x - \sin x \cos x}{2 \sin x} dx. \quad (9.12)$$

In Section 9.3.2, we compare our model with experiments of a spreading squalane drop.

## 9.3 Application to physical systems

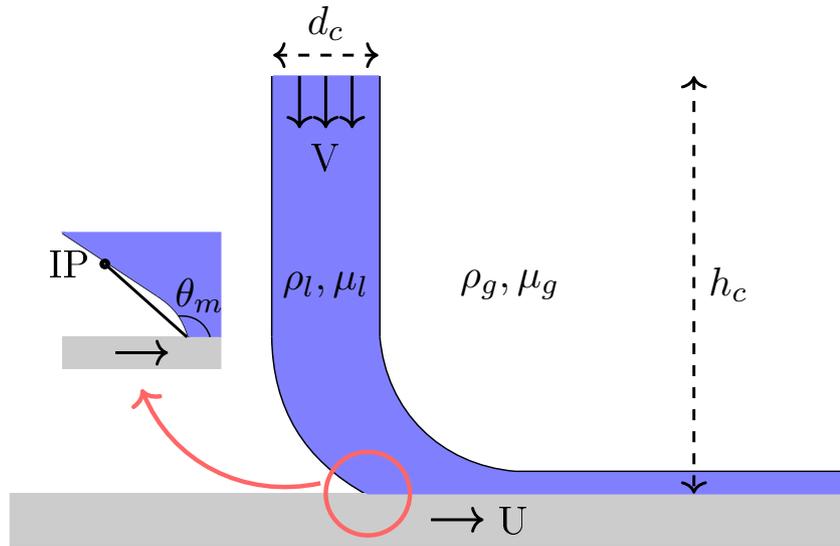
### 9.3.1 Onset of wetting failure in a curtain coating setup

The first system considered is the curtain coating [Fullana *et al.* 2020]. In the curtain coating system (Figure 9.5), a liquid is falling with a velocity  $V$  on a plate moving at velocity  $U$ .

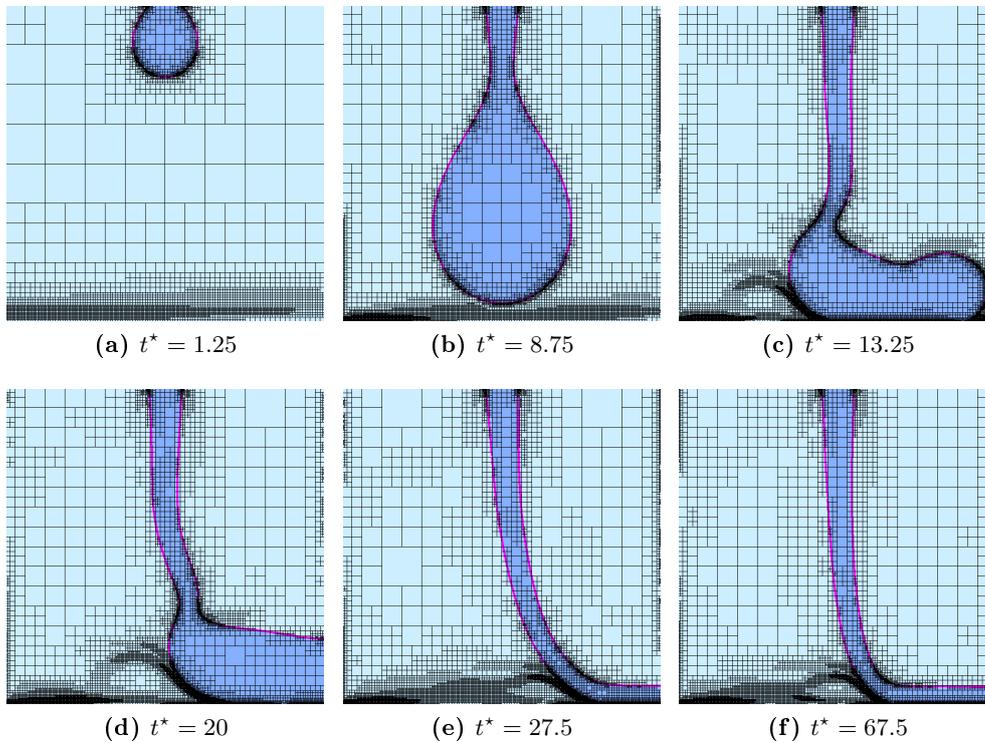
When the liquid reaches the solid substrate, it starts coating the free surface, as shown in the time series example (Figure 9.6). A steady-state solution is only obtained for given sets of physical parameters and the onset of wetting failure can be predicted by studying a range of capillary and Reynolds numbers by varying  $U$  and  $V$ .

We will make the same assumption as Liu [Liu *et al.* 2016]: a constant contact angle coupled with a Navier boundary condition is sufficient to model the dynamic wetting system, provided that the air stresses are taken into account.

In our model, these stresses are directly taken into account in the two-phase Navier-Stokes solver. A constant contact angle  $\theta_m$  is imposed through height functions [Afkhani & Bussmann 2008, Afkhani & Bussmann 2009] and we use the Navier boundary condition for the tangential velocity at the wall. The multi-scale nature of the curtain coating configuration can lead to numerical difficulties, in particular on the resolution of the smallest length scale, the slip length  $\lambda$ .



**Figure 9.5:** Schematic of the curtain coating configuration. The system parameters are:  $h_c$  the curtain height,  $d_c$  the curtain width,  $\rho_l$ ,  $\rho_g$  and  $\mu_l$ ,  $\mu_g$  the densities and viscosities of the liquid phase and the gas phase respectively,  $U$  the substrate velocity,  $V$  the feed flow velocity and  $\theta_m$  the imposed contact angle. The inflection point noted IP corresponds to the point at which the curvature of the interface is zero. The distance from the triple point to the IP will be used as a control quantity.



**Figure 9.6:** Time series of a curtain coating simulation for  $Re = 12$ ,  $Ca = 1.33$ . The dimensionless time  $t^*$  is scaled with the viscous time scale  $t_\mu = \mu_l / \rho_l h_c^2$ .

Moreover, the regularization of the contact line paradox is directly dependent on the accuracy of the solution near the contact line. The adaptive mesh refinement allows a good resolution of the interface at the triple point. We will study the convergence of the solution of the Liu [Liu *et al.* 2016] configuration as we increase the number of grid points per slip length  $\lambda/\Delta$ . The system will then be extended to compare with experiments of [Blake *et al.* 1999] and [Marston *et al.* 2009].

As a first step, to validate our model, we reproduce the curtain coating configuration described in [Liu *et al.* 2016]. We consider a small curtain height  $h_c = 10^{-2}\text{m}$  and small curtain width  $d_c = 10^{-3}\text{m}$  with a large slip length  $\lambda = 10^{-5}\text{m}$ . As the accuracy of the interface reconstruction depends on the resolution of the smallest length scale, these previous considerations drastically decrease the computational cost of this multi-scale problem.

The fluid properties, corresponding to glycerinated water and air, are the following

$$\begin{aligned}\rho_l &= 1000\text{kg.m}^{-3}, & \rho_g &= 1.2\text{kg.m}^{-3}, \\ \mu_l &= 25\text{mPa.s}, & \mu_g &= 0.018\text{mPa.s},\end{aligned}$$

with the surface tension set to  $\sigma = 70\text{mN.m}^{-1}$ . The viscosity ratio  $\mu_g/\mu_l = 7.2 \cdot 10^{-4}$  is kept constant in our computations. The substrate velocity  $U$  is varied from 0.1 to 10  $\text{m.s}^{-1}$  and the feed flow velocity  $V$  from 0.1 to 1  $\text{m.s}^{-1}$ . The dimensionless numbers governing the flow are: the capillary number

$$\text{Ca} = \frac{\mu_l U}{\sigma},$$

varying from 0.1 to 2.5, the Reynolds number

$$\text{Re} = \frac{\rho_l V d_c}{\mu_l},$$

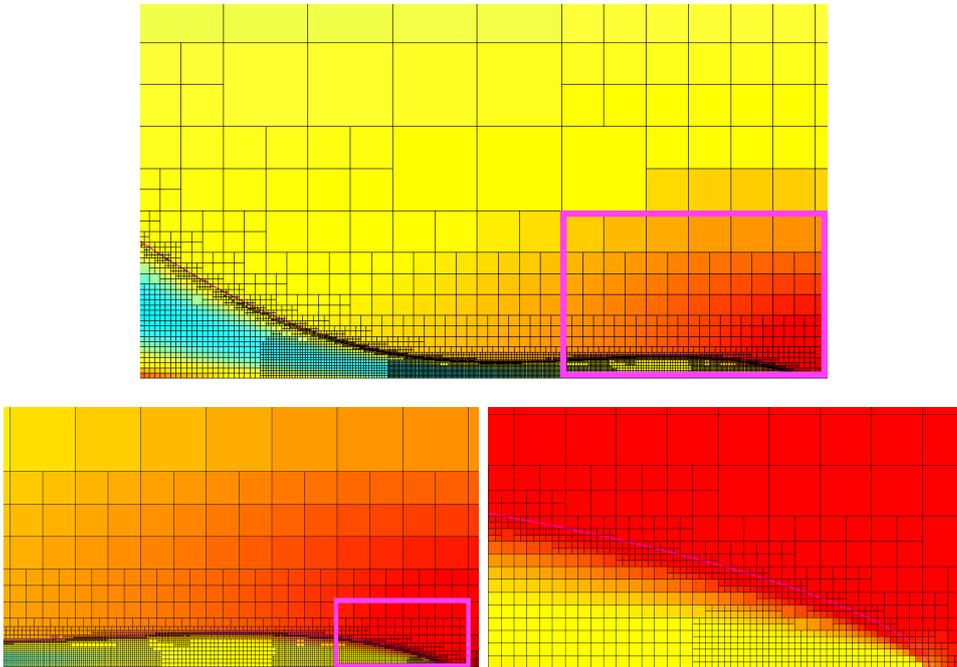
varying from 1 to 40 and the Bond number

$$\text{Bo} = \left(\frac{\rho_l g}{\sigma}\right) \left(\frac{d_c V}{U}\right)^2,$$

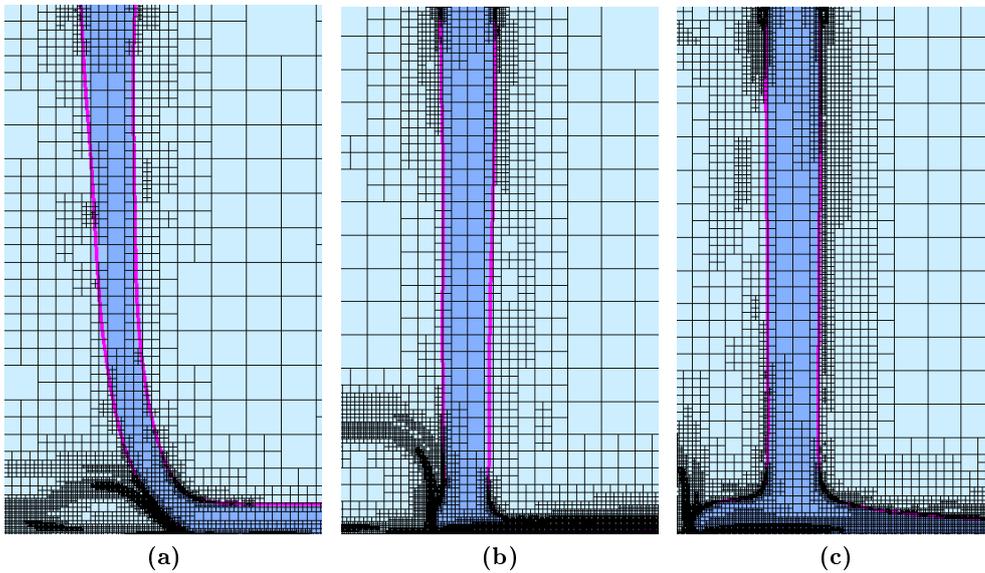
varying from  $10^{-3}$  to  $10^{-1}$ . The contact angle is kept constant,  $\theta_m = 90^\circ$ .

To determine whether the simulation with a given set of physical parameters reaches the steady-state solution, we set a very large final time and compute, at each time step, the difference on the velocity field between two subsequent time steps. If the difference is lower than a given threshold, we can conclude that the flow has reached a steady state and that there is no wetting failure.

In Figure 9.7, we show an example of a VOF simulation for a grid  $\Delta = 0.156 \mu\text{m}$ , corresponding to 64 grid spacings per slip length. The adaptive mesh refinement allows a good resolution of the interface at the triple point. In this particular case, for  $\text{Re} = 30$  and  $\text{Ca} = 2.6$ , a steady state solution is reached.



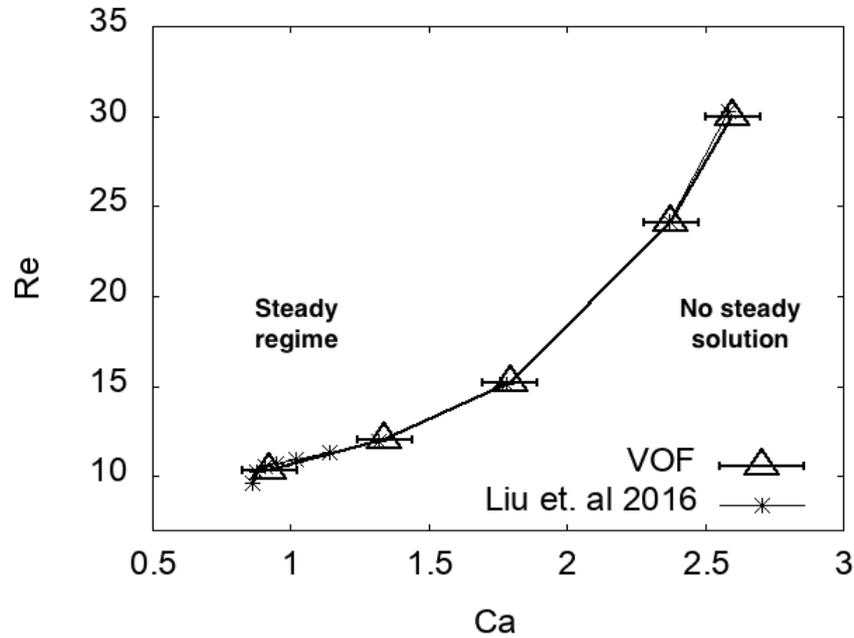
**Figure 9.7:** Example of resolution of the contact line for  $\lambda = 10 \mu\text{m}$  with 64 grid spacing per slip length. The color map corresponds to the x-component of the velocity field. Top: foot of the curtain. Bottom: from left to right, successive zooms on the contact line.



**Figure 9.8:** Steady state solutions for the three different flow configurations. From left to right: bead pulling ( $\text{Re} = 12$ ,  $\text{Ca} = 1.33$ ), beneath the liquid curtain ( $\text{Re} = 30$ ,  $\text{Ca} = 2.6$ ), heel formation ( $\text{Re} = 35$ ,  $\text{Ca} = 1.36$ ).

By varying the substrate velocity and the feed flow velocity, we are able to recover the same qualitative flow configurations: bead pulling, right beneath the liquid curtain and heel formation (Figure 9.8) and a similar coating window (Figure 9.9) as in [Liu *et al.* 2016].

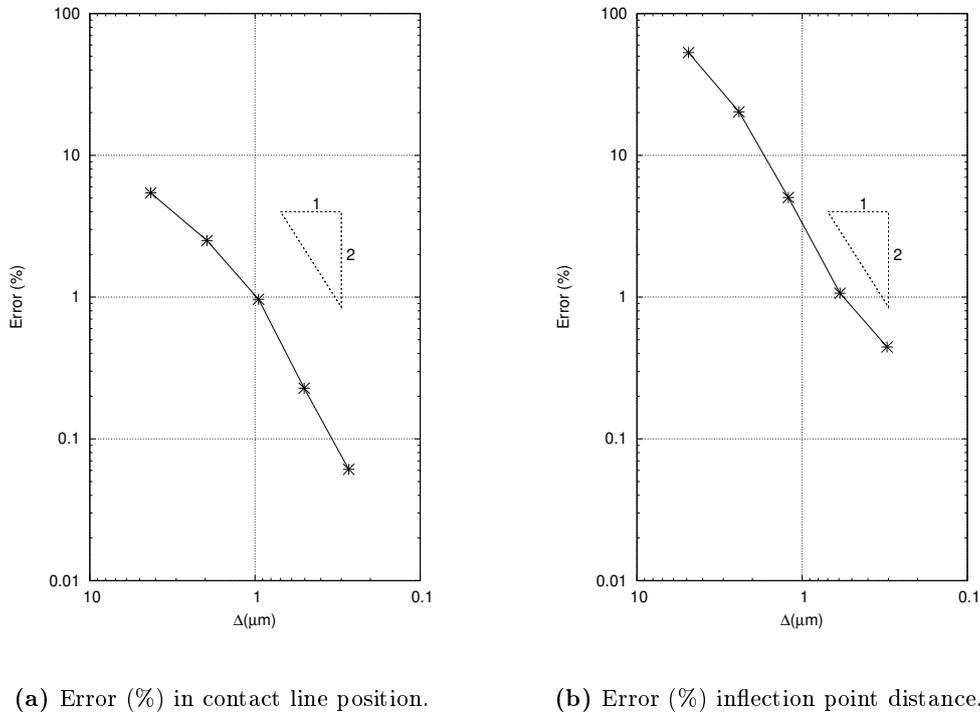
The hydrodynamic assist has the most impact (ie. the moving plate velocity  $U$  is maximum) when the contact line is beneath the liquid curtain. This configuration allows a stronger pressure due to the liquid inertia at the triple point preventing the formation of bubbles and therefore preventing the wetting failure from occurring. To determine the coating window, we look for the first unsteady solution while increasing the Ca number for a given Re number. The error bar relates to the difference in Ca values between the last steady solution and the first unsteady one.



**Figure 9.9:** Stability analysis of the reduced curtain coating system and comparison with computations of Liu [Liu *et al.* 2016].

A convergence study of the resolution of the interface is conducted for this configuration. As the maximal level of refinement is increased, the resolution of the interface at the contact line is improved. The microscopic contact angle tends to the prescribed one of  $90^\circ$  as the smallest cell size is decreased from  $10 \mu\text{m}$  to  $0.156 \mu\text{m}$ , corresponding to a number of grid points per slip length increasing from 1 to 64. We choose the 64 grid points per slip length solution as the reference solution for the contact line position. In Figure 9.10, we plotted the relative error of the contact line position and the relative error of the distance from the contact line position to the inflection point as a function of  $\Delta$  for the  $\text{Re} = 30$ ,  $\text{Ca} = 2.6$  case, which is close to the stability limit. The results obtained for both quantities considered

show a second-order convergence of the VOF method. Moreover, the distance from the contact line position to the inflection point of the reference solution is  $50 \mu\text{m}$ . This result compares favorably with the experimental techniques used to compute the contact angle [Blake *et al.* 1999, Blake & Shikhmurzaev 2002]. We have demonstrated that for a sufficient resolution of the interface, our VOF model with a Navier boundary condition on the moving substrate coupled to an imposed contact angle is able to reproduce the main stability results of the curtain coating configuration of Liu [Liu *et al.* 2016]. Moreover, this is done with far fewer points per slip length than in the finite element computations suggesting a higher accuracy and robustness of the VOF method with height functions.

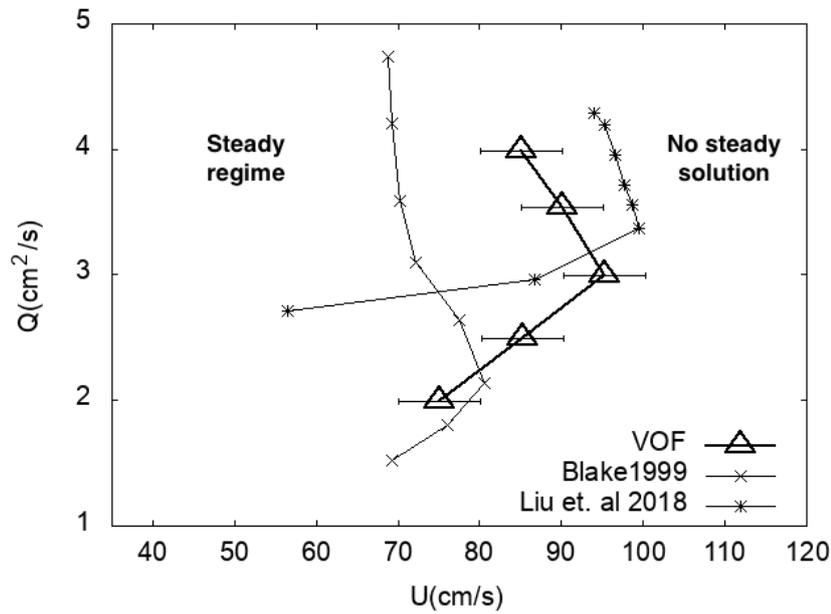


**Figure 9.10:** Convergence study of the reduced curtain coating system for  $\text{Re} = 30$ ,  $\text{Ca} = 2.6$  and  $\lambda = 10 \mu\text{m}$ . The reference solution is taken for 64 grid points per slip length.

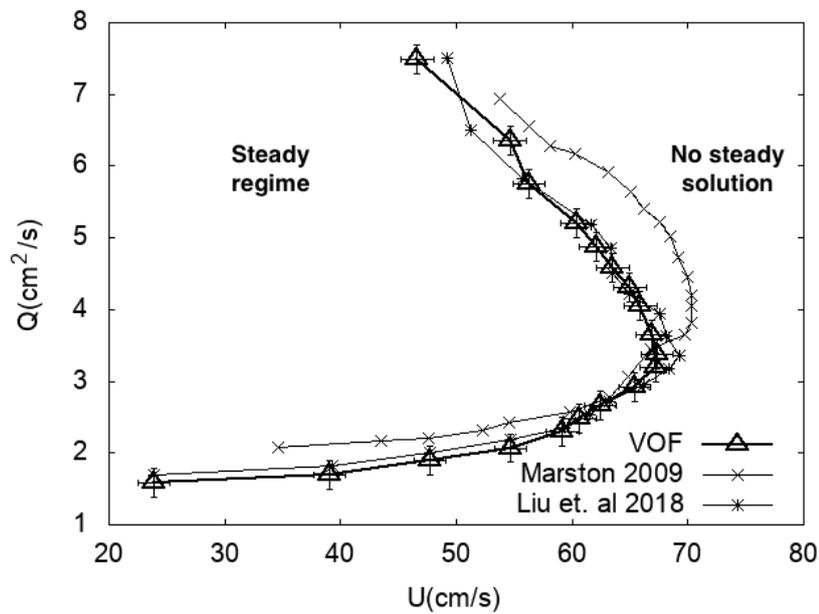
We now compare our model prediction with the two experimental observations of Blake [Blake *et al.* 1999] and Marston [Marston *et al.* 2009]. In each case, the values of experimental parameters of the system are taken as inputs for our simulations (curtain height, liquid viscosity, equilibrium surface tension, and imposed contact angle).

**Comparison with the Blake experiments** We choose the physical slip length used in [Liu *et al.* 2018] is  $\lambda = 5\text{nm}$ . In this case, the slip length is not well-resolved as  $\Delta = 230\text{nm}$  and there is an implicit numerical slip of  $\Delta/2$ . Therefore, the modification of the slip length does not affect the substrate velocity and induces no modification in the stability window in the numerics as the smallest  $\Delta$  attainable is larger than  $\lambda$ . The stability limit curve of the VOF model is computed by interpolating between a stable and unstable solution, showed by the error bars, for a fixed feed flow rate  $Q = Vd_c$  (Figure 9.11). The numerical results show a maximum flow feed velocity  $V$  of 3 cm/s for a plate velocity  $U$  of 90 cm/s whereas the experimental results from [Blake *et al.* 1999] give a maximum flow feed velocity of 2.15 cm/s for a plate velocity of 80.8 cm/s. The discrepancy between experimental observations and numerical results may be a consequence of the poor resolution of the smallest length scale in this case but also of the simplifications made at the boundary. The microscopic contact angle could depend on the parameters controlling the flow [Blake & Shikhmurzaev 2002, Shikhmurzaev 1994, Wilson *et al.* 2006, Eggers & Evans 2005]. A further regularization of the contact line singularity with a generalized Navier boundary condition [Qian *et al.* 2003a], where the contact angle is a dynamic one, might be needed. Even though the numerical results do not quantitatively match the experiments, we are still able to recover the shape of the stability-limit curve. The non-monotonic behavior of the system is still preserved.

**Comparison with the Marston experiments** In this configuration,  $\Delta = 230\text{nm}$ , so we are able to resolve the slip length determined in [Marston *et al.* 2009] of the order of hundreds of nanometers. By increasing  $\lambda$ , the substrate velocity increases and the stability window is shifted towards the left, meaning that for the maximum substrate velocity the solution becomes unstable. We found that the best slip value was the one used in [Liu *et al.* 2018],  $\lambda = 450\text{nm}$ . As we can see in Figure 9.12, the VOF model results are in good agreement with the experimental observations and almost perfectly match previous numerical computations. As noted in [Liu *et al.* 2018], the very large value of slip length found may come from the condition on the substrate. The substrate is pre-wetted in Marston experiments [Marston *et al.* 2009] whereas it is dry in our simulations.



**Figure 9.11:** Stability analysis and comparison with the experiments of Blake and computations of Liu. The model parameters are:  $\mu_l = 25$  mPa.s,  $\mu_g = 0.018$  mPa.s,  $\sigma = 64$  mN.m<sup>-1</sup>,  $h_c = 3$  cm,  $\theta_m = 67^\circ$  and  $\lambda = 5$  nm.



**Figure 9.12:** Stability analysis and comparison with the experiments of Marston and computations of Liu. The model parameters are:  $\mu_l = 117$  mPa.s,  $\mu_g = 0.018$  mPa.s,  $\sigma = 67$  mN.m<sup>-1</sup>,  $h_c = 2.6$  cm,  $\theta_m = 67^\circ$  and  $\lambda = 450$  nm.

### 9.3.2 Squalane drop spreading with dynamic contact angle

In this study, we use the model ‘Dyn3’ described in [Legendre & Maglio 2015]. The slip length  $\lambda$  is fixed to  $\Delta/2$  and  $l$  is a small length scale of the order of the nanometer. At each time step, the dynamic contact angle  $\theta_d$  is computed and imposed through the height functions. Equation 9.10 gives

$$\theta_d = g^{-1} \left( g(\theta_S) + \text{Ca}_{\text{CL}} \log \left( \frac{L}{l} \right) \right). \quad (9.13)$$

In practice,  $g$  and  $g^{-1}$  can be approximated by a fitting polynomial as described in [Dupont & Legendre 2010]

$$g(x) = \frac{1}{9} x^3 - 0.00183985 x^{4.5} + 1.845823 \times 10^{-6} x^{12.258487},$$

$$g^{-1}(x) = \frac{1}{4.33} x^{1/3} + 0.0727387 x - 0.0515388 x^2 + 0.00341336 x^3.$$

We compare our VOF dynamic angle model with experiments from Lavi and Marmur [Lavi & Marmur 2004]. They consider the spreading of a squalane drop on a solid substrate. The drop of volume  $V = 4\pi R_0/3$  is initialized on a wall with no initial velocity. The Ohnesorge number characterizing the flow is

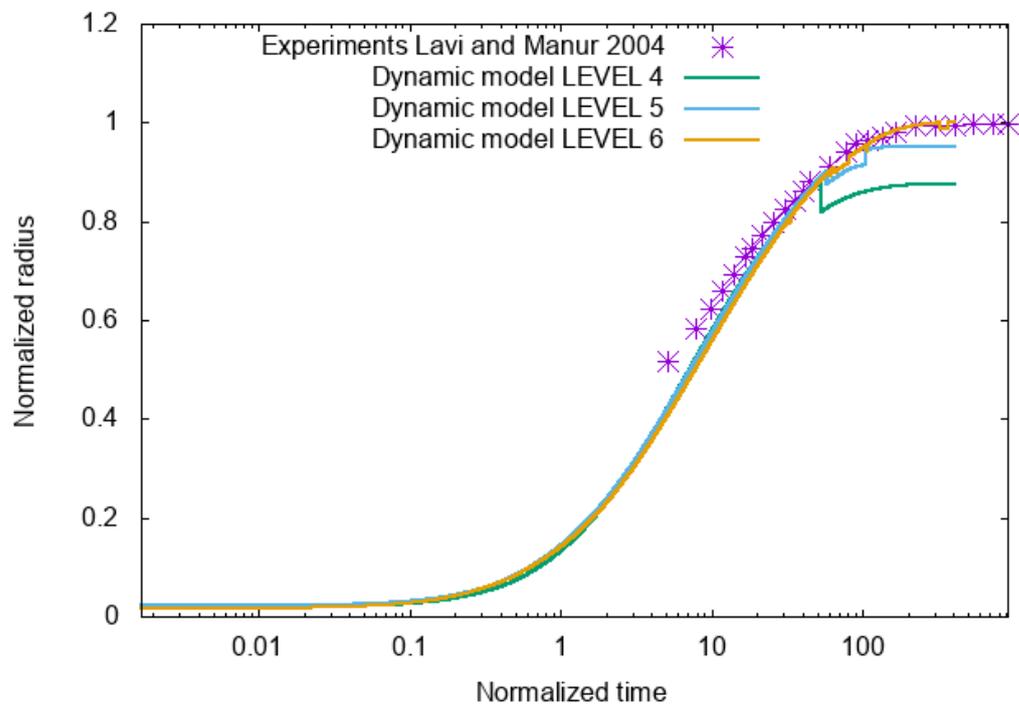
$$\text{Oh} = \frac{\mu l}{\rho_l \sigma R_0} = 0.21,$$

and static contact angle is  $\theta_S = 41.5^\circ$ .

We carry out simulations using the same nanoscopic length  $l = 10^{-9}\text{m}$  as in [Legendre & Maglio 2015]. In Figure 9.13, we present the comparison between our simulations and the experimental data for different grids. The level of refinement corresponds to the smallest grid size in the quadtree partitioning of the computational domain. Levels 4 to 6 corresponds to 8, 16 and 32 grid points per diameter. For low level of refinement, the contact line eventually gets pinned and the motion is stopped. At level 6, the final radius matches the experimental one. Moreover, in transient state, we can see a good agreement between both, validating our implementation of the dynamic angle model in VOF framework.

In the next chapter, we present the newly developed in models for moving contact lines:

1. The super-slip model with a further regularization of the velocity by introducing a second order slip length.
2. The generalized Navier boundary condition where the uncompensated Young’s stress is taken into account.
3. The toy model where we relate the mass diffusion existing in the phase-field to a simple dynamic angle relation.



**Figure 9.13:** Comparison between VOF simulations and the experiments of Lavi and Marmur for a spreading squalane drop. The time is normalized with the viscous time scale and the radius by the final radius  $r_f$ .

# Advances in models for contact lines

## Contents

<b>10.1 The super-slip model . . . . .</b>	<b>121</b>
10.1.1 The porous substrate analogy . . . . .	121
10.1.2 Application to a spreading drop . . . . .	122
<b>10.2 The generalized Navier boundary condition . . . . .</b>	<b>126</b>
10.2.1 The uncompensated Young stress . . . . .	126
10.2.2 Forced dewetting and film transition . . . . .	127
<b>10.3 The toy model: mimic Phase-Field at lower cost . . . . .</b>	<b>135</b>
10.3.1 Relating mass flux to curvature . . . . .	135
10.3.2 Phase-Field simulations of dynamic wetting by the Volume-of-Fluid method . . . . .	139
<b>10.4 Summary of models and applications . . . . .</b>	<b>143</b>

## 10.1 The super-slip model

### 10.1.1 The porous substrate analogy

As shown in Chapter 9, the Navier boundary condition –coupled to a constant or dynamic contact angle– is sufficient to regularize the solution at the contact line in some cases. The no-slip paradox is relaxed and the numerical slip in simulations can be used as a fitting parameter.

Nevertheless, Devauchelle, Josserand and Zaleski [Devauchelle *et al.* 2007] pointed out (in Appendix B) that the NBC fails to suppress all the singularities at the contact line. In particular, the capillary pressure diverges at the triple point. By studying the interfacial slip in porous media, they showed that from the Brinkman equations and the Darcy law in the case of a thin boundary layer, one can derive the following boundary condition for the tangential velocity at the wall

$$u_x - \lambda \frac{\partial u_x}{\partial y} - a \lambda^2 \frac{\partial^2 u_x}{\partial y^2} = U, \quad (10.1)$$

where  $u_x$  is the x-component of the velocity,  $\lambda$  the slip length,  $a$  a positive coefficient of order one and  $U$  the wall velocity. By introducing two separate slip lengths  $\lambda_1$

and  $\lambda_2$ , Equation 10.1 becomes

$$u_x - \lambda_1 \frac{\partial u_x}{\partial y} - \lambda_2 \frac{\partial^2 u_x}{\partial y^2} = U. \quad (10.2)$$

This formulation allows us to independently vary the first order-slip  $\lambda_1$  and second-order slip  $\lambda_2$  to investigate the effect of each term. The numerical implementation and validation of the super-slip model in the VOF framework are presented in Appendix E.

### 10.1.2 Application to a spreading drop

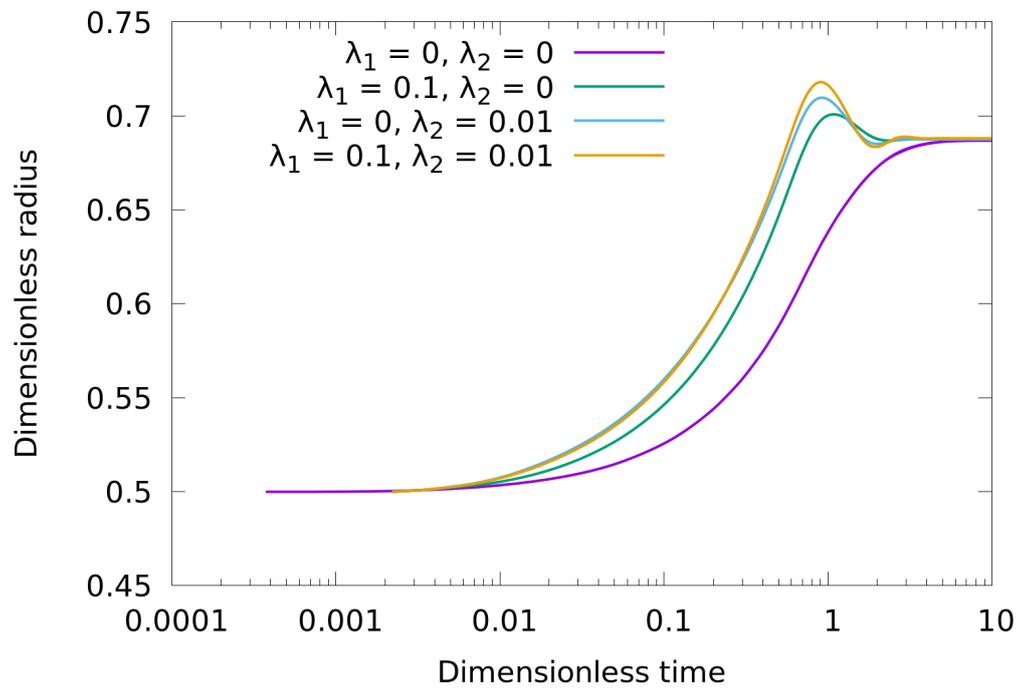
To study the effect of the super-slip model on the dynamic contact line, we consider a drop initialized with a  $\theta_0 = 90^\circ$  contact angle on a static substrate ( $U = 0$ ). We impose an constant angle  $\theta_e = 60^\circ$  that will determine the equilibrium shape. The effect of  $\lambda_1$  and  $\lambda_2$  will only appear in the transient state.

The viscosities and densities in both phases ( $\mu_l, \mu_g, \rho_l, \rho_g$ ), and the surface tension  $\sigma$  are equal and set to one. The initial radius is  $R_0 = 0.5$  and all the simulations are run until a final time  $t_f = 10$  with a fixed grid spacing  $\Delta = 0.0078125$  corresponding to 128 points per initial diameter. We consider four different cases

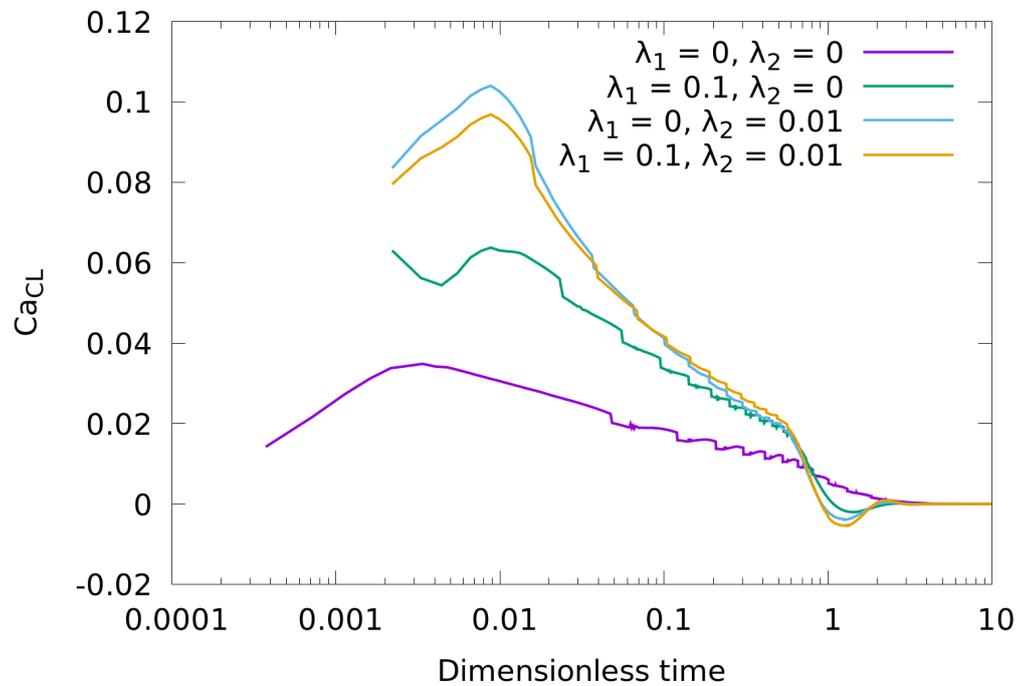
1.  $\lambda_1 = 0$  and  $\lambda_2 = 0$  corresponding to the homogeneous Dirichlet boundary condition  $u_x = 0$  (no-slip) at the wall.
2.  $\lambda_1 = 0.1$  and  $\lambda_2 = 0$  corresponding to a Navier boundary condition as described in Section 9.2.1.
3.  $\lambda_1 = 0$  and  $\lambda_2 = 0.01$  where only the second-order slip modifies the spreading motion.
4.  $\lambda_1 = 0.1$  and  $\lambda_2 = 0.01$  where  $\lambda_2 = \lambda_1^2$  as in Equation 10.1.

In the following figures, we look at the drop displacement, the contact line speed, the contact angle and the shear stress as a function of time for the four cases.

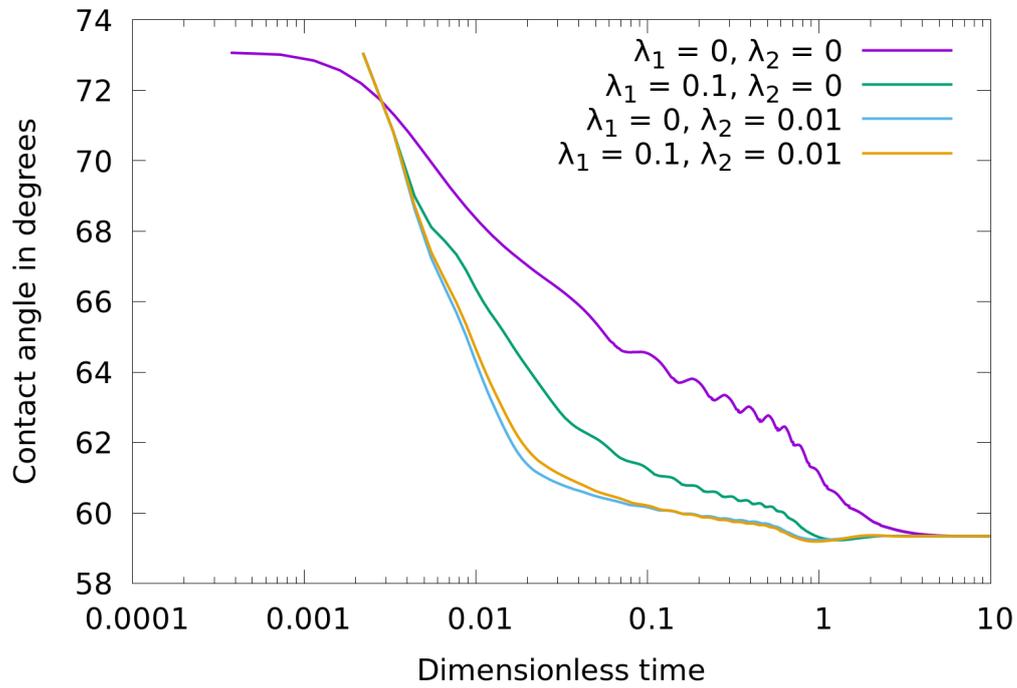
- In Figure 10.1, we plotted the radius as function of time. The first notable effect is the fact that cases 2,3 and 4 overshoot the final static solution. In those cases, the contact line position relaxes back to the equilibrium one whereas in case 1 (no-slip), it always remains below the latter. This is one of the natural characteristics of the slip models; allowing the contact line to slip over the substrate results in a faster motion that may pass over the equilibrium position. Interestingly, it appears that the second-order slip is the leading order term. The ‘bump’ is larger in case 3 (second order slip) than in case 2 (first order slip). Moreover, we can see that the effect of both slips is added as the bump is maximal in the fourth case. In all the cases the contact line eventually relaxes to its equilibrium position as expected.



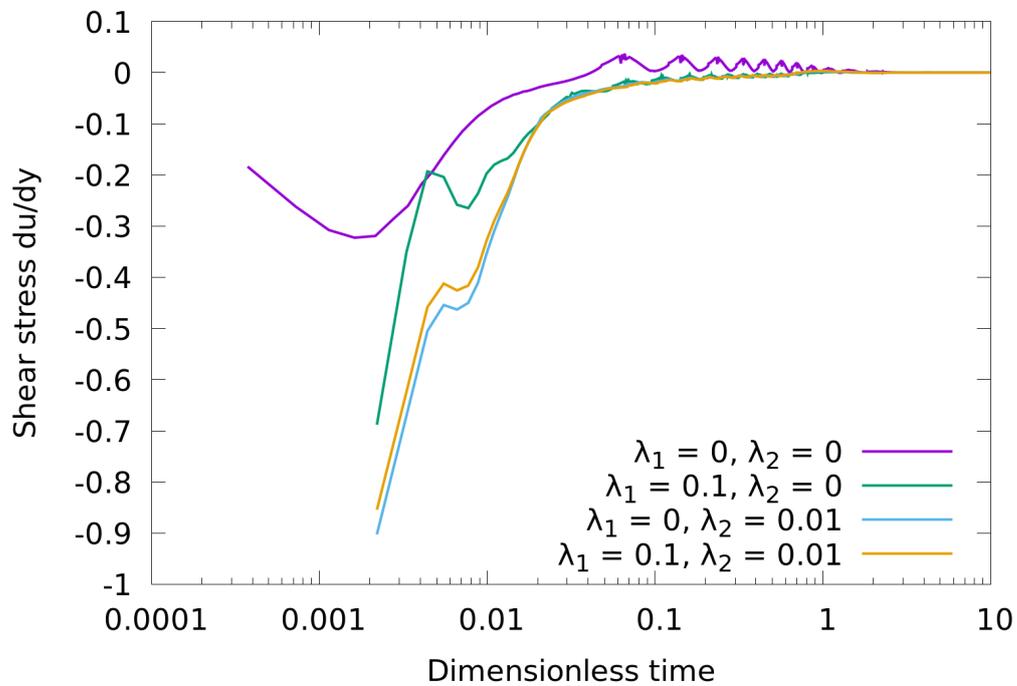
**Figure 10.1:** Dimensionless radius as a function of the dimensionless time in logarithmic scale for  $\lambda_1 = 0, 0.1$  and  $\lambda_2 = 0, 0.01$ .



**Figure 10.2:** Local capillary number depending on the contact line speed as a function of the dimensionless time in logarithmic scale for  $\lambda_1 = 0, 0.1$  and  $\lambda_2 = 0, 0.01$ .



**Figure 10.3:** Contact angle as a function of the dimensionless time in logarithmic scale for  $\lambda_1 = 0, 0.1$  and  $\lambda_2 = 0, 0.01$ .



**Figure 10.4:** Shear stress at the contact line as a function of the dimensionless time in logarithmic scale for  $\lambda_1 = 0, 0.1$  and  $\lambda_2 = 0, 0.01$ .

- Figure 10.2 shows the local capillary number (Equation 9.11) computed in the contact line cell as the drop is spreading. The contact line speed is initially zero, increases until a maximal value before tending to zero as the static shape is reached. In cases 2, 3 and 4 the speed becomes negative after overshooting the equilibrium position clearly indicating that the contact line motion is reversed to reach the equilibrium. In the first case, however,  $\text{Ca}_{\text{CL}}$  is always positive or equal to zero. The highest  $\text{Ca}_{\text{CL}}$  is attained in case 4, indicating again, the addition of the effects of  $\lambda_1$  and  $\lambda_2$ . In the no-slip case, we can observe oscillations of the speed as it is tending to zero. These oscillations are related to grid effects. As the contact line crosses a cell boundary, the newly computed volume fraction is very small. This induces a loss of accuracy in the computation of the surface tension term. The oscillations, although smaller, are also present in the slip case whereas they are almost entirely removed in cases 3 and 4 (where  $\lambda_2 \neq 0$ ). This last observation illustrates the regularizing effect of the second-order slip.
- In Figure 10.3, we show the variation of the contact angle as a function of time. The angle is computed at each time step in the contact line cell. The first noticeable thing to observe is the correlation between oscillations in contact line speed and contact angle. This is in agreement with the previous remarks on the loss of accuracy when the contact line crosses a cell boundary. Indeed, as detailed in Section 9.1, the contact angle directly influences the calculation of the curvature term therefore modifying the value of the velocity when solving the two-phase Navier-Stokes equations. We also note that the final contact angle is slightly different from the imposed equilibrium angle  $\theta_e = 60^\circ$ . There is a difference between the ‘imposed’ angle through height functions and the ‘extracted’ angle. This  $\Delta\theta_{\text{num}}$ , that exists in all of our contact line simulation, seems to depend on the flow characteristics and the grid resolution, and is still being investigated.
- Finally, in Figure 10.4, we look at the shear stress  $\partial u_x / \partial y$  at the contact line as a function of time. This last comparison between the cases is essential to understanding the regularizing effect of the second-order slip. In the no-slip simulation, the shear stress strongly oscillates and changes sign as the drop reaches the equilibrium shape. The change in sign indicates a strong modification of the flow around the contact line position. These oscillations – although with no change in sign – are still present in case 2 whereas in cases 3 and 4 ( $\lambda_2 \neq 0$ ), the shear stress is smooth out.

All these considerations demonstrate the regularizing effect of the second-order slip. Nonetheless, the super-slip model still needs to be tested on physical systems. The newly considered systems would need to allow a (very) high number of grid points per slip  $\lambda_1$  as  $\lambda_2 = \lambda_1^2$  and  $\lambda_1 \ll 1$ .

## 10.2 The generalized Navier boundary condition

### 10.2.1 The uncompensated Young stress

The main idea of the generalized Navier boundary condition (GNBC) is to add the uncompensated Young stress to the NBC in order to allow a deviation between the dynamic contact angle and the equilibrium one. This stress comes from the deviation of the fluid-fluid interface from the static configuration [Qian *et al.* 2003b, Qian *et al.* 2006]. Although the GNBC was formulated in a diffuse interface model, Fricke [Fricke 2020] showed that the model is well-posed in the sharp interface limit. The boundary condition for the tangential velocity is

$$u_x - \lambda \frac{\partial u_x}{\partial y} = U + f\left(\frac{x}{\varepsilon}\right) \frac{\sigma}{\mu_l} (\cos(\theta_d) - \cos(\theta_e)), \quad (10.3)$$

where  $\mu_l$  is the liquid viscosity,  $\sigma$  the surface tension,  $\theta_d$  the dynamic contact angle,  $\theta_e$  the prescribed contact angle and  $f\left(\frac{x}{\varepsilon}\right)$  a bell function defined as

$$f\left(\frac{x}{\varepsilon}\right) = \frac{\left(1 - \tanh^2\left(\frac{x}{\varepsilon}\right)\right)}{\varepsilon}. \quad (10.4)$$

Note that this bell function is different than the one defined in Section 9.2.1. The width  $\varepsilon$  is now related to the diffuse interface model.

The relation between the contact angle and the velocity of the contact line is now  $\text{Ca}_{\text{CL}} = f(\theta_d)$ . This means that the contact angle should be a solution of the flow instead of being prescribed, as it was the case until now. In the VOF method such ‘non-imposition’ of the angle is not possible at first glance. Indeed, even if we were not to impose an angle through the height functions, the homogeneous Neumann boundary condition on the volume fraction at the wall

$$\frac{\partial c}{\partial y}\Big|_{y=0} = 0$$

would still be required. This symmetry boundary condition imposes – by default – a  $90^\circ$  contact angle for the interface.

Fortunately, from kinematic relations on the dynamic contact line, the authors of [Fricke *et al.* 2018, Fricke *et al.* 2019] derived a relation between the numerical contact angle and an ‘apparent’ angle, a specific distance away from the wall. The numerical angle can be extrapolated using the following formula

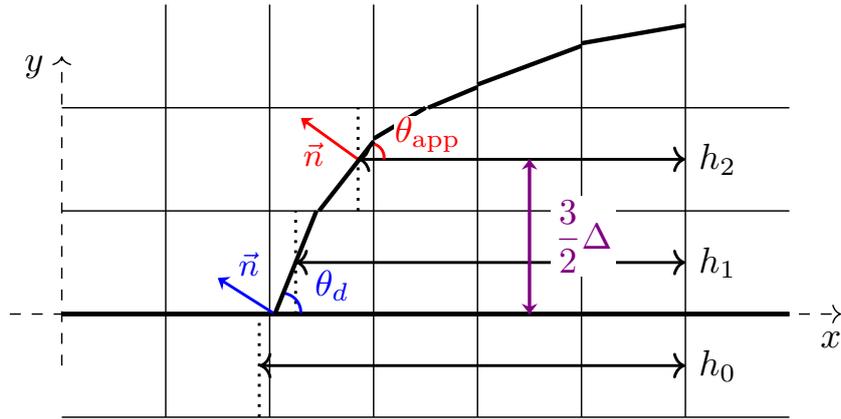
$$\theta_{\text{ext}} = \theta_{\text{app}} + \frac{3}{2} \Delta \frac{\kappa \sqrt{1 + h_y}}{\sin(\theta_{\text{app}})} \quad (10.5)$$

where  $\theta_{\text{ext}}$  is the extrapolated angle,  $\theta_{\text{app}}$  the apparent angle,  $\Delta$  the grid spacing,  $\kappa$  the curvature and  $h_y$  the first order derivative of the height function in the  $y$  direction (normal to the wall). A schematic of this extrapolation is presented in

Figure 10.5. Once the extrapolated angle is computed, we impose it through height functions, as if we would for a regular constant contact angle

$$\theta_d = \theta_{\text{ext}}$$

This method allows us to have ‘free’ dynamic contact angle in the sense that it is now a solution of the flow. A validation of the extrapolation method is provided in Appendix F.



**Figure 10.5:** Extrapolation of the contact angle using the apparent angle located  $3/2 \Delta$  away from the wall.

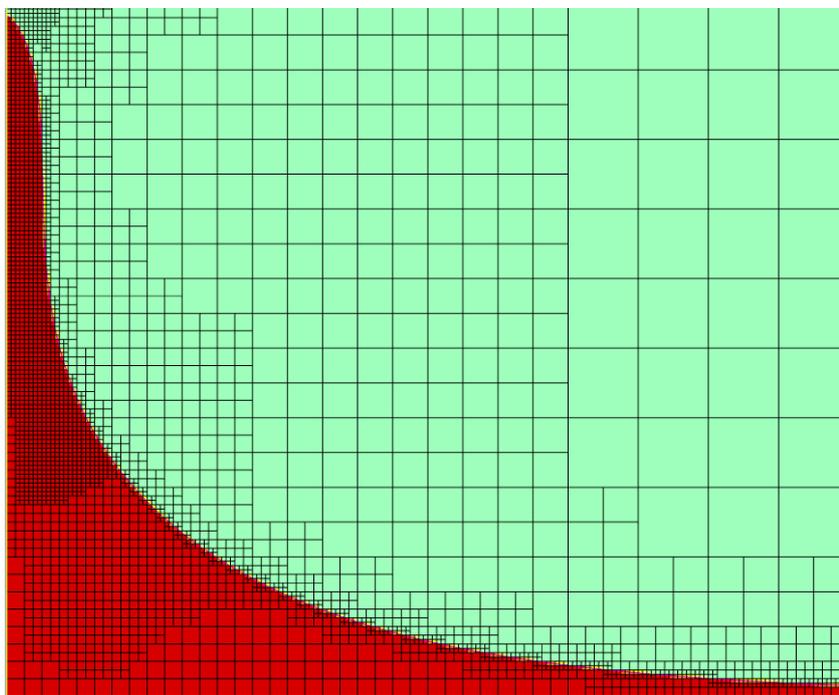
### 10.2.2 Forced dewetting and film transition

We study the effect of the GNBC on the onset of film transition in a forced dewetting setup [Afkhami *et al.* 2017]. We consider a solid substrate that is withdrawn from a viscous liquid pool. The interface may either sustain a stationary state meniscus, if below a critical capillary number,  $Ca_c$ , or continue to move up the substrate until depositing a thin film to arbitrary heights. An example of a steady state meniscus is shown in Figure 10.6.

In [Afkhami *et al.* 2017], the authors used a Dirichlet boundary condition for the tangential velocity and showed that there exists an implicit grid-dependent slip controlling the transition to a liquid film. In an attempt to demonstrate that the GNBC is grid-independent – provided that the width  $\varepsilon$  of the bell function is well resolved – we will carry out similar simulations for different widths, grids, capillary numbers and equilibrium angles.

The solid plate on the left is withdrawn at a speed  $U$  and the computational domain is a  $L \times L$  box where  $L = 9l_c$  with  $l_c = 1$  the capillary length. The liquid density is set to  $\rho_l = 1$  and the gas density to  $\rho_g = 0.2$ . The liquid and gas viscosities are equal and set to  $\mu_l = \mu_g = U$ , such that, with  $\sigma = 1$ , we have the relation

$$U = \sqrt{Ca}$$



**Figure 10.6:** Example of steady state meniscus for  $Ca = 0.03$ . The solid substrate on the left is moving up at a speed  $U$ . The color map shows the volume fractions with the red phase corresponding to the viscous liquid being withdrawn.

Moreover, the acceleration of gravity, set to  $\mathbf{g} = -1.25 \mathbf{e}_y$ , is opposing the upwards movement of the liquid. The initial liquid pool is flat and the interface is located at a height  $h_0 = 3.1$ . The simulations are carried out until a final time  $8 t_c$  with  $t_c = l_c/U$  the capillary time scale.

**Study of the effect of the width  $\varepsilon$**  As a first step, in order to investigate the effect of the width  $\varepsilon$  and the ratio  $\varepsilon/\Delta$ , we fix the capillary number to  $Ca = 0.1$  and an width  $\varepsilon_0 = 0.078125$  that will serve as a base. We run the simulation for 18 different cases, corresponding to 2 equilibrium angles, 3 widths and 3 grid spacings:

- $\theta_e = 80^\circ, 90^\circ$ ,
- $\varepsilon = \varepsilon_0, 2\varepsilon_0, 4\varepsilon_0$ ,
- $\Delta = \varepsilon_0/8, \varepsilon_0/4, \varepsilon_0/2$ .

The number of grid points per width  $\varepsilon/\Delta$  will therefore range from 2 to 8 in the case  $\varepsilon = \varepsilon_0$ , from 4 to 16 in the case  $\varepsilon = 2\varepsilon_0$  and from 8 to 32 in the case  $\varepsilon = 4\varepsilon_0$ . Additionally, the slip length is set  $\lambda = \varepsilon$  in order to have a unique microscopic length scale to control.

In the following results, we will investigate the contact line height and the contact angle as a function of time as well as the shear stress at final time.

- Figure 10.7 shows the normalized contact line height  $(h - h_0)/l_c$  as a function of time. In all of the cases, the contact line reaches a steady state position implying that  $Ca = 0.1 < Ca_c$ . In the  $\theta_e = 80^\circ$  cases the final height is higher than for the  $\theta_e = 90^\circ$  cases which is expected as a more hydrophobic substrate ( $\theta_e = 80^\circ$ ) will force the contact line towards the top of the domain. When decreasing  $\varepsilon$ , the effect of the uncompensated Young stress is further localized around the contact line and the slip length is decreased. The final position of the contact line is maximal for the smallest  $\varepsilon$ . Looking at the ratio  $\varepsilon/\Delta$ , we can see a clear convergence pattern. The height at highest resolution converges towards a certain value comprised between the 2 other resolutions. However, looking the  $\varepsilon = \varepsilon_0$  with 2 grid points per width cases (dotted black lines), we can observe a significant drop in the final position with respect to the 4 and 8 grid points per width cases. This clearly shows that 2 points per width is not enough to resolve the smallest length scale. This assertion is in accordance with the previous study on the Navier boundary condition (Section 9.3.1).
- Figure 10.8 shows the contact angle as a function of time in logarithmic scale. Note that this angle is not the one we impose through the extrapolation method (Equation 10.5). It is the extracted angle at the contact line cell (similarly to Figure 10.3). Consider first the  $\theta_e = 80^\circ$  cases. The angle starts from  $90^\circ$ , crosses the equilibrium angle and eventually settles towards a finite value. When the dynamic angle crosses this threshold, the sign of the uncompensated Young stress changes, as  $\cos(\theta_d) - \cos(\theta_e)$  is negative when  $\theta_d > \theta_e$  and positive when  $\theta_d < \theta_e$ . This transition appears relatively fast, meaning that the effect the stress pushes further the contact line in the direction of the moving plate. In the  $80^\circ$  cases, the angle is always below the threshold except for  $\varepsilon = \varepsilon_0$  with 2 grid points per width. In that specific case, one can observe significant oscillations of the angle that are in accordance with the previous remarks on the final height. Indeed, the final values of the angles are almost the same regardless of the number points per width meaning that these oscillations, only appearing in transient state, still influence the steady state solution. If we now consider the differences induced by modifying the absolute value of  $\varepsilon$ , we can observe that the final angle seems to converge towards a certain value as the effect of the Young stress is further localized.
- Finally, in Figure 10.9 we look at the shear stress at final time as a function of the  $y$  position of the wall. We can clearly see the effect of the reduction of the width. For large  $\varepsilon$ , the shear stress, while still reaching a maximum at the contact line position, is further spread out than for small  $\varepsilon$ . One interesting observation is the non-monotonic variation of the stress in the smallest  $\varepsilon$  cases. There is an inflexion appearing at certain distance away from the contact line. This effect still needs to be investigated.

From these results, we can draw two conclusions

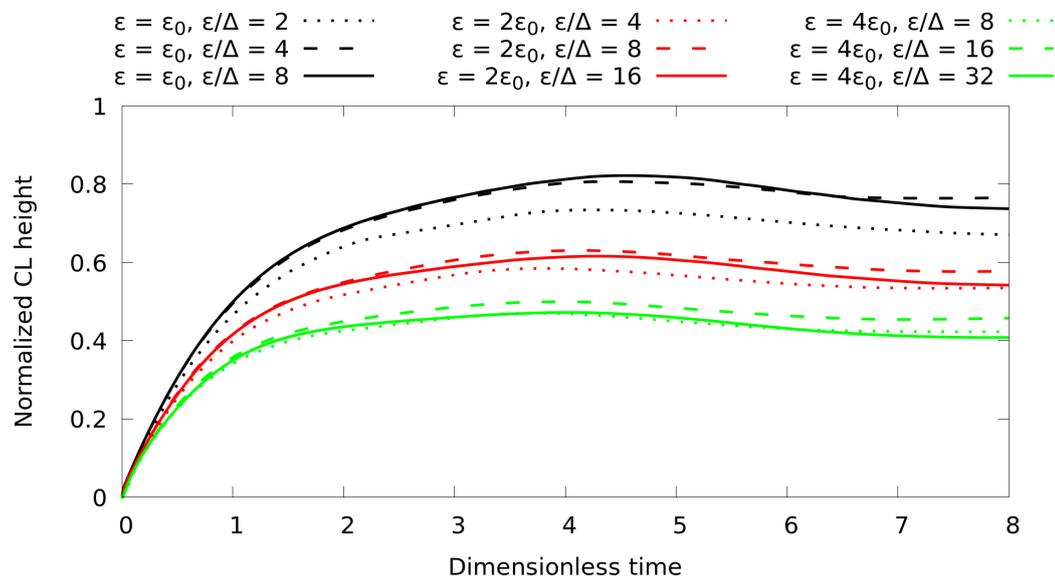
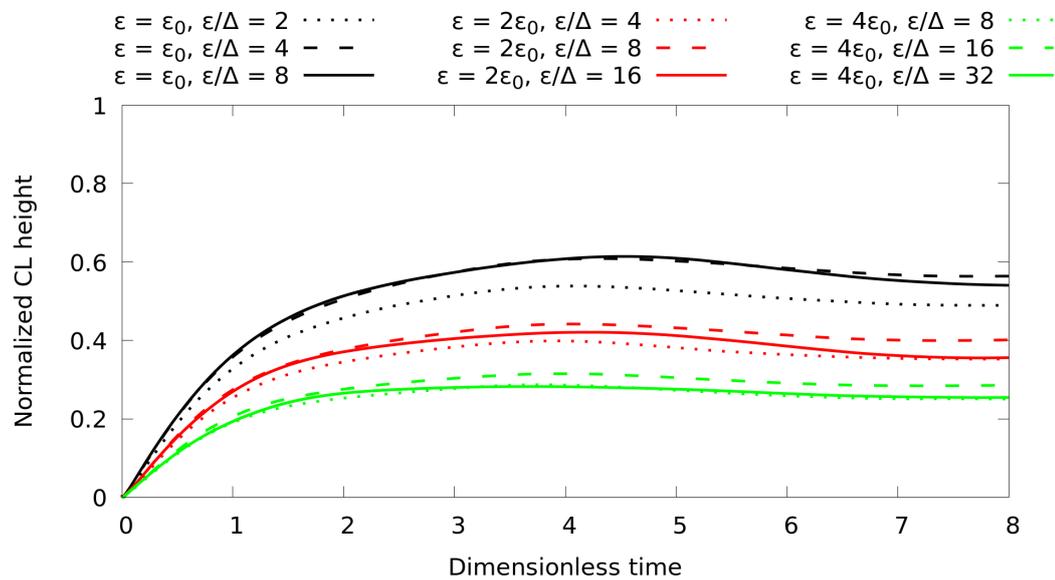
1. A small value of  $\varepsilon$  with respect to the macroscopic length scale – the capillary length in this case – appears as the natural way of treating the GNBC in the sharp interface limit. This is understandable as  $\varepsilon$  is related to the interface width in the diffuse interface model.
2.  $\varepsilon$  needs to be well-resolved in order to obtain converged results. As expected, the smallest length scale needs to be much larger than the grid spacing  $\Delta \ll \varepsilon$ .

Bearing these conclusions in mind, we can now compare our GNBC simulations to the no-slip simulations in order to investigate the grid-dependence – or independence – of the models.

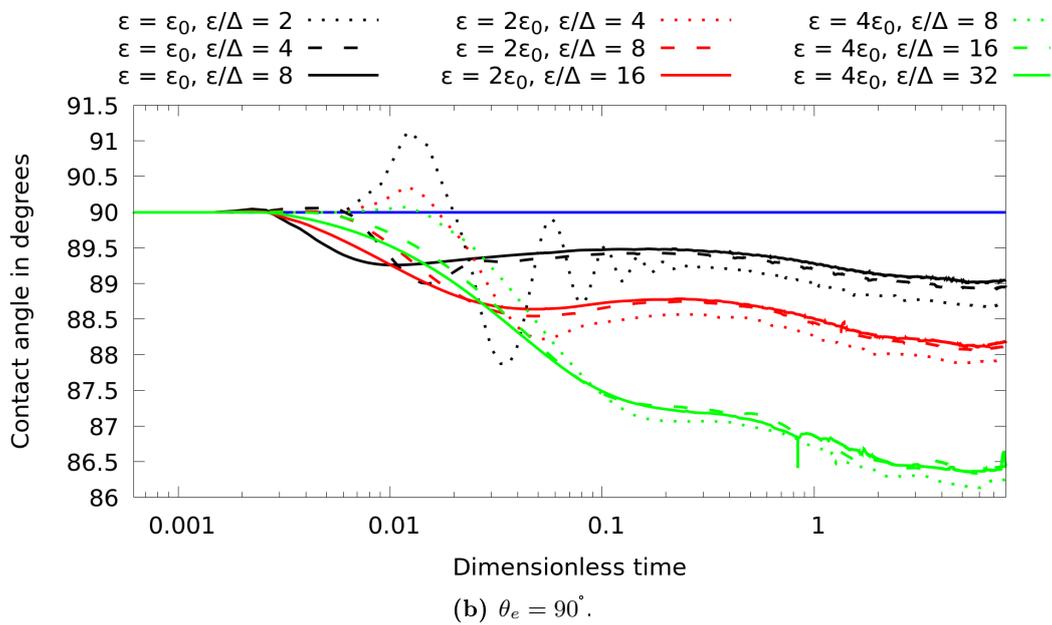
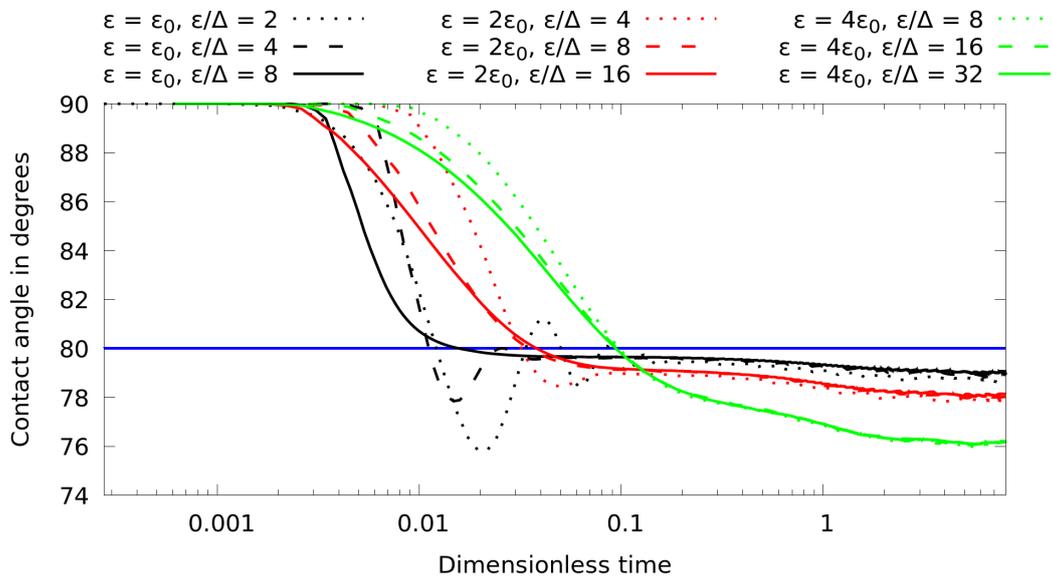
**Comparison between the no-slip and GNBC models** We consider the same setup, this time with  $\text{Ca} = 0.03$ , two different equilibrium angles  $\theta_e = 60^\circ, 90^\circ$  and a fixed width  $\varepsilon = 0.078125$ . The grids considered are  $\Delta = \varepsilon/4 = 0.01953125$  (coarse grid),  $\Delta = \varepsilon/8 = 0.009765625$  (medium grid) and  $\Delta = \varepsilon/16 = 0.004882813$  (fine grid). Figure 10.10a shows the results obtained with the no-slip model ( $u_x = U$ ). In the  $90^\circ$  cases, the liquid pool reaches steady state and the transition to a film does not occur. There is, as expected, a grid-dependence of the model, as the final height is increased when  $\Delta$  is decreased. This dependence is amplified in the  $60^\circ$  cases. The transition now occurs in the fine grid case whereas it does not in the other cases. As the grid is decreased, the numerical slip appearing in the no-slip model is decreased as well, therefore tending to a true ‘no-slip’ motion where the transition would exist for small values of  $\text{Ca}$ . These observations are in agreement with the ones of [Afkhani *et al.* 2017].

Figure 10.10b shows the results obtained with our GNBC model. It is clear, from the results that, in this particular case, that the GNBC model exhibits a true grid-independent behavior. In the  $\theta_e = 60^\circ$  case there is no transition to a film formation even with a fine grid.

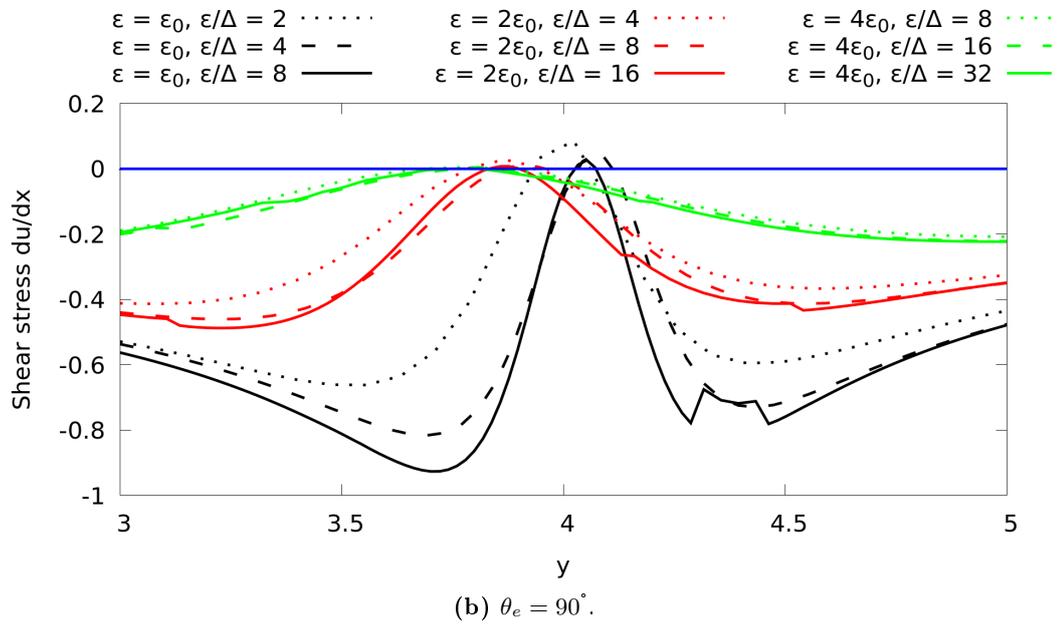
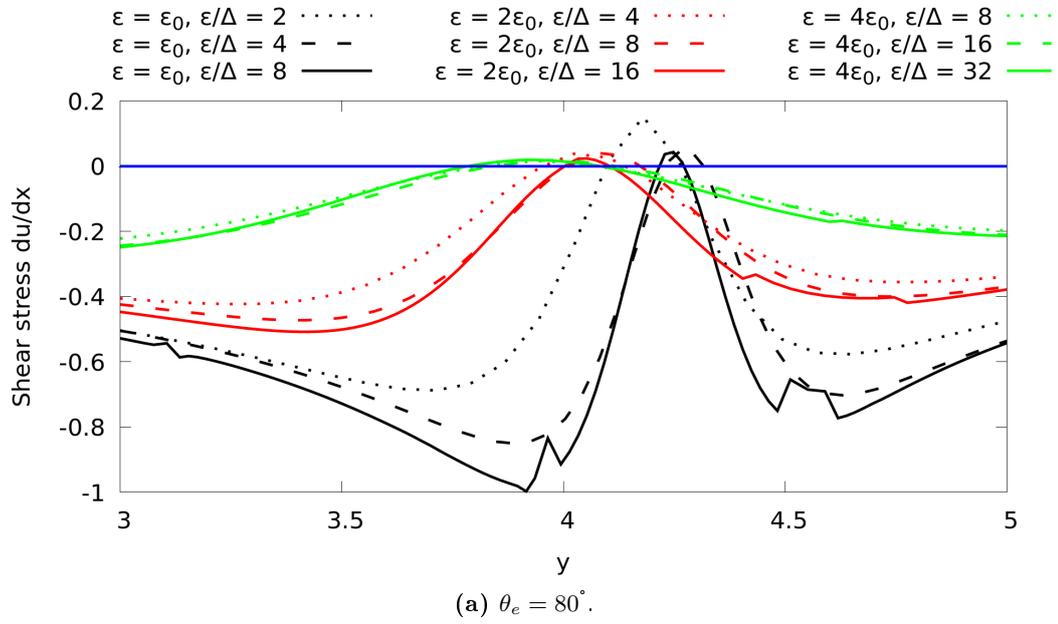
From this comparison and the previous study on the effect of the width, we can conclude that, in this particular case with  $\text{Ca} = 0.03$ , our GNBC model is an improvement with respect to the no-slip model. However, we need to further investigate this behavior by simulating a wide range of capillary numbers and looking at the effect of the uncompensated Young stress independently of the slip ( $\lambda = 0$ ).

(a)  $\theta_e = 80^\circ$ .(b)  $\theta_e = 90^\circ$ .

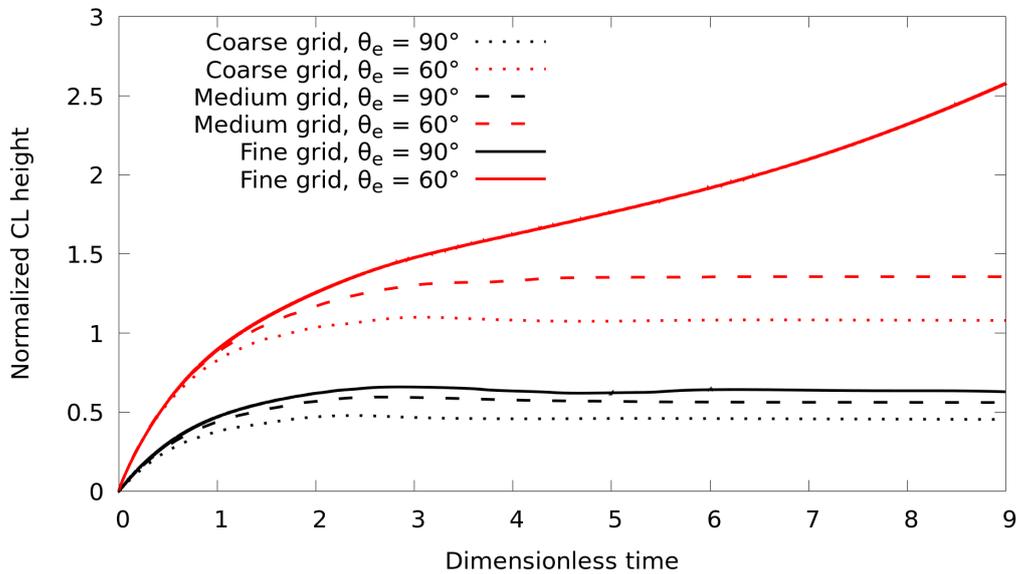
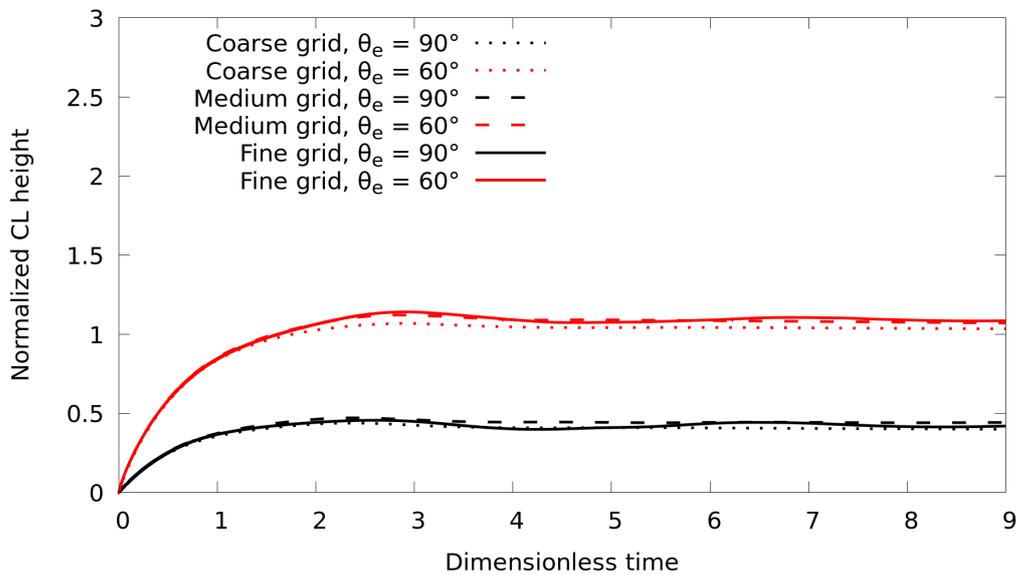
**Figure 10.7:** Normalized contact line position as a function of the dimensionless time for  $Ca = 0.1$  and  $\theta_e = 80^\circ, 90^\circ$ .



**Figure 10.8:** Contact angle as a function of the dimensionless time in logarithmic scale for  $Ca = 0.1$  and  $\theta_e = 80^\circ, 90^\circ$ . The blue lines correspond to the equilibrium angles.



**Figure 10.9:** Shear stress at final time as a function of  $x$  for  $Ca = 0.1$  and  $\theta_e = 80^\circ, 90^\circ$ . The blue line indicates that  $\partial u_y/\partial x = 0$ .

(a) No-slip boundary condition with a constant contact angle  $\theta_e$ .

(b) Generalized Navier boundary condition.

**Figure 10.10:** Comparison between the no-slip model and the GNBC in the withdrawing plate setup. The capillary number is  $Ca = 0.03$  and the equilibrium angles  $\theta_e = 60^\circ, 90^\circ$ . The coarse, medium and fine grids correspond to grid spacings  $\Delta/l_c = 0.01953125, 0.009765625, 0.004882813$  respectively.

## 10.3 The toy model: mimic Phase-Field at lower cost

### 10.3.1 Relating mass flux to curvature

Practical simulation of real dynamic wetting flows are still challenging [Huh & Scriven 1971, Bonn 2009, Snoeijer & Andreotti 2013]. In the Phase-Field (PF) method, the Cahn-Hilliard equations are formulated from the thermodynamics of an immiscible two-component mixture and a phase function is used to represent the moving interface. The velocity field can satisfy a no-slip condition at the contact line, and the contact line moves due the diffusive mass transport that is present in a two-component system [Yue & Feng 2011]. While the PF method has the advantage of needing less empirical fitting than the VOF method, it is considerably more expensive.

In this study, we show how to obtain results that are essentially equivalent to results from the PF method, by doing much cheaper VOF simulations. We introduce a computational boundary at a specified short distance from the wall in the VOF simulation. There we apply a NBC for tangential velocity, and an dynamic contact angle relation that accounts for the interface curvature at the wall. The basis for the model is a theoretical relation between total diffusive mass transport across the interface and curvature at the wall.

We compare full phase field simulations with VOF simulations using these boundary conditions. The input parameters for the VOF simulations are the same as for the full PF solution and the VOF solutions reproduce the PF solutions, at a much lower cost.

We start by taking the dimensionless incompressible two-phase Navier-Stokes equations in the Phase-Field approximation (defined in Chapter 7). As reference quantities we choose  $U$ ,  $L$ ,  $L/U$  and  $\mu_l U/L$  for velocity, length, time and pressure, with  $L$  denoting a macroscopic length, for example the droplet radius.

The dimensionless numbers that appear are the Reynolds number

$$\text{Re} = \frac{\rho_l U L}{\mu_l},$$

the Peclet number

$$\text{Pe} = \frac{U L}{\alpha},$$

where  $\alpha = M\sigma/\varepsilon$  is the mass diffusion coefficient. The Cahn number – or dimensionless interface width – is defined as

$$\text{Cn} = \frac{\varepsilon}{L}.$$

$\text{Ca}_F = \mu_F U/\sigma$  is the line friction capillary number, based on the line friction parameter  $\mu_F$ . In the rest of this study, we consider the line friction coefficient to be zero ( $\mu_F = 0$ ), in order to enforce the dynamic contact angle to the equilibrium one.

The dimensionless equations therefore read

$$\rho_l \left( \frac{\partial u}{\partial t} + u \cdot \nabla u \right) = -\frac{1}{\text{Re}} \nabla P + \frac{1}{\text{Re}} \nabla^2 u + \frac{1}{\text{Re} \text{Cn} \text{Ca}} \phi \nabla C, \quad (10.6)$$

$$\frac{\partial C}{\partial t} + u \cdot \nabla C = \frac{1}{\text{Pe}} \nabla^2 \phi, \quad (10.7)$$

with

$$\phi = -\text{Cn}^2 \nabla^2 C + \Psi'(C), \quad (10.8)$$

and

$$\Psi'(C) = (C^2 - 1)C.$$

The wetting boundary condition, with  $\mu_F = 0$ , is now

$$\frac{3}{2\sqrt{2}} \text{Cn} \nabla C \cdot n = \cos(\theta_e) g'(C),$$

with

$$g(C) = \frac{2 + 3C - C^3}{4}.$$

Assume a length scale  $\delta$  for diffusion, which is also coupled to the flow field. Assume a viscous balance in the momentum equation, and that the velocity is properly scaled (dimensionless  $|u| \sim 1$ ). Estimates of the relevant terms Equations 10.6 to 10.8 give

$$\frac{1}{\text{Re}} \frac{1}{\delta^2} \sim \frac{1}{\text{Ca} \text{Cn} \text{Re}} \frac{\phi}{\delta},$$

and

$$\frac{1}{\delta} \sim \frac{1}{\text{Pe}} \frac{\phi}{\delta^2}.$$

Solving for  $\delta$  yields

$$\delta = \sqrt{\frac{\text{Ca} \text{Cn}}{\text{Pe}}}, \quad (10.9)$$

which agrees with the scaling of Yue-Zhou-Feng [Yue *et al.* 2007]. We now derive expression for chemical potential on a curved interface.

The first step is to calculate the value of the chemical potential on a curved interface. For a droplet,  $C$  approaches +1 inside the droplet, and  $-1$  outside. Assuming a planar geometry and a circular interface of radius  $R$ , the chemical potential as defined in Equation 10.8 can be written as

$$\phi = -\text{Cn}^2 \left( \frac{\partial^2 C}{\partial r^2} + \frac{1}{r} \frac{\partial C}{\partial r} \right) + \Psi'(C). \quad (10.10)$$

Since the dimensionless interface width is  $\text{Cn} \ll 1$ , the solution to this equation has a boundary layer character, and we introduce a new radial coordinate  $p$  such that

$r = R + \text{Cn} \cdot p$  and expand Equation 10.10 at zero and first order

$$C = C_0 + \text{Cn} \cdot C_1 + O(\text{Cn}^2),$$

$$\phi = \phi_0 + \text{Cn} \cdot \phi_1 + O(\text{Cn}^2).$$

The problems for the zeroth and first order are now

$$\phi_0 = -\frac{\partial^2 C_0}{\partial p^2} + \Psi'(C_0), \quad (10.11)$$

and

$$\phi_1 = -\left(\frac{\partial^2 C_1}{\partial p^2} - \frac{1}{R} \frac{\partial C_0}{\partial p}\right) + C_1 \Psi''(C_0). \quad (10.12)$$

The zeroth order equation 10.11 gives the equilibrium solution as  $C_0 = -\tanh(p/\sqrt{2})$ , with  $\phi_0 = 0$ .

Multiplying the first order equation 10.12 by  $\partial C_0/\partial p$  and partially integrating, a solvability condition is obtained for the first-order value of the chemical potential

$$\int \phi_1 C_{0,p} dp = -\int (C_{0,p} C_{1,pp} + C_1 (\Psi'(C_0))_{,p}) dp - \frac{1}{R} \int (C_{0,p})^2 dp, \quad (10.13)$$

where  $,p$  denotes derivative with respect to  $p$ . The chemical potential is continuous at the interface, so the first order chemical potential  $\phi_1$  is a constant in equation 10.12. By partially integrating the first term on the right hand side that is seen to be zero. Using that

$$\int \phi_1 C_{0,p} dp = -2,$$

and

$$\int (C_{0,p})^2 dp = \frac{2\sqrt{2}}{3},$$

the result gives the value of the chemical potential at the interface

$$\phi = \text{Cn} \phi_1 = \frac{1}{2} \frac{2\sqrt{2}}{3} \frac{\text{Cn}}{R}. \quad (10.14)$$

We will now apply the same perturbation approximation to Equation 10.7. Again, the zeroth order gives the equilibrium concentration profile, but the first order becomes

$$-u \cdot \text{Pe} \cdot C_{0,p} = (-C_{1,pp} + C_1 \Psi''(C_0))_{,pp} - \frac{1}{R} C_{0,ppp}. \quad (10.15)$$

In order to derive a relation between the mass flux on the left hand side and the curvature, we multiply this equation by

$$F = \int_0^p C_0 dp. \quad (10.16)$$

By successive partial integrations, the first term in the right hand side is shown to

vanish, and the end result is

$$\int u \cdot \text{Pe} \cdot F F_{pp} dp = \frac{1}{R} \int F F_{pppp} dp, \quad (10.17)$$

with

$$\int F F_{pp} dp = 2\sqrt{2}(1 - \ln 2) = 0.8679,$$

and

$$\int F F_{pppp} dp = \frac{2\sqrt{2}}{3} = 0.9428.$$

In Equation 10.17, a factor  $\delta$ , the YZF distance to the stagnation point, is introduced on the left hand side to account for an integration in the vertical direction. The velocity profile on the left hand side goes to zero beyond  $\delta$ , but the right hand side comes from the diffusion field around the contact line which can have contributions from further away.

We now consider the Navier-Stokes equations (Equation 10.6) to couple this to the velocity field. Assuming a balance between the viscous term and the capillary term, we estimate the relevant terms in the Navier-Stokes equation as

$$\frac{1}{\text{Re}} \frac{a u}{\delta} \sim \frac{1}{R} \frac{2\sqrt{2}}{3} \frac{1}{\text{Ca Cn Re}}.$$

Here a dimensionless parameter  $a$  is introduced to account for the viscous stress  $\tau$  as

$$\tau = \frac{a u}{\delta}.$$

Putting everything together yields the following relations: the chemical potential  $\phi$  is related to the radius of curvature  $R$  at equilibrium by

$$\phi = \frac{1}{2} \frac{2\sqrt{2}}{3} \frac{\text{Cn}}{R}, \quad (10.18)$$

and the local curvature is a function of the flow parameters

$$R = \frac{2\sqrt{2}}{3} \frac{1}{\sqrt{0.8679}} \sqrt{\frac{\text{Cn}}{\text{Pe Ca } a}}. \quad (10.19)$$

Translating these equations in terms of boundary condition for the dynamic apparent angle  $\theta_{\text{app}}$ , yields

$$\cos(\theta_{\text{app}}) = \cos(\theta_e) + \frac{3}{2\sqrt{2}} \text{Ca}_{\text{CL}} a. \quad (10.20)$$

This apparent angle relation exists at the stagnation point – a distance  $+\delta$  away from the wall – where  $\delta$  is

$$\delta = \frac{1}{\sqrt{0.8679}} \sqrt{\frac{\text{Ca Cn } a}{\text{Pe}}}.$$

### 10.3.2 Phase-Field simulations of dynamic wetting by the Volume-of-Fluid method

We consider a drop spreading over a no-slip motionless substrate – located at  $y = 0$  – where the equilibrium angle  $\theta_e$  governs the final shape of the drop. Using the relations described above, we compare full PF simulations – in the sense that the full domain is solved – with VOF simulations using the toy model approximation. In the toy model, the bottom boundary is now a computational boundary that is located a distance  $+\delta$  away from the wall. The domain comprised between the wall  $y = 0$  and the computational boundary  $y = \delta$  is not simulated. There, we apply the dynamic angle relation (Equation 10.20), where the apparent angle  $\theta_{\text{app}}$  is now a dynamic contact angle  $\theta_d$

$$\theta_d = \cos^{-1} \left( \cos \theta_e + \frac{3}{2\sqrt{2}} \text{Ca}_{\text{CL}} a \right), \quad (10.21)$$

with  $\text{Ca}_{\text{CL}}$  the contact line capillary number. Moreover, in order to simulate a true no-slip condition at the wall, we use  $\delta$  as the slip length for the tangential velocity (again located at the computational boundary)

$$u_x|_{y=\delta} - \delta \frac{\partial u_x}{\partial y}|_{y=\delta} = 0. \quad (10.22)$$

When considering a spreading drop with this model, we need to relax the impermeability condition in the NBC by allowing a mass flux through the computational boundary. This sinking velocity  $v_{\text{sink}}$  is related to the amount of mass that we lose – in the case of  $\theta_e < \theta_0$  – or gain – in the case of  $\theta_e > \theta_0$ . The boundary condition for the normal component of the velocity at the wall is now

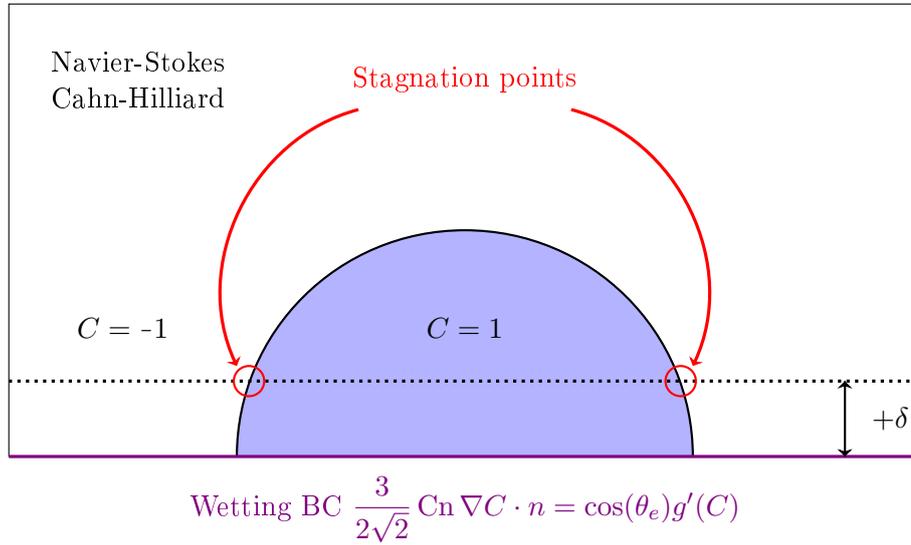
$$u_y|_{y=\delta} = v_{\text{sink}}. \quad (10.23)$$

A validation of the numerical implementation of  $v_{\text{sink}}$  is provided in Appendix G. Figure 10.11 shows the computational domains of both PF and VOF setups with the respective boundary conditions.

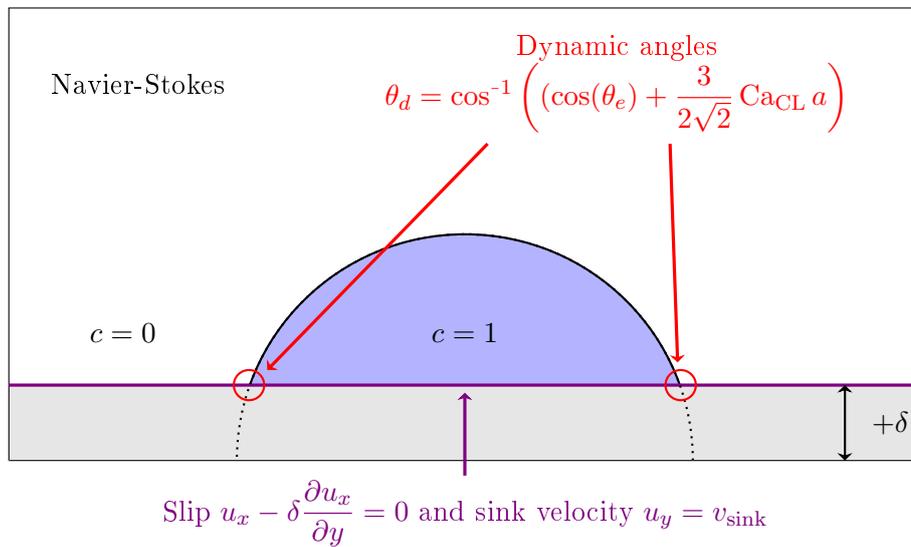
The initial radius of the drop is  $R_0 = 0.5$  with an initial contact angle  $\theta_0 = 90^\circ$ . The equilibrium angle is set to  $\theta_e = 70^\circ$ . The viscosities and densities ratio between both phases are

$$\bar{\mu} = \frac{\mu_g}{\mu_l} = 10^{-2},$$

$$\bar{\rho} = \frac{\rho_g}{\rho_l} = 10^{-2}.$$



(a) Full PF setup.



(b) Toy model VOF.

**Figure 10.11:** Schematic of the full PF and the toy model VOF computational domains with their respective boundary conditions.

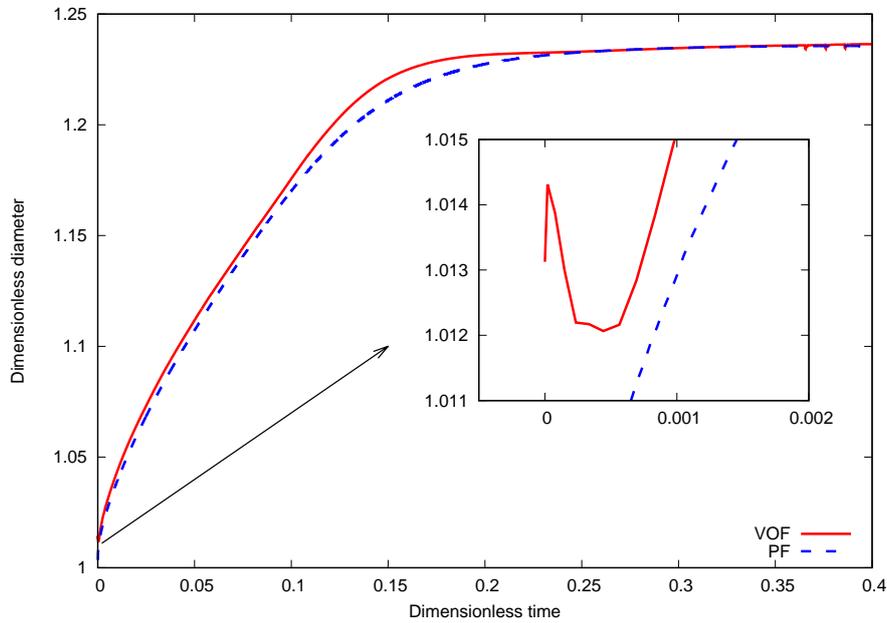
The dimensionless parameters governing the flow are set to

$$\begin{aligned} \text{Ca} &= 0.0212, \\ \text{Re} &= 3.978, \\ \text{Cn} &= 0.01, \\ \text{Pe} &= 1. \end{aligned}$$

Note that the choice of Peclet number agrees with the sharp interface limit approximation of Yue and Feng [Yue & Feng 2011]

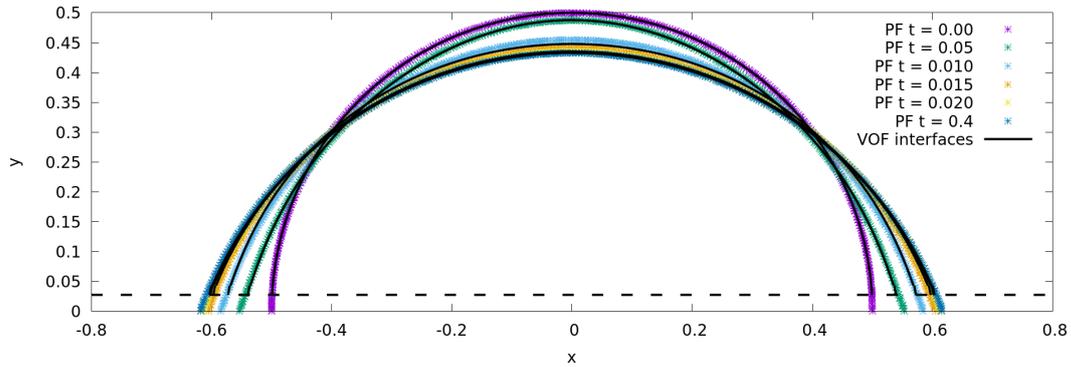
$$\text{Pe} = 1 < 16 \sqrt{\mu} \frac{\text{Ca}}{\text{Cn}} = 3.392.$$

The geometrical factor accounting for the viscous flow is set to  $a = 3$ . This value was found by fitting the radius of curvature and the right hand side of Equation 10.19. The PF simulations are carried out using FreeFEM-NS-CH, a script compilation for FreeFEM [Hecht 2012] to solve the coupled system of Navier-Stokes Cahn-Hilliard equations developed, by U. Lācis and S. Bagheri. The grid size used in the PF simulation is  $\Delta_{\text{PF}} = 0.0625 \varepsilon$  and the one in the VOF simulation is  $\Delta_{\text{VOF}} = 0.577 \delta$ . The results obtained are summarized in the following figures.



**Figure 10.12:** Dimensionless diameter as a function of time for the PF and VOF simulations. The inset shows the spurious initial oscillation of the displacement in the VOF simulation.

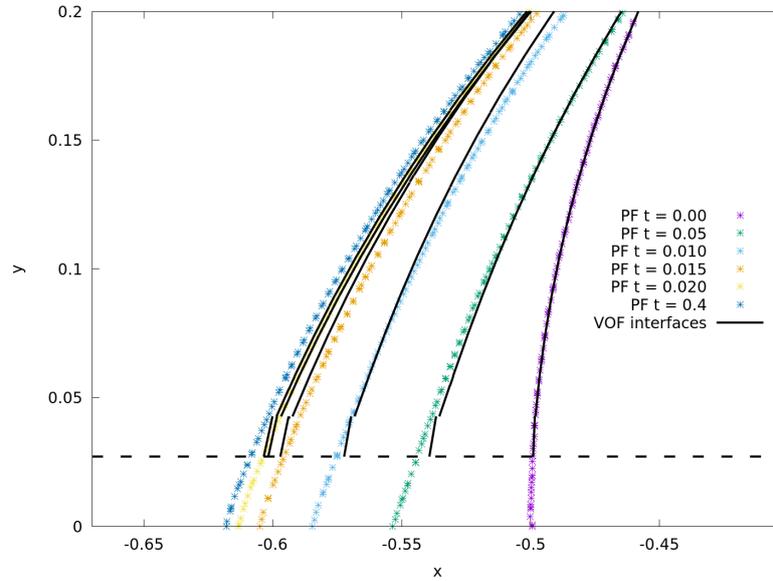
Figure 10.12 shows the diameter of the drop as a function of time for the PF and the VOF simulations. In order to compare the displacement, the position of the contact line in the VOF setup is projected back to the wall  $y = 0$  using the extracted apparent angle located at  $y = \delta + \Delta$ , in order to avoid the difference in imposed-extracted angles  $\Delta\theta_{\text{num}}$  at  $y = \delta$ , as detailed in Section 10.1.2. From the comparison between both models, we can observe that the displacement curves superimpose almost perfectly. The final equilibrium position of the contact line is the same in both cases, implying that the sinking velocity  $v_{\text{sink}}$  allowed us to accurately represent the mass flux through the computational boundary. In transient state, however, we see a slight difference between times 0.1 and 0.3 where the VOF simulation overshoots the solution. This behavior is similar to ones observed previously in the super-slip model and is related to the effect of the slip. The inset of Figure 10.12 shows the displacement zoomed-in at initial time. We can observe a spurious initial oscillation in the VOF model that is damped in a few iterations. This oscillation comes from the initial condition on the dynamic angle  $\theta_d$  that is too strong.



**Figure 10.13:** PF and VOF interfaces at times  $t = 0, 0.05, 0.1, 0.15, 0.2$  and  $0.4$ . The dotted black line corresponds to the location of the computational boundary in VOF toy model.

Figure 10.13 shows the interfaces for both models at times 0, 0.05, 0.1, 0.15, 0.2 and 0.4. The dotted black line corresponds to the location of the computational boundary. In the VOF simulation, the domain is cut at  $y = \delta$  and anything below it is not simulated. The superposition of the PF and VOF interfaces show good agreement. In Figure 10.14, we show the same interfaces zoomed at the contact line. The first observation is the appearance of a discontinuity in the last VOF fragment. Again, this showcases the numerical difference  $\Delta\theta_{\text{num}}$ , where the actual angle is always different from the one imposed through height functions the ghost layer (Section 9.3.1). Nevertheless, the VOF interfaces show a relatively good agreement with respect to the PF ones.

As a next step, one should carry out PF simulations for different Peclet numbers and compare them the VOF toy model to observe if a change in mass diffusion coefficient can be reproduced through our newly developed simple Navier boundary condition and dynamic contact angle relation.



**Figure 10.14:** Zoom around the contact line on the PF and VOF interfaces at times  $t = 0, 0.05, 0.1, 0.15, 0.2$  and  $0.4$ . The dotted black line corresponds to the location of the computational boundary in VOF toy model.

## 10.4 Summary of models and applications

We now summarize the implemented models and their applications. The different models can, first, be categorized by their type of boundary condition for the tangential velocity, namely:

1. The slip model or Navier boundary condition (NBC), Equation 9.7 in Section 9.2.1.
2. The local slip model, Equation 9.8 in Section 9.2.1.
3. The super-slip model, Equation 10.2 in Section 10.1.1.
4. The generalized Navier boundary condition (GNBC), Equation 10.3 in Section 10.2.1.

The second categorization corresponds to the numerical treatment of the contact angle:

1. A constant contact angle, Section 9.1. In this case the angle imposed through height functions is arbitrarily fixed and can be used as a fitting parameter.
2. A dynamic contact angle, Equation 9.10 in Section 9.2.2. In this case, however, the angle is a function of the flow at the contact line and the equilibrium angle.
3. A free contact angle, Equation 10.5 in Section 10.2.1. In this final case, the contact angle is considered to be a solution of the flow. The notion of free

contact angle, developed in this work, is used in combination with the GNBC – and with the no-slip model for numerical validation, see Appendix F). The GNBC can be viewed as a boundary condition for the momentum equation with an extracted angle – as opposed to imposed – that is measured at each time step.

In Figure 10.15, we present a table of the different models with the aforementioned categorizations. The color code is as follows:

- In green, the models implemented and tested in the VOF framework that are presented in Chapter 9 and Chapter 10. The toy model is a special type of slip-dynamic angle combination where the computational boundary is lifted off the solid substrate and necessitates a special treatment of the normal velocity component in the boundary condition (Appendix G).
- In blue, the models that were implemented and tested but not presented in this dissertation. In particular, the GNBC patch method [Fullana 2019] was the first iteration of the GNBC where we modified the velocities around the contact line without changing the Navier boundary condition. In that case, a constant contact angle was used.
- In red, the models that are yet to be tested.
- In grey, the combinations of boundary conditions and contact angles that have no ‘meaning’ in our numerical framework. For example, in the case of a GNBC with a dynamic contact angle, the fact that, on one hand the contact line speed  $Ca_{CL}$  is a function of the dynamic angle  $\theta_d$ , but on the other hand  $\theta_d$  is itself a function of the contact line speed  $Ca_{CL}$  clearly leads to a paradox. The free dynamic contact angle in the case of a slip or super-slip model leads to an ill-posed problem, where the condition on the contact angle is lacking and does not enter the boundary condition of the momentum equation.

	Constant angle	Dynamic angle	Free angle
No slip	<ul style="list-style-type: none"> <li>• <math>u_x = U</math></li> <li>• <math>\theta = \theta_e</math></li> <li>• Height functions Section 9.1</li> <li>• Spurious currents Appendix D</li> <li>• Forced dewetting [Afkhami <i>et al.</i> 2017]</li> </ul>	<ul style="list-style-type: none"> <li>• <math>u_x = U</math></li> <li>• <math>\theta_d = f(\theta_e, Ca, L, l)</math> Section 9.2.2</li> <li>• ‘Dyn 1’ model [Legendre &amp; Maglio 2015]</li> </ul>	<ul style="list-style-type: none"> <li>• <math>u_x = U</math></li> <li>• <math>\theta_d = f(\theta_{app}, \kappa, \Delta, h)</math> Section 10.2.1</li> <li>• Angle transport Appendix F</li> </ul>
Slip	<ul style="list-style-type: none"> <li>• <math>u_x = f(\lambda, U)</math> Section 9.2.1</li> <li>• <math>\theta = \theta_e</math></li> <li>• Curtain coating Section 9.3.1 [Fullana <i>et al.</i> 2020]</li> </ul>	<ul style="list-style-type: none"> <li>• <math>u_x = f(\lambda, U)</math></li> <li>• <math>\theta_d = f(\theta_e, Ca, L, l)</math></li> <li>• ‘Dyn 3’ model</li> <li>• Squalane drop Section 9.3.2</li> <li>• Toy model Section 10.3</li> </ul>	
Local slip	<ul style="list-style-type: none"> <li>• <math>u_x = f(\varepsilon, \lambda, U)</math> Section 9.2.1</li> <li>• <math>\theta = \theta_e</math></li> <li>• Sheared droplet Benchmark VOF vs PF and MD [Lācis <i>et al.</i> 2020]</li> </ul>	<ul style="list-style-type: none"> <li>• <math>u_x = f(\varepsilon, \lambda, U)</math></li> <li>• <math>\theta_d = f(\theta_e, Ca, L, l)</math></li> </ul>	
Super-slip	<ul style="list-style-type: none"> <li>• <math>u_x = f(\lambda_1, \lambda_2, U)</math> Section 10.1.1</li> <li>• <math>\theta = \theta_e</math></li> <li>• Validation Appendix E</li> <li>• Spreading droplet Section 10.1.2</li> </ul>	<ul style="list-style-type: none"> <li>• <math>u_x = f(\lambda_1, \lambda_2, U)</math></li> <li>• <math>\theta_d = f(\theta_e, Ca, L, l)</math></li> </ul>	
GNBC	<ul style="list-style-type: none"> <li>• <math>u_x = f(\varepsilon, \theta_d, \theta_e, U)</math></li> <li>• <math>\theta = \theta_e</math></li> <li>• Patch method [Fullana 2019]</li> </ul>		<ul style="list-style-type: none"> <li>• <math>u_x = f(\varepsilon, \theta_d, \theta_e, U)</math> Section 10.2.1</li> <li>• <math>\theta_d = f(\theta_{app}, \kappa, \Delta, h)</math></li> <li>• Forced dewetting Section 10.2.2</li> </ul>

Figure 10.15: Summary of the different models for the dynamic contact line.



# Conclusion

In the first part of this dissertation, we proposed a level set method coupled with a novel Cut Cell approach to tackle two-phase Stefan problems as well as an adjoint-based optimization procedure. The key features of the level set related algorithms are, (i) an implicit-explicit scheme to solve the level set advection allowing us to relax the usual CFL condition, (ii) the high-order Johansen-Colella method used to compute the normal gradient across the interface, and (iii) the sub-cell resolution reinitialization procedure to retain the signed distance function property as the interface moves. Moreover, we use the Cut Cell method coupled with a Crank-Nicolson time integrator that allows us to solve the two-phase problem for any given geometry. The method was then extended to solve the Navier-Stokes equations in the fluid phase, and compares favorably with previous computations on the onset of Rayleigh-Bénard instabilities. This Cut Cell method is currently being extended to higher dimensions.

In the context of classical two-phase Stefan problems, the adjoint-based optimization procedure is shown to be a robust algorithm to control the shape of a melting or solidification front, even in the presence of dendritic instabilities and anisotropic effects. The comparison with derivative-free methods yields favorable results, as the gradient-based method converge faster. Nevertheless, the final value of the cost functional is, as expected, lower in the derivative-free cases. As a future work, one could consider to employ hybrid optimization methods, such as the GDYCORs (Gradient-enhanced DYnamic COordinate search using Response Surface models).

The derivation of the continuous adjoint in the presence of a flow in the liquid phase could also be a direction to follow. In that case, the use of an incomplete adjoint – were the velocity field is a ‘known’ quantity in the adjoint problem – might be required. In this particular case, to avoid any substantial simplifications, one could also consider the ‘discretize-then-differentiate’ approach were a discrete-adjoint-based optimization algorithm – as opposed to a continuous one – would be built.

In the second part of this work we presented several contact line models in a Volume-Of-Fluid framework. These models were validated and tested on physical setups. Describing the physics of the dynamic contact line with sharp-interface method is still a challenge. The models yield different results depending on the physical problems considered and a consensus on the ‘best’ approach to adopt is still debated. Our work is an attempt to enrich the field with various new numerical methods for contact lines.

In particular, the GNBC based on the VOF method, where the angle is a function of the flow, presents attractive grid-independent properties, provided that the smallest length scale is well-resolved. On the other hand, the toy model model is an interesting way of modeling evaporation – mechanism driving of the motion of the

contact line – through simple boundary conditions in the VOF method. This model still needs to be tested on a wider range of problems.

As a final remark, one could implement these models in our level set method, thus modeling phase-change phenomena in the presence of contact lines. An interesting advantage would be to use the already existing adjoint-based – or derivative-free – optimization procedures and test them on case where the control variable is, for example, the contact angle.



Part III

Appendices

# Transport and shape calculus theorems

---

In this section, we present the transport and shape calculus theorems, similarly to [Bernauer & Herzog 2011], that are used in the continuous adjoint derivation (Chapter 2).

**Theorem A (Reynolds transport theorem)** *The derivative of the quantity*

$$F(t) := \int_{\Omega(t)} f(x, t) dx,$$

*is given by*

$$\begin{aligned} \frac{dF}{dt}(t) &= \int_{\Omega(t)} \frac{\partial f}{\partial t} + \operatorname{div}(f V) dx \\ &= \int_{\Omega(t)} \frac{df}{dt} + f \operatorname{div}(V) dx \\ &= \int_{\Omega(t)} \frac{\partial f}{\partial t} dx + \int_{\partial\Omega(t)} f V \cdot n ds, \end{aligned}$$

*where  $V$  is the velocity field in which the control volume  $\Omega(t)$  moves.*

**Corollary A (Integration by Parts in Time in Moving Domains)** *For  $g = g(x, t)$  and  $h = h(x, t)$ , we have*

$$\begin{aligned} \int_0^{t_f} \int_{\Omega(t)} g h_t dx dt &= \int_{\Omega(t_f)} g(x, t_f) h(x, t_f) dx - \int_{\Omega(0)} g(x, 0) h(x, 0) dx \\ &\quad - \int_0^{t_f} \int_{\Omega(t)} g_t h dx dt - \int_0^{t_f} \int_{\partial\Omega(t)} g h V \cdot n ds dt. \end{aligned}$$

**Theorem B (Derivative of boundary integral)** *Let  $J(\Omega) = \int_{\Gamma} f ds$  be a boundary integral, the derivative is given by*

$$\begin{aligned} dJ(\Omega; V) &= \left( \frac{d}{d\lambda} \int_{\Gamma_\lambda} f ds_\lambda \right) \Big|_{\lambda=0} \\ &= - \int_D \frac{\delta\phi}{|\nabla\phi|} \left( \frac{\partial f}{\partial n} + f \kappa \right) ds, \end{aligned}$$

*where  $V$  is the velocity field and  $\kappa$  is the mean curvature of  $\Gamma$ .*

**Theorem C (Surface transport theorem)** *Let  $f(\cdot, t) : \mathcal{S}_t \rightarrow \mathbb{R}$  be a scalar field defined on the moving surface  $\mathcal{S}_t$ . Then*

$$\frac{d}{dt} \int_{\mathcal{S}_t} f(x, t) d\mathcal{S}_t = \int_{\mathcal{S}_t} \dot{f}(x, t) + f(x, t) \operatorname{div}_{\mathcal{S}_t} \mathbf{w}(x, t) d\mathcal{S}_t,$$

where  $\mathbf{w}$  is the normal velocity of the moving surface  $\mathcal{S}_t$  and  $\dot{f}$  is the parameter-time derivative of  $f$ . If  $f(\cdot, t)$  is the restriction of a function  $\hat{f}(\cdot, t)$  to  $\mathcal{S}_t$ , then

$$\dot{f}(x, t) = \mathbf{w}(x, t) \cdot \nabla \hat{f}(x, t) + \frac{\partial}{\partial t} \hat{f}(x, t).$$

**Corollary C (Integration by Parts in Time on a Moving Surface)**

$$\begin{aligned} \int_0^T \int_{\mathcal{S}_t} \hat{g}(x, t) \hat{h}_t(x, t) d\mathcal{S}_t dt &= \int_{\mathcal{S}_T} g(x, T) h(x, T) d\mathcal{S}_T - \int_{\mathcal{S}_0} g(x, 0) h(x, 0) d\mathcal{S}_0 \\ &\quad - \int_0^T \int_{\mathcal{S}_t} \hat{g}_t(x, t) \hat{h}(x, t) + \mathbf{w}(x, t) \cdot \nabla(\hat{g}(x, t) \hat{h}(x, t)) \\ &\quad + g(x, t) h(x, t) \operatorname{div}_{\mathcal{S}_t} \mathbf{w} d\mathcal{S}_t dt, \end{aligned}$$

where  $g$  and  $h$  are restrictions of  $\hat{g}$  and  $\hat{h}$  to  $\mathcal{S}_t$ .

# Cut Cell method for Navier-Stokes equations

---

This section presents the proposed discretization of the incompressible Navier-Stokes equations for an isotropic Newtonian fluid in the context of the Cut Cell method. Details of this discretization – for stationary geometries – can be found in [Quirós Rodríguez *et al.* 2022]. The equations read

$$\begin{cases} \rho \left[ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = -\nabla p + \nabla \cdot (2\mu \mathbf{s}) + \rho \mathbf{g}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases}$$

where  $\mathbf{u}$  and  $p$  respectively denote the fluid's velocity and pressure fields,  $\rho$  its constant density and  $\mathbf{g}$  the gravitational acceleration. Additionally,  $\mu$  denotes the fluid's constant dynamic viscosity and

$$\mathbf{s} \equiv \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^\top}{2},$$

the strain-rate tensor.  $P = (P_{ij})$  represents the (cell-centered) pressure field, and

$$\mathbf{U} = (U_x, U_y) = \left( (U_{i+1/2,j}^x), (U_{i,j+1/2}^y) \right),$$

the (staggered) Cartesian components of the velocity field. Finally,  $\mathbf{D} = (D_x, D_y)$  denotes the (staggered) boundary conditions to be applied on the velocity field.

**Velocity divergence and pressure gradient** Let  $\Omega_{ij}^f = \Omega_{ij} \cap \Omega^f$  denote the subset of  $\Omega_{ij}$  wet by the fluid,  $\mathbf{u}$  the continuous fluid velocity field and  $\mathbf{d}$  the boundary condition. Then, Stokes' divergence theorem

$$\int_{\Omega_{ij}^f} \nabla \cdot \mathbf{u} = \int_{\partial \Omega_{ij}^f \setminus \Gamma} \mathbf{u} \cdot \mathbf{n} + \int_{\partial \Omega_{ij}^f \cap \Gamma} \mathbf{d} \cdot \mathbf{n},$$

states that the volume integral of the velocity divergence matches the net volume fluxes, summed over the surfaces immersed in the fluid itself and adjacent to the boundary. The former term, referred to as homogeneous, quantifies the exchange of volume with the neighboring fluid elements, and the latter, referred to as heterogeneous, quantifies this exchange with the exterior domain through the boundary.

This decomposition is reflected at the discrete level by discretizing the volume-integrated velocity divergence as

$$\text{cont}(\mathbf{U}, \mathbf{D}) \equiv \sum_{\alpha} \left( \frac{\delta A_{\alpha} U_{\alpha}}{\delta \xi_{\alpha}} + \frac{\delta (\overline{B_{\alpha}^{\alpha}} - A_{\alpha}) D_{\alpha}}{\delta \xi_{\alpha}} - \frac{\delta \overline{B_{\alpha}^{\alpha}} D_{\alpha}}{\delta \xi_{\alpha}} \right). \quad (\text{B.1})$$

The divergence free condition, then, is expressed as

$$\text{cont}(\mathbf{U}, \mathbf{D}) = 0,$$

and the (volume integrated)  $\alpha$  component of the pressure gradient, a linear operator denoted as  $\text{pres}_{\alpha}$ , is simply defined as the negative transpose of the Jacobian of Equation B.1 with respect to  $U_{\alpha}$ , namely

$$\forall \alpha \in \{x, y\}, \quad \frac{\partial \text{pres}_{\alpha}}{\partial P} = - \left( \frac{\partial \text{cont}}{\partial U_{\alpha}} \right)_{U_{\beta \neq \alpha}, \mathbf{D}}^{\top}, \quad (\text{B.2})$$

which yields

$$\forall \alpha \in \{x, y\}, \quad \text{pres}_{\alpha}(P) \equiv A_{\alpha} \frac{\delta P}{\delta \xi_{\alpha}}.$$

This construction is rooted in the geometric interpretation of the incompressible Navier-Stokes equations [Arnold 1966], which exposes the dual role of the pressure in imposing the divergence-free condition, and commonly used in both structured and unstructured settings [Chen & Botella 2010, Perot 2011].

**Strain-rate tensor** The components of the diagonal element of the strain-rate tensor are cell-centered discrete counterparts of

$$s_{\alpha\alpha} = \frac{\partial u_{\alpha}}{\partial x_{\alpha}}, \quad \alpha \in \{x, y\},$$

defined based upon the gradient formula Equation 3.12. First, the surface and volume capacities  $\mathbf{W} = (W_{\beta})$ ,  $\mathbf{A} = (A_{\beta})$  and  $\mathbf{B} = (B_{\beta})$  are replaced by those after shifting the mesh in half a grid spacing along direction  $\alpha$  defined in Section 3.1, namely  $\mathbf{W}^{\alpha} = (W_{\alpha\beta})$ ,  $\mathbf{A}^{\alpha} = (A_{\alpha\beta})$  and  $\mathbf{B} = (B_{\alpha\beta})$ . Second, the dependent field  $T$  and the Dirichlet boundary condition  $D$  are substituted with  $U_{\alpha}$  and  $D_{\alpha}$ , respectively. This finally yields

$$\forall \alpha \in \{x, y\}, \quad \text{strain}_{\alpha\alpha}(\mathbf{U}, \mathbf{D}) = \frac{1}{W_{\alpha\alpha}} \left[ \frac{\delta B_{\alpha\alpha} U_{\alpha}}{\delta \xi_{\alpha}} + \frac{\delta (\overline{A_{\alpha\alpha}^{\alpha}} - B_{\alpha\alpha}) D_{\alpha}}{\delta \xi_{\alpha}} - \frac{\delta \overline{A_{\alpha\alpha}^{\alpha}} D_{\alpha}}{\delta \xi_{\alpha}} \right]. \quad (\text{B.3})$$

This process is repeated for the components of the off-diagonal elements of the strain-rate tensor, defined in the continuous case as

$$s_{\alpha\beta} = \frac{1}{2} \left( \frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right), \quad \alpha \neq \beta,$$

and in the discrete case as the node-centered field

$$\begin{aligned} \forall (\alpha, \beta) \in \{x, y\}^2, \quad \alpha \neq \beta, \quad \text{strain}_{\alpha\beta}(\mathbf{U}, \mathbf{D}) = \\ \frac{1}{2W_{\alpha\beta}} \left[ \frac{\delta B_{\alpha\beta} U_\alpha}{\delta \xi_\beta} + \frac{\delta (\overline{A_{\alpha\beta}^\beta} - B_{\alpha\beta}) D_\alpha}{\delta \xi_\beta} - \frac{\delta \overline{A_{\alpha\beta}^\beta}}{\delta \xi_\beta} D_\alpha \right] \\ + \frac{1}{2W_{\beta\alpha}} \left[ \frac{\delta B_{\beta\alpha} U_\beta}{\delta \xi_\alpha} + \frac{\delta (\overline{A_{\beta\alpha}^\alpha} - B_{\beta\alpha}) D_\beta}{\delta \xi_\alpha} - \frac{\delta \overline{A_{\beta\alpha}^\alpha}}{\delta \xi_\alpha} D_\beta \right]. \end{aligned} \quad (\text{B.4})$$

It should finally be noted that the latter formula (Equation B.4) is also valid in the diagonal case ( $\alpha = \beta$ ), in which case it simply reduces to Equation B.3.

**Viscous transport term** Prior to proceeding with the discretization of the viscous transport term, it should first be noted that, in the case where the second argument ( $\mathbf{N}$ ) of the divergence operator (Equation 3.13 summed over  $\alpha$ ) matches the first argument ( $\mathbf{Q}$ ), Equation 3.13 may be simplified using the identities presented by [Morinishi 2010] as

$$\text{div}(\mathbf{Q}, \mathbf{Q}) = \sum_{\beta} B_{\beta} \frac{\delta Q_{\beta}}{\delta \xi_{\beta}}. \quad (\text{B.5})$$

Therefore, the discretization of the viscous transport term,  $\nabla \cdot (2\mu s)$ , is performed similarly to that of the strain-rate operator, by translating the definition of the capacities to yield

$$\forall \alpha \in \{x, y\}, \quad \text{visc}_{\alpha}(\mathbf{S}) = \sum_{\beta} B_{\alpha\beta} \frac{\delta S_{\alpha\beta}}{\delta \xi_{\beta}},$$

where  $\mathbf{S} = (S_{\alpha\beta})$  is defined as a function of  $\mathbf{U}$  and  $\mathbf{D}$  by Equations B.3 and B.4.

**Convective transport term** The convective term in the momentum transport equation along  $\alpha \in \{x, y\}$  is rewritten in conservative form using the divergence-free condition,

$$(\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \mathbf{u} = \nabla \cdot (\mathbf{u} \otimes \mathbf{u}),$$

which in discrete form can be written as

$$\text{conv}_{\alpha}(\mathbf{U}, \mathbf{U}^{\dagger}, \mathbf{D}, \mathbf{D}^{\dagger}) =$$

$$\sum_{\beta} \left\{ \frac{\overline{\delta A_{\beta} U_{\beta}^{\alpha} U_{\alpha}^{\dagger \beta}}}{\delta \xi_{\beta}} + \left[ \frac{\overline{\delta (\overline{B_{\beta}^{\beta}} - A_{\beta}) D_{\beta}}}{\delta \xi_{\beta}} - \frac{\overline{\delta B_{\beta}^{\beta}}}{\delta \xi_{\beta}} D_{\beta} \right] \frac{U_{\alpha}^{\dagger} + D_{\alpha}^{\dagger}}{2} \right\}. \quad (\text{B.6})$$

This multilinear operator is typically evaluated at  $\mathbf{U}^{\dagger} = \mathbf{U}$  and  $\mathbf{D}^{\dagger} = \mathbf{D}$  but the distinction might bear significance, in the context of Picart linearization for example where a distinction applies between  $\mathbf{U}$  which is typically *frozen* whereas  $\mathbf{U}^{\dagger}$  is updated. This discretization can be considered as the generalization of the centered scheme to the cut cell method, which can be demonstrated as follows. In the continuous case,

$$\forall (\alpha, \beta) \in \{x, y\}^2, \quad u_{\alpha} \frac{\partial u_{\beta} u_{\alpha}}{\partial x_{\beta}} = \frac{\partial u_{\beta} u_{\alpha}^2 / 2}{\partial x_{\beta}} + \frac{u_{\alpha}^2}{2} \frac{\partial u_{\beta}}{\partial x_{\beta}}, \quad (\text{B.7})$$

which, upon summation over  $\alpha$ , yields a similar equation for the specific kinetic energy  $k \equiv \|\mathbf{u}\|^2/2$ , ultimately conserved in the inviscid limit. The proposed discretization of the convective transport term (Equation B.6) preserves this property at the discrete level. Using the identities presented by [Morinishi 2010], it can be shown that

$$\forall (\alpha, \beta) \in \{x, y\}^2, \quad U_{\alpha}^{\dagger} \frac{\overline{\delta A_{\beta} U_{\beta}^{\alpha} U_{\alpha}^{\dagger \beta}}}{\delta \xi_{\beta}} = \frac{\overline{\delta A_{\beta} U_{\beta}^{\alpha} \widetilde{U_{\alpha}^{\dagger} U_{\alpha}^{\dagger}}}}{\delta \xi_{\beta}} / 2 + \frac{U_{\alpha}^{\dagger 2}}{2} \frac{\overline{\delta A_{\beta} U_{\beta}^{\alpha}}}{\delta \xi_{\beta}}, \quad (\text{B.8})$$

where  $\widetilde{\cdot}$  denotes the permanent product

$$\widetilde{\phi \psi} \Big|_{i+1/2, j}^x = \frac{\phi_{i+1, j} \psi_{ij} + \psi_{i+1, j} \phi_{ij}}{2}, \quad (\text{B.9})$$

also introduced by [Morinishi 2010] and easily extended to other dimensions and arrangements as previously done for differentiation and interpolation. Equation B.8, together with the continuity operator (Equation B.1), can be used to show that  $\forall \alpha \in \{x, y\}$

$$\begin{aligned} U_{\alpha}^{\dagger} \text{conv}_{\alpha} (\mathbf{U}, \mathbf{U}^{\dagger}, \mathbf{D}, \mathbf{D}^{\dagger}) = \\ \sum_{\beta} \left\{ \frac{\overline{\delta A_{\beta} U_{\beta}^{\alpha} \widetilde{U_{\alpha}^{\dagger} U_{\alpha}^{\dagger}}}}{\delta \xi_{\beta}} / 2 + \left[ \frac{\overline{\delta (\overline{B_{\beta}^{\beta}} - A_{\beta}) D_{\beta}}}{\delta \xi_{\beta}} - \frac{\overline{\delta B_{\beta}^{\beta}}}{\delta \xi_{\beta}} D_{\beta} \right] \frac{U_{\alpha}^{\dagger} D_{\alpha}^{\dagger}}{2} \right\} \\ + \frac{U_{\alpha}^{\dagger 2}}{2} \overline{\text{cont} (\mathbf{U}, \mathbf{D})}^{\alpha}. \quad (\text{B.10}) \end{aligned}$$

This identity can be interpolated in each direction  $\alpha$ , and summed over  $\alpha$ , to ultimately state the proposed discretization (Equation B.6) conserves kinetic energy, in

the sense that the rate of change of the discrete kinetic energy

$$\text{kinetic}(\mathbf{U}^\dagger) \equiv \sum_{\alpha} \frac{1}{2} \overline{V_{\alpha} U_{\alpha}^{\dagger} U_{\alpha}^{\dagger \alpha}},$$

is a result of an exchange with the neighboring fluid elements (first term in the right-hand side of Equation B.10) and across the boundary (second term).

**Semi-discrete system** The face-centered mass matrices appearing in front of the rate of change and body forces are diagonal with coefficients  $\mathbf{V} = (V_{\alpha})$  (the volume of the staggered control volumes, defined in Section 3.1) and are denoted as

$$\forall \{x, y\}, \quad \mathcal{M}_{\alpha} \equiv \text{diag}(V_{\alpha}).$$

Gathering all the terms, the proposed semi-discrete momentum equations then read ( $\alpha \in \{x, y\}$ )

$$\rho \left[ \mathcal{M}_{\alpha} \frac{dU_{\alpha}}{dt} + \text{conv}_{\alpha}(\mathbf{U}, \mathbf{U}, \mathbf{D}, \mathbf{D}) \right] = -\text{pres}_{\alpha}(P) + \text{visc}_{\alpha}(2\mu\mathbf{S}) + \rho\mathcal{M}_{\alpha}\mathbf{g}, \quad (\text{B.11})$$

with divergence-free condition

$$\text{cont}(\mathbf{U}, \mathbf{D}) = 0. \quad (\text{B.12})$$

The system is closed with the discrete strain-rate tensor  $\mathbf{S}$ , defined as a function of  $\mathbf{U}$  and  $\mathbf{D}$  as follows,

$$\forall (\alpha, \beta) \in \{x, y\}^2, \quad S_{\alpha\beta} = \text{strain}_{\alpha\beta}(\mathbf{U}, \mathbf{D}), \quad (\text{B.13})$$

where the operators  $\text{strain}_{\alpha\beta}$  are defined by Equations B.3 and B.4.

All of the operators appearing in Equations B.11, B.12 and B.13 are linear in all dependent variables ( $P$ ,  $\mathbf{U}$  and  $\mathbf{S}$ ) and boundary condition  $\mathbf{D}$  with the exception of the convective transport operators ( $\text{conv}_{\alpha}$ ) defined in Equation B.6) which is quadratic when evaluated at  $\mathbf{U}^\dagger = \mathbf{U}$  and  $\mathbf{D}^\dagger = \mathbf{D}$ .

**Projection method** The discretization of the aforementioned incompressible Navier-Stokes equations results in a saddle point system of equations [Benzi *et al.* 2005], sometimes also called Karush-Kuhn-Tucker (KKT) system [Nocedal & Wright 2006] in optimization. A wide range of algorithms have been devised to efficiently solve saddle point systems (or approximation thereof). In the field of fluid mechanics, a common approach is the fractional step method [Chorin 1968]. In the present work, the method referred to as projection method II (PmII) by [Brown *et al.* 2001], which ensures a second order discretization of the equations, is employed.

In this projection method, the convective term is discretized using the explicit second-order Adams-Bashforth scheme and the viscous term is discretized using the

implicit Crank-Nicolson scheme. The first step of the method consists of obtaining an intermediate velocity field  $\mathbf{U}^*$  by solving

$$\begin{aligned} \rho \mathcal{M}_\alpha \frac{U_\alpha^* - U_\alpha^n}{\tau} + \frac{3\rho}{2} \text{conv}_\alpha(\mathbf{U}^n, \mathbf{U}^n, \mathbf{D}^n, \mathbf{D}^n) \\ - \frac{\rho}{2} \text{conv}_\alpha(\mathbf{U}^{n-1}, \mathbf{U}^{n-1}, \mathbf{D}^{n-1}, \mathbf{D}^{n-1}) = -\text{pres}_\alpha(P^{n-1/2}) \\ + \text{visc}_\alpha(\mu \mathbf{S}^*) + \text{visc}_\alpha(\mu \mathbf{S}^n) + \rho \mathcal{M}_\alpha \mathbf{g}, \end{aligned} \quad (\text{B.14})$$

where  $\tau$  denotes the time step and the superscript  $n$  the iteration number. The boundary conditions applicable to  $\mathbf{U}^*$  (the predicted velocity field) and used in  $\mathbf{S}^*$  are those of the velocity field at the next time step ( $\mathbf{D}^* = \mathbf{D}^{n+1}$ )

$$\forall (\alpha, \beta), \quad S_{\alpha\beta}^n = \text{strain}_{\alpha\beta}(\mathbf{U}^n, \mathbf{D}^n) \quad \text{and} \quad S_{\alpha\beta}^* = \text{strain}_{\alpha\beta}(\mathbf{U}^*, \mathbf{D}^{n+1}).$$

In the projection step, the velocity field is updated by projecting  $\mathbf{U}^*$  using the intermediate pressure field  $\Phi^{n+1}$ , which is obtained by solving the following Poisson equation

$$\tau \text{cont}(\text{pres}(\Phi^{n+1}), \mathbf{0}) = \text{cont}(\mathbf{U}^*, \mathbf{D}^{n+1}), \quad (\text{B.15})$$

with a homogeneous Neumann boundary conditions being used for the intermediate pressure ( $\mathbf{0}$ ). The velocity field is ultimately corrected as

$$U_\alpha^{n+1} = U_\alpha^* - \tau \text{pres}_\alpha(\Phi^{n+1}). \quad (\text{B.16})$$

The pressure is finally updated as

$$P^{n+1/2} = P^{n-1/2} + \Phi^{n+1} - \frac{\tau\mu}{2\rho} \text{cont}(\text{pres}(\Phi^{n+1}), \mathbf{0}), \quad (\text{B.17})$$

where the last term ensures the second order accuracy of the pressure field.

Thus far, only Dirichlet boundary conditions for the velocity field have been considered, which are paired with homogeneous boundary conditions for the pressure in the projection step. Cases will be considered in the following section where Neumann boundary conditions are required along the outflow boundaries. Along their vicinity, a Dirichlet boundary condition for the pressure is employed in order to uphold the compatibility equation B.2.

Finally, the use of periodic and/or Neumann boundary conditions gives rise to a rank deficiency in the Laplacian operator. This results in the pressure field being known up to a constant. This knowledge is exploited in the iterative solution of the Poisson equation by projecting the updates in the space of zero-mean solutions.

# Flower.jl package description

---

In this section, we detail the Julia code *Flower.jl* used in the first part of the dissertation. The code architecture can be summarized as follows:

- The numerical and physical parameters – defined by the user when initializing a problem – the meshes, discrete operators and fields are contained in different mutable data structures that enter the main solving functions.
- Two main functions, with options determining the type of problem to consider, are defined. The first one – *run\_forward(...)* – solves the forward Stefan problem (Algorithm 4) and the second one – *run\_backward(...)* – solves the adjoint Stefan problem (Algorithm 6).

We now present a typical Julia file that uses *Flower.jl* to solve, for example, the Rayleigh-Bénard convection case (Section 5.4). We start by defining the numerical and physical variables contained in the *num* structure:

```
using Flower      # Load the structures and functions contained in the package

L0 = 1.0         # Size of the domain
H0 = 0.05        # Initial height of the fluid layer
R = 8            # Aspect ratio
nx = R*64        # Number of points in the x-direction
ny = 64          # Number of points in the y-direction

x = LinRange(0.0, R*L0, nx+1)      # x and y segments with the origin at (0,0)
y = LinRange(0.0, L0, ny+1)

# Define the numerical and physical parameters
num = Numerical(CFL = 0.5,          # CFL condition, defines the time step
  x = x,                             # Set x in num
  y = y,                             # Set y in num
  NB = 4,                             # Narrow band width
  reinit_every = 5,                  # Reinitialize the level set every 5 time steps
  TEND = 1.0,                        # Final time of the simulation
  save_every = 100,                  # Save the data every 100 time steps
  u_inf = 0.0,                       # Initial velocity field
  T_d = 0.1,                         # Melting temperature
  ε_k = 0.0,                         # Surface tension coefficient
  ε_v = 0.0,                         # Molecular kinematic coefficient
  Ra = 10^6,                         # Global Rayleigh number
  L_H = 2.0,                         # Latent heat of solidification
  T1 = 1.0,                          # Bottom wall temperature
  T2 = 0.0                           # Top wall temperature
)
```

Next, the meshes ( $gp, gu, gv$ ), Cut Cell operators in both phases ( $opS, opL$ ), fields in both phases ( $phS, phL$ ), a structure to save the data as the ‘forward’ simulation progresses ( $fwd$ ) and the level set function ( $gp.phi$ ) are initialized:

```
# Initialize the meshes (centered, staggered in x, staggered in y)
gp, gu, gv = init_mesher(num)

# Initialize the discrete operators and fields in both phases
opS, opL, phS, phL, fwd = init_fields(num, gp, gu, gv);

# Define the initial level set function phi as a plane located at y = H0
@. gp.phi = -gp.y + H0
```

The boundary conditions for the level set function, temperature, velocity and pressure fields in both phases are set as follow (if not specified, the boundary conditions are of homogeneous Neumann type on all the domain boundaries):

```
# Set the boundary conditions for the different fields
# If not specified, the boundary condition is a homogeneous Neumann BC

# Liquid temperature field
BC_TL = Boundaries(
    bottom = Boundary(t = dir, val = num.T1)    # Dirichlet at the bottom wall
)

# Solid temperature field
BC_TS = Boundaries(
    top = Boundary(t = dir, val = num.T2)       # Dirichlet at the top wall
)

# x component of the velocity field in the liquid phase
BC_uL = Boundaries(
    bottom = Boundary(t = dir, val = 0.0),      # Dirichlet at the bottom wall
    top = Boundary(t = dir, val = 0.0)         # Dirichlet at the top wall
)

# y component of the velocity field in the liquid phase
BC_vL = Boundaries(
    bottom = Boundary(t = dir, val = 0.0),      # Dirichlet at the bottom wall
    top = Boundary(t = dir, val = 0.0)         # Dirichlet at the top wall
)
```

We can now run the simulation using the main function `run_forward(...)`. In this case, we solve:

- The heat equation (Equation 1.1) in the solid phase.
- The Stefan condition (Equation 1.2) at the interface.
- The diffusion-convection equation and the Navier-Stokes equations with the Boussinesq approximation (Equation 1.11) in the liquid phase.

```
# Run the simulation
run_forward(num,
    gp, gu, gv,
    opS, opL,
    phS, phL,
    fwd,
    BC_TL = BC_TL,           # Set the boundary conditions for the solver
    BC_TS = BC_TS,
    BC_uL = BC_uL,
    BC_vL = BC_vL,

    periodic_x = true,      # Periodic in the x-direction
                           # Overwrites the homogeneous Neumann BC

    stefan = true,         # Stefan condition at the interface

    advection = true,      # Level set advection equation

    heat = true,           # Solve the heat equations
    heat_convection = true, # with the convective term
    heat_liquid_phase = true, # in the liquid phase
    heat_solid_phase = true, # and in the solid phase

    navier_stokes = true,  # Solve the NS equations
    ns_advection = true,   # with the convective term
    ns_liquid_phase = true, # only in the liquid phase
    ns_solid_phase = false,

    adaptive_t = true,     # Adaptive time step

    verbose = true,        # Show simulation statistics
)
```



# Spurious currents study for static drops

---

The presence of spurious currents when studying flows at low capillary number are a known issue in the CSF-VOF method [Popinet 2018], as well as for other methods (level set, front-tracking). Nevertheless, some improvements in the balance of surface forces and discrete pressure gradients are still possible. In [Abu-Al-Saud *et al.* 2018], the authors presented a new numerical scheme to model surface tension for an interface represented by a level-set function that conserves fluid momentum and recovers Laplace’s equilibrium exactly.

In this section, we look at the spurious currents for a static drop in a closed domain. The system is characterized by the Laplace number, representing the ratio of surface tension to the viscous dissipation

$$\text{La} = \frac{\sigma \rho D}{\mu^2},$$

where  $D$  is a characteristic length scale of the system, the diameter of the drop in our case. Similarly to [Abu-Al-Saud *et al.* 2018], we will look at the effect of the Laplace number and the grid resolution on the spurious currents by measuring the maximal velocity as simulation proceeds in time. In addition to that, we will study the effect of an initial shift in the x-direction – therefore modifying the volume fraction in the interfacial cells – and the effect of a  $90^\circ$  contact angle imposed at the bottom wall. We will consider 6 ( $2 \times 3$ ) cases (presented in Figure D.1)

- A drop initialized in the middle of the domain (no-contact) or in contact with the substrate.
- A initial shift in the x-direction ( $0, \Delta/4$  or  $\Delta/2$ ) with  $\Delta$  the grid size.

In each case, we will vary

- The Laplace number :  $\text{La} = 200, 400, 800, 1200$
- The grid resolution :  $D/\Delta = 16, 32, 64$

In our simulations, the drop is initialized with a diameter  $D = 1$  in a  $2 \times 2$  domain. The densities of both phases and the surface tension are set to 1. The viscosity (equal in both phases) will vary as

$$\mu = \frac{1}{\sqrt{\text{La}}}.$$

The simulations are run until a final time  $t/t_\mu = 100$  where  $t_\mu = D^2/\mu$  is the viscous time scale. We look at the evolution in time of the maximal dimensionless velocity scaled with the capillary number

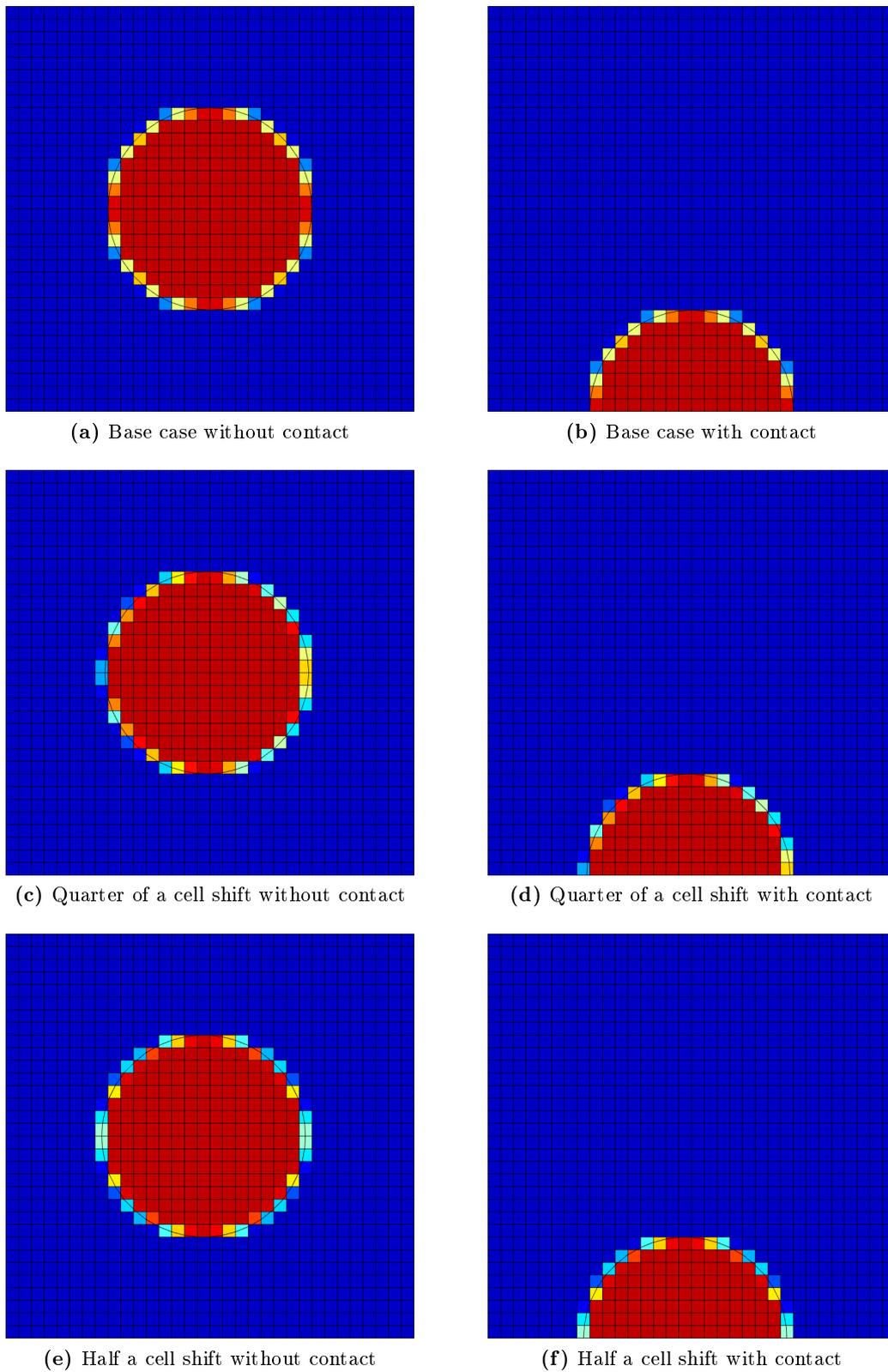
$$\text{Ca}_{\max} = \frac{\mu|u|_\infty}{\sigma}.$$

In Figures D.2 to D.4, the results will be compared two by two (no-contact or contact) the show the effect of the boundary condition on the spurious currents.

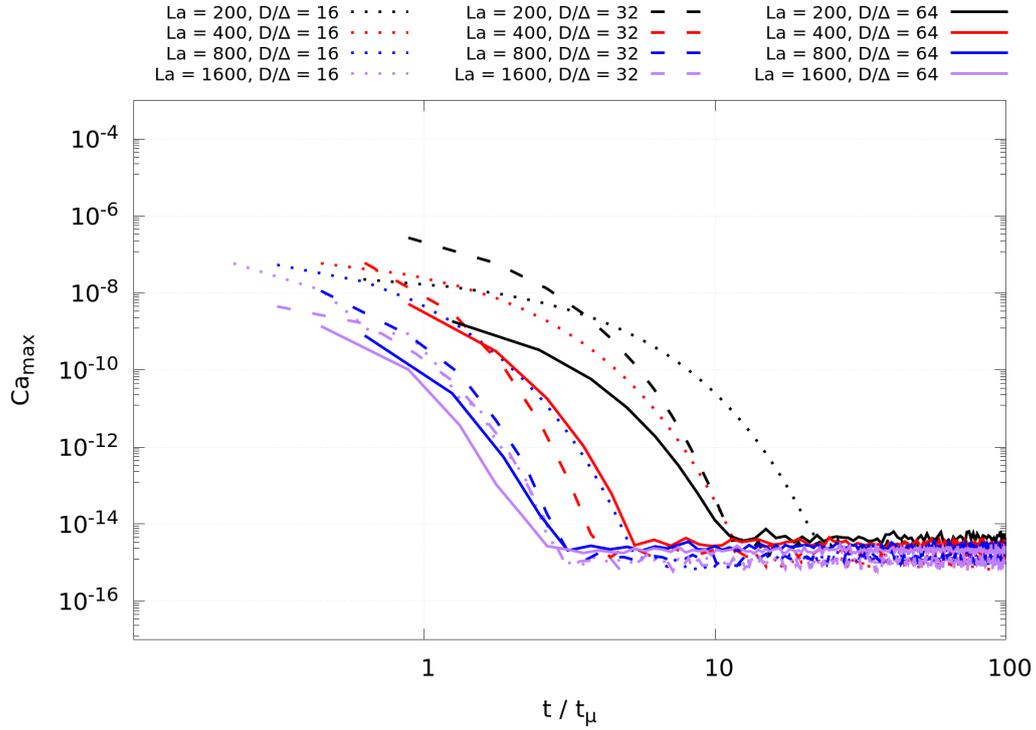
Figure D.2 shows the spurious currents in the case where no shift is applied. For all the combination of parameters, the maximum velocity eventually converges to zero within machine precision. The first clear observation is that the damping time in which the spurious currents are killed is strongly dependent on the Laplace number. As expected, as  $\text{La}$  increases (lines from black to purple), the time to reach machine precision decreases. In Figure D.2a (no-contact), for  $\text{La} = 200$ , we observe a clear convergence as the grid size is decreased. In Figure D.2b (contact), the time to damping is greatly amplified in the  $D/\Delta = 16$  cases. This might come from numerical errors in the height functions due to an insufficient resolution.

Figure D.3 shows the spurious currents in the case where a  $\Delta/4$  shift is applied. The previously grid aligned volume fractions, where  $c = 1$ , are now displaced. In the no-contact case, we see similar results than in the previous figure. However, the grid convergence is affected as the highest resolution cases are not the ones with smaller damping times. In Figure D.3b, we see spurious currents that do not vanish as time proceeds. For  $\text{La} = 200$ , at the highest resolution  $D/\Delta = 64$ , the maximal velocity does not converge towards machine accuracy. This is an example of problematic spurious currents. In practical simulations, the flow around the contact line could be considerably affected leading to an accumulation of numerical errors.

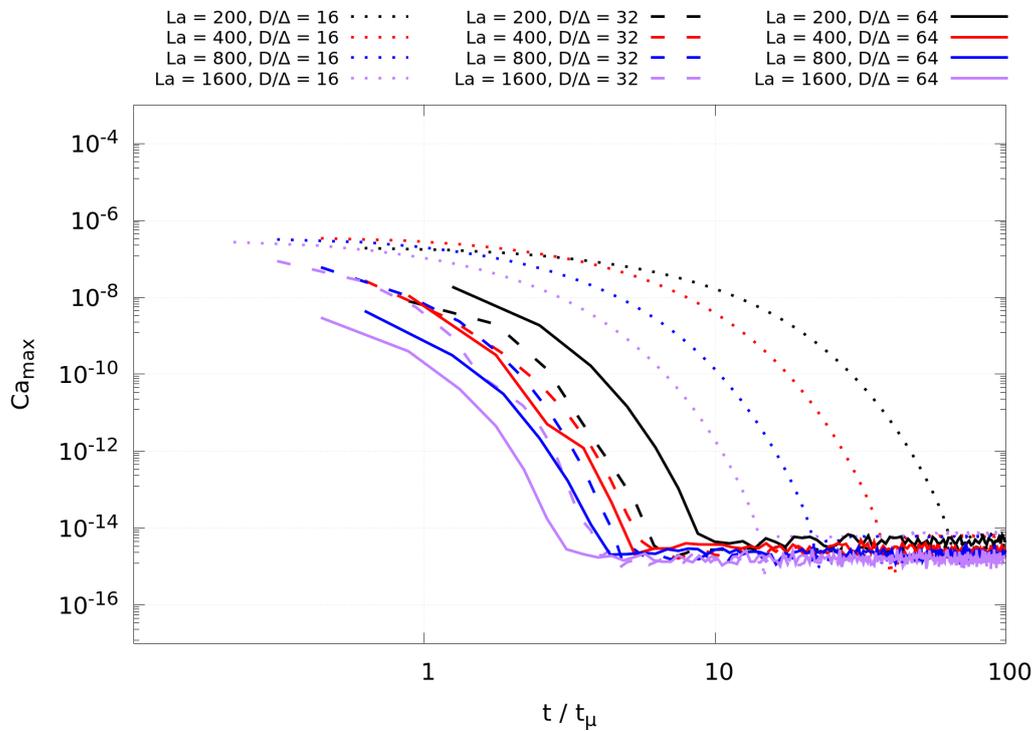
In Figure D.4 ( $\Delta/2$  shift), however, we notice the opposite situation. The contact with the substrate seems to have a regularizing effect on the solution as all the velocities tend to machine precision. In the no-contact case, all the simulations with a grid  $D/\Delta = 32$  exhibit non-vanishing spurious currents regardless of the Laplace number. This counter-intuitive case might be an example of ‘special case’ where the combination of these specific shift, grid size and initial position of the drop leads to numerical errors in the computation of the curvature and therefore in the surface tension term.



**Figure D.1:** Initialization of a 2D drop of diameter  $D = 1$  in a  $2 \times 2$  domain without contact (left figures) or with contact with the bottom boundary (right figures). From top to bottom, the shift in the x-direction is  $0$ ,  $\Delta/4$  and  $\Delta/2$ . The color map corresponds to the volume fraction  $c$ .

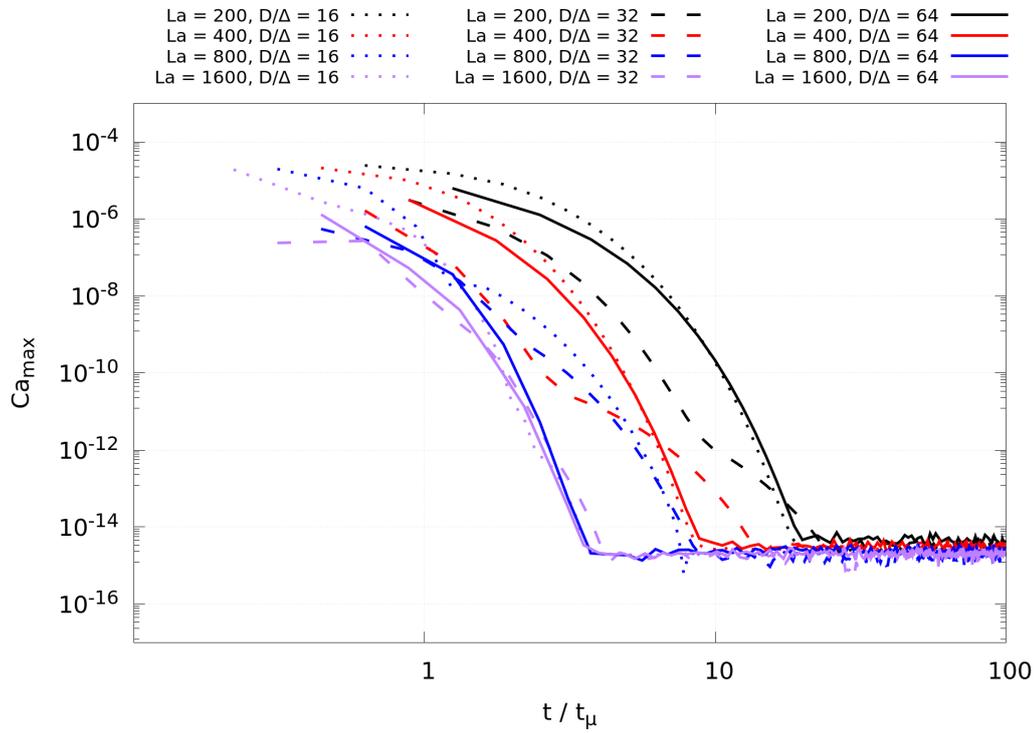
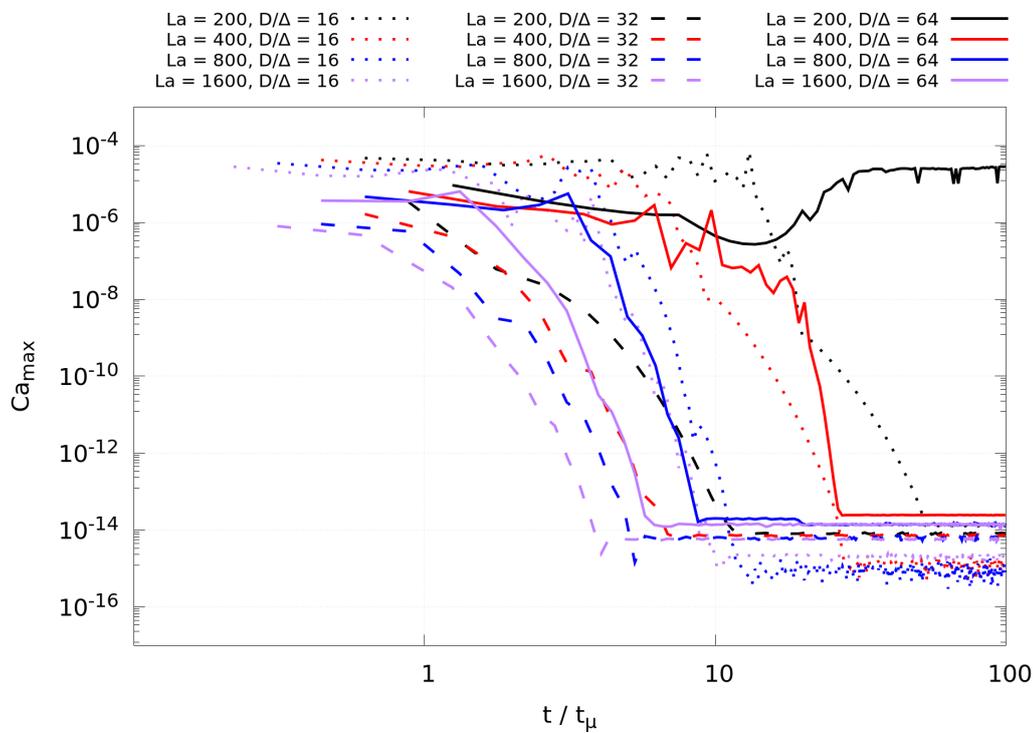


(a) No shift without contact.

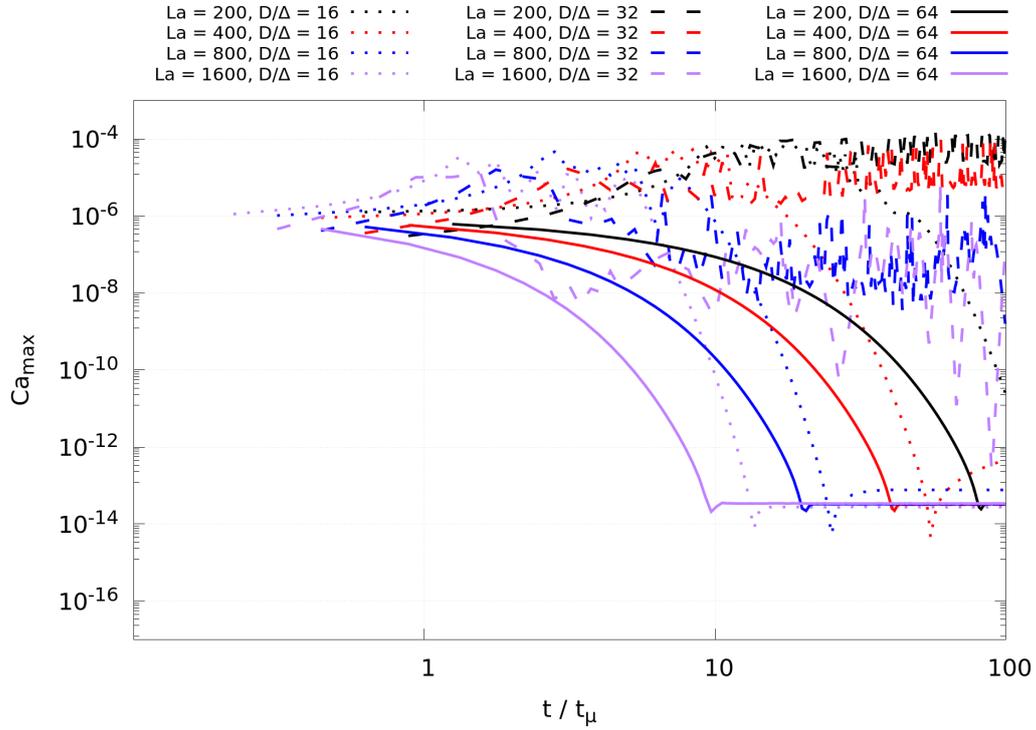
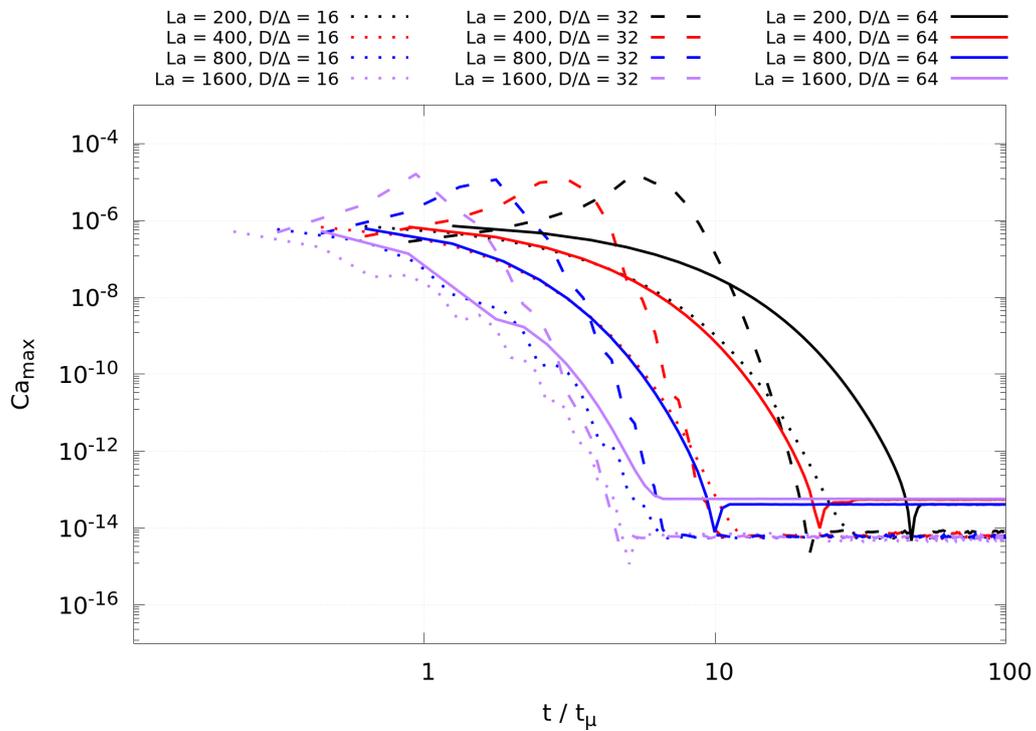


(b) No shift with contact.

**Figure D.2:** Evolution of the maximal dimensionless velocity as a function of time for the Laplace numbers and grid resolutions indicated for a static drop with no initial shift, and with or without contact.

(a)  $\Delta/4$  shift without contact.(b)  $\Delta/4$  shift with contact.

**Figure D.3:** Evolution of the maximal dimensionless velocity as a function of time for the Laplace numbers and grid resolutions indicated for a static drop with a  $\Delta/4$  initial shift, and with or without contact.

(a)  $\Delta/2$  shift without contact.(b)  $\Delta/2$  shift with contact.

**Figure D.4:** Evolution of the maximal dimensionless velocity as a function of time for the Laplace numbers and grid resolutions indicated for a static drop with a  $\Delta/2$  initial shift, and with or without contact.

# Numerical implementation of the super-slip model

---

We present here the implementation of the discrete super-slip model and its validation on a Poiseuille flow. Using the same notations than in Chapter 9 and recalling the super-slip boundary condition in continuous form

$$u_x - \lambda_1 \frac{\partial u_x}{\partial y} - \lambda_2 \frac{\partial^2 u_x}{\partial y^2} = U,$$

we can define the discrete equivalent as

$$\begin{aligned} \frac{u[ghost] + u[]}{2} + \lambda_1 \frac{u[ghost] - u[]}{\Delta} + \lambda_2 \frac{u[ghost] - 2u[] + u[1]}{\Delta^2} &= U \\ \Leftrightarrow u[ghost] &= \frac{2\Delta^2}{\Delta^2 + 2\Delta\lambda_1 + 2\lambda_2} U - \frac{\Delta^2 - 2\Delta\lambda_1 - 4\lambda_2}{\Delta^2 + 2\Delta\lambda_1 + 2\lambda_2} u[] \\ &\quad - \frac{2\lambda_2}{\Delta^2 + 2\Delta\lambda_1 + 2\lambda_2} u[1], \end{aligned}$$

with  $u_x[ghost]$  the tangential velocity at the ghost cell,  $u_x[]$  the tangential velocity of the cell inside the domain,  $u_x[1]$  the tangential velocity of the cell one grid point away from the wall and  $\Delta$  the grid spacing.

We validate the numerical implementation by considering a Poiseuille flow in a channel bounded by two walls  $y \in [-0.5, 0.5]$  where we apply a constant pressure gradient in the x-direction, such that

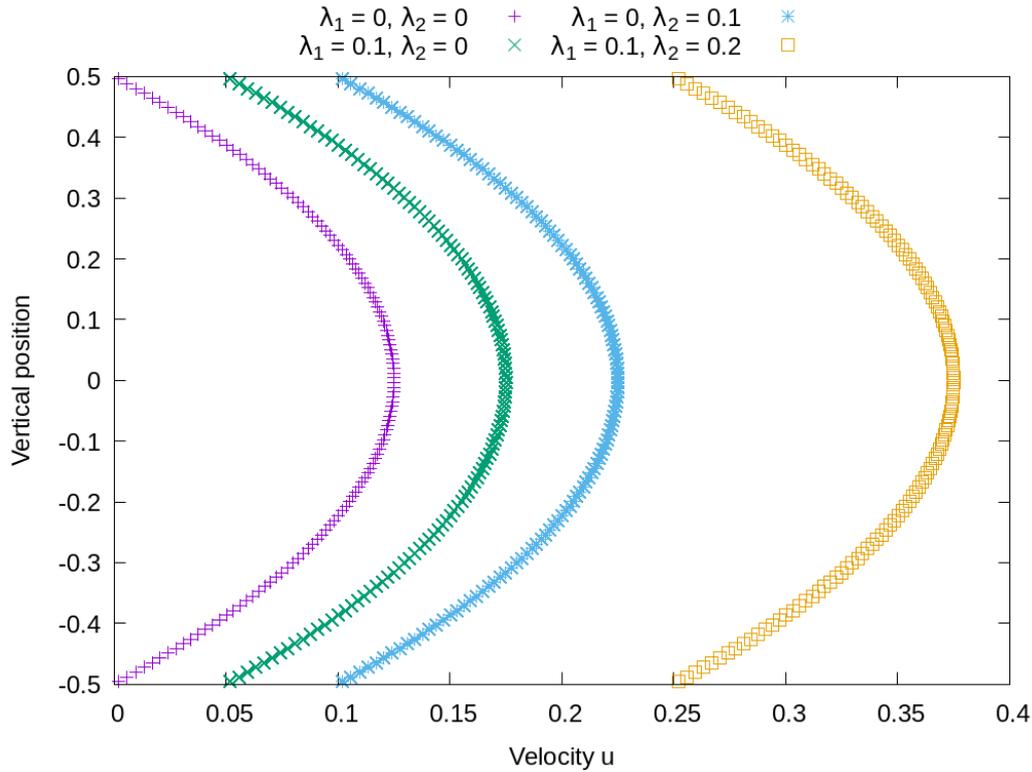
$$\frac{\partial^2 u}{\partial y^2} = -1.$$

By applying the super-slip boundary condition on both walls

$$\begin{aligned} u(-0.5) &= -\lambda_1 \frac{\partial u}{\partial y} \Big|_{y=-0.5} - \lambda_2 \frac{\partial^2 u}{\partial y^2} \Big|_{y=-0.5}, \\ u(0.5) &= -\lambda_1 \frac{\partial u}{\partial y} \Big|_{y=0.5} - \lambda_2 \frac{\partial^2 u}{\partial y^2} \Big|_{y=0.5}, \end{aligned}$$

we obtain the following analytical solution for the velocity field as function of  $y$

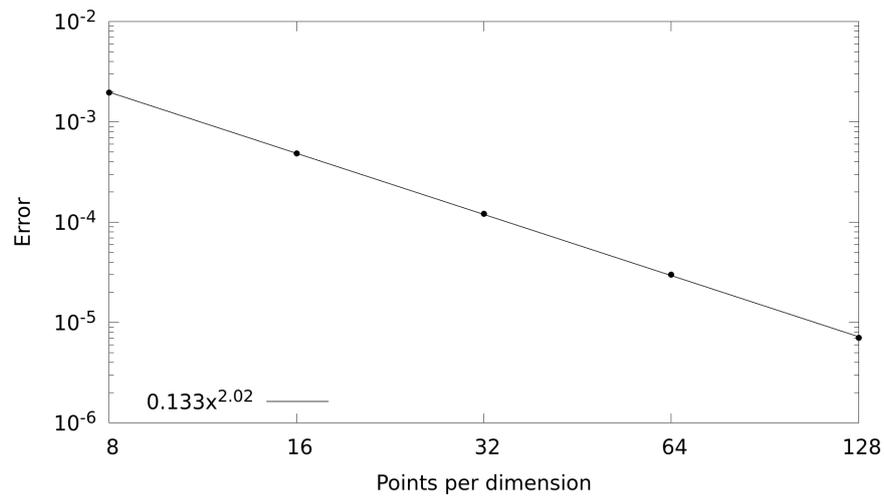
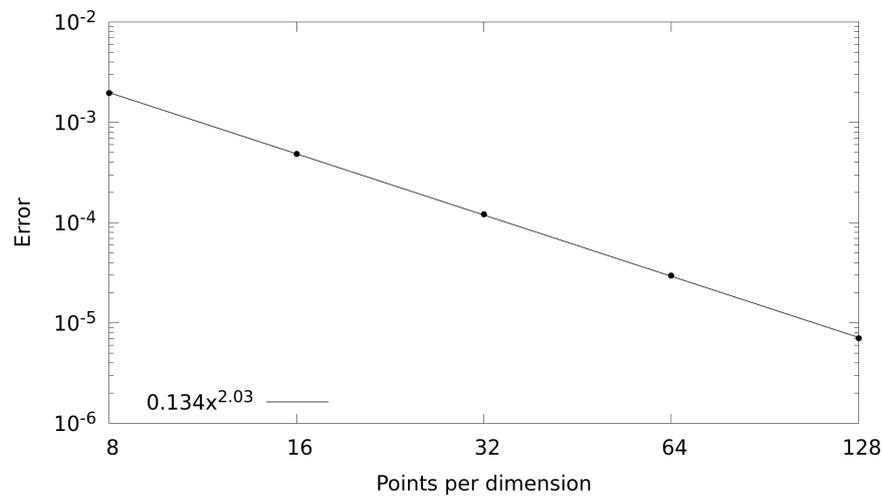
$$u(y) = \frac{1}{2} (0.25 + \lambda_1 + 2\lambda_2 - y^2).$$



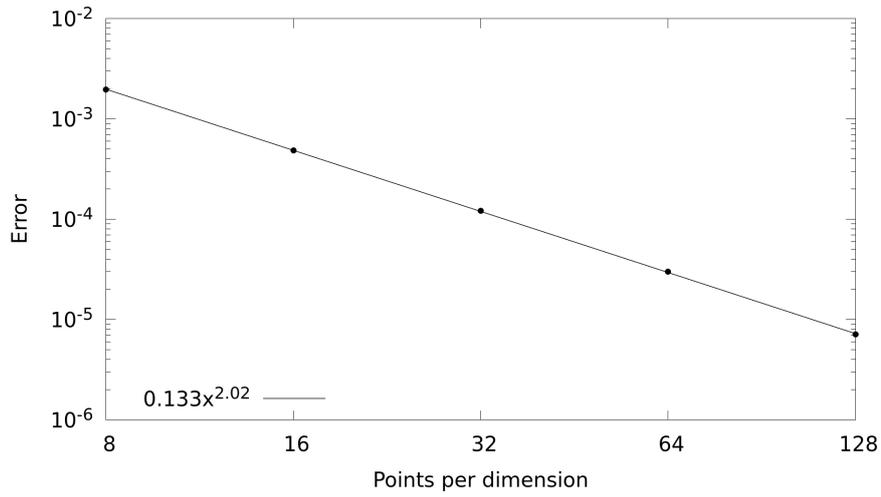
**Figure E.1:** Velocity profiles of a Poiseuille flow for the indicated values of  $\lambda_1$  and  $\lambda_2$ .

Figure E.1 is an example of different numerical velocity profiles obtained for  $\lambda_1 = 0, 0.1$  and  $\lambda_2 = 0, 0.1, 0.2$ .

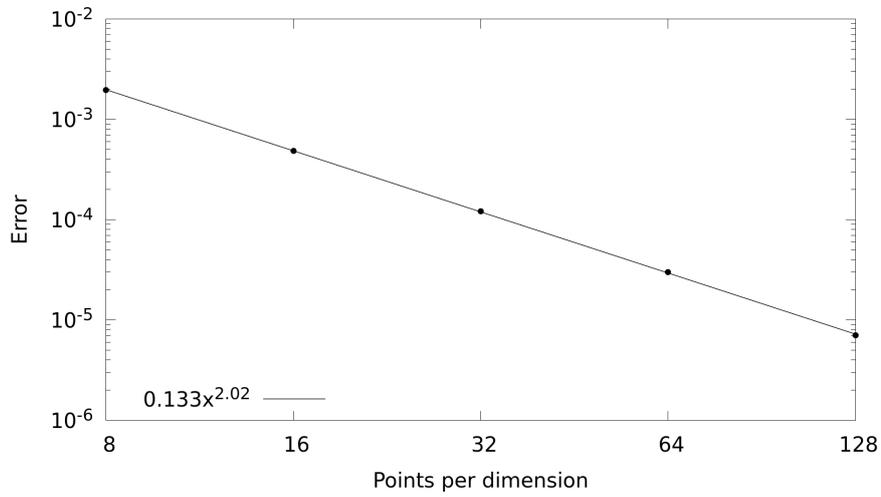
We carry out a convergence study by computing the L-2 norm of the error in velocity field for  $\lambda_1 = 0, \Delta$  and  $\lambda_2 = 0, \Delta^2$ . Figures E.2 and E.3 show a second-order convergence rate for all the cases indicating that the implemented super-slip boundary condition does not affect the convergence of the flow solver. The case  $\lambda_1 = \Delta, \lambda_2 = 0$  also validates the Navier boundary condition presented in Section 9.2.1.

(a)  $\lambda_1 = 0, \lambda_2 = 0$ (b)  $\lambda_1 = \Delta, \lambda_2 = 0$ 

**Figure E.2:** Validation of the super-slip implementation on a Poiseuille flow with  $\lambda_1 = 0, \Delta$  and  $\lambda_2 = 0$ .



(a)  $\lambda_1 = 0, \lambda_2 = \Delta^2$



(b)  $\lambda_1 = \Delta, \lambda_2 = \Delta^2$

**Figure E.3:** Validation of the super-slip implementation on a Poiseuille flow with  $\lambda_1 = 0, \Delta$  and  $\lambda_2 = \Delta^2$ .

# Kinematic transport of the contact angle

In this Section, we present a validation of the of the implementation of the free contact angle as defined in Equation 10.5. The idea is to obtain by extrapolation a numerical contact angle  $\theta_{\text{ext}}$  at the wall using an apparent angle  $\theta_{\text{app}}$  located one cell above the contact line. The dynamic contact angle  $\theta_d$  at the wall is then set to this extrapolated value. This method allows  $\theta_d$  to be a solution of the flow (instead of being imposed). Algorithm 7 summarizes the steps needed to compute  $\theta_{\text{ext}}$  at a given instant.

---

**Algorithm 7:** Angle extrapolation pseudo-code
 

---

**for each cell do**

Locate the cell one grid point above the contact line  
 Compute the apparent angle  $\theta_{\text{app}}$  using the unit normal  $\mathbf{n}$   
 Compute the first order derivative of the height function  $h_x$   
 Compute the interface curvature  $\kappa$   
 Compute the extrapolated angle

$$\theta_{\text{ext}} = \theta_{\text{app}} + \frac{3}{2} \Delta \frac{\kappa \sqrt{1 + h_y}}{\sin(\theta_{\text{app}})}$$

**end**

Apply the extrapolated angle at the wall through height functions

$$\theta_d = \theta_{\text{ext}}$$


---

From kinematic considerations, Fricke [Fricke 2020] derived an analytical solution for the transport of the contact angle in the case of an incompressible flow. Using this relation, we validate our extrapolation method in the VOF framework by carrying simulations of an oscillating drop for different grid sizes. We consider a drop of diameter  $D = 1$  in a  $2 \times 2$  domain that is initialized over a static substrate with a contact angle  $\theta_0 = 90^\circ$ . The velocity field in the whole domain, is set to

$$\begin{aligned} u_x &= a \cos(\pi t) x, \\ u_y &= -a \cos(\pi t) y, \end{aligned}$$

where  $a$  is a positive constant. In this study, we only consider the advection equation of the color function

$$\partial_t c + \nabla \cdot (c\mathbf{u}) = 0.$$

The prescribed incompressible velocity field will induce oscillations of the drop in both vertical and horizontal directions. The angle formed at the contact line will be affected by this motion and vary in time. We will compare the observed numerical contact angle with the analytical one  $\theta_{\text{exact}}$ , given by the following relation [Fricke 2020]

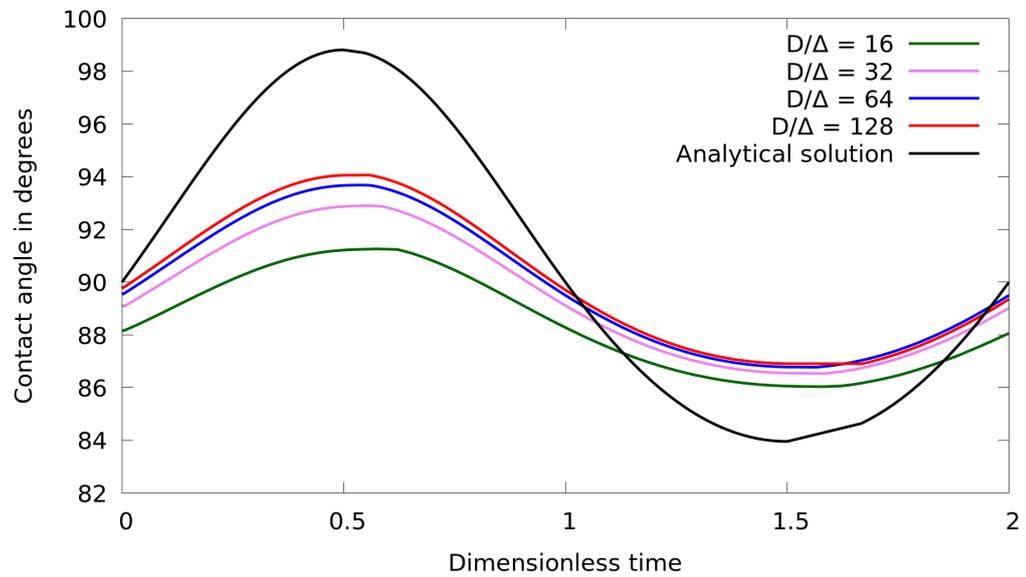
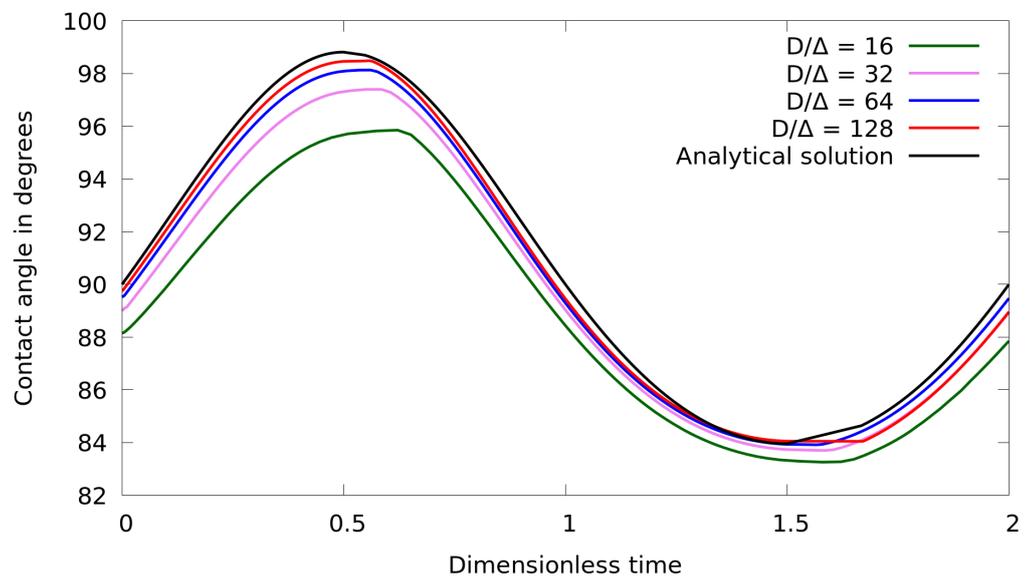
$$\theta_{\text{exact}} = \frac{\pi}{2} + \tan^{-1} \left( \frac{-1}{\tan \theta_0} e^{2aS} \right),$$

with

$$S = \frac{\sin(\pi t)}{\pi}.$$

We carry out two sets of simulations in order to assess the method. In the first one, the contact angle is constant and set to  $90^\circ$  whereas in the second one, we impose the extrapolated angle. The final time is set to  $t_f = 2$  and the grid sizes considered range from 16 to 128 points per diameter.

Figure F.1 summarizes the results obtained. The convergence of the contact angle with respect to the analytical solution is only obtained in the  $\theta_d = \theta_{\text{ext}}$  case thus validating the extrapolation method.

(a) Imposed  $90^\circ$  contact angle.(b)  $\theta_d = \theta_{\text{ext}}$ .

**Figure F.1:** Validation of the angle extrapolation method. The contact angle as a function of time is compared to the analytical formula for the indicated grids.



# Relaxation of the impermeability condition in the toy model

---

In the toy model (Section 10.3), a computational boundary is introduced at a distance  $+\delta$  from the wall where we apply a Navier boundary condition. When considering a spreading drop, the impermeability condition ( $u_y|_{y=\delta} = 0$ ) needs to be relaxed in order to take into account the mass flux through that computational boundary. To that end, we introduce a sink velocity  $v_{\text{sink}}$ , such that

$$u_y|_{y=\delta} = v_{\text{sink}}.$$

To calculate the amount of mass lost during one time step, we compute the area of the polygon formed by the two contact line points, at times  $t^{n-1}$  and  $t^n$ , and their linear projection onto the actual wall (at  $y = 0$ ). In practice, the angle used for the projection is an apparent angle located one cell above the contact line (similarly to the extrapolation angle method described in Appendix F). The computed area is then translated to the sink velocity by the simple formula

$$v_{\text{sink}} = \frac{1}{\tau} \frac{\mathcal{A}}{L},$$

where  $\tau$  is the time step,  $L$  the length of the wall and  $\mathcal{A}$  the signed area – defined later. Algorithm 8 summarizes the required steps at a given instant.

We validate the method by considering a drop spreading over a substrate for different values of  $\delta$ . The drop is initialized with a diameter  $D = 1$  in a  $2 \times 2$  domain and we apply a constant contact angle  $\theta_e = 70^\circ$ . The viscosities, densities and the surface tension are set to 1. The simulations are run until the equilibrium position is reached. In Figure G.1, we show the total mass as a function of time when applying the sink velocity for  $\delta = 0.2, 0.05, 0.02$ . The initial mass inside the drop decreases as spreading occurs. In Figure G.2, we compare the final interfaces for three cases: (i) toy model with sink velocity (ii) toy model without sink velocity (iii) full domain. From the results, we can observe the sink velocity allows us to match to the final true equilibrium shape (full domain). The error in final interface is a function of  $\delta$ . As  $\delta$  is increased, the linear extrapolation procedure – that does not take into account the curved interface at the contact line – induces a higher difference in final shapes. In practice, when using the toy model on a real setup,  $\delta$  is small compared to the domain size.

---

**Algorithm 8:** Sink velocity pseudo-code

---

**for each cell do**

Locate the cell one grid point above the contact line  
 Compute the contact line position  $x_{\text{CL}}^n$  at  $y = \delta + \Delta$   
 Compute the apparent angle  $\theta_{\text{app}}$  using the unit normal  $\mathbf{n}$   
 Project the contact line position

$$\tilde{x}_{\text{CL}}^n = x_{\text{CL}}^n + (\delta + \Delta) \tan\left(\frac{\pi}{2} - \theta_{\text{app}}\right)$$

Compute the polygon area  $\mathcal{A}$  formed by the points

$$(x_{\text{CL}}^n, \delta + \Delta), (\tilde{x}_{\text{CL}}^n, 0), (\tilde{x}_{\text{CL}}^{n-1}, 0), (x_{\text{CL}}^{n-1}, \delta + \Delta)$$

where  $x_{\text{CL}}^{n-1}$  and  $\tilde{x}_{\text{CL}}^{n-1}$  were stored in the previous time step

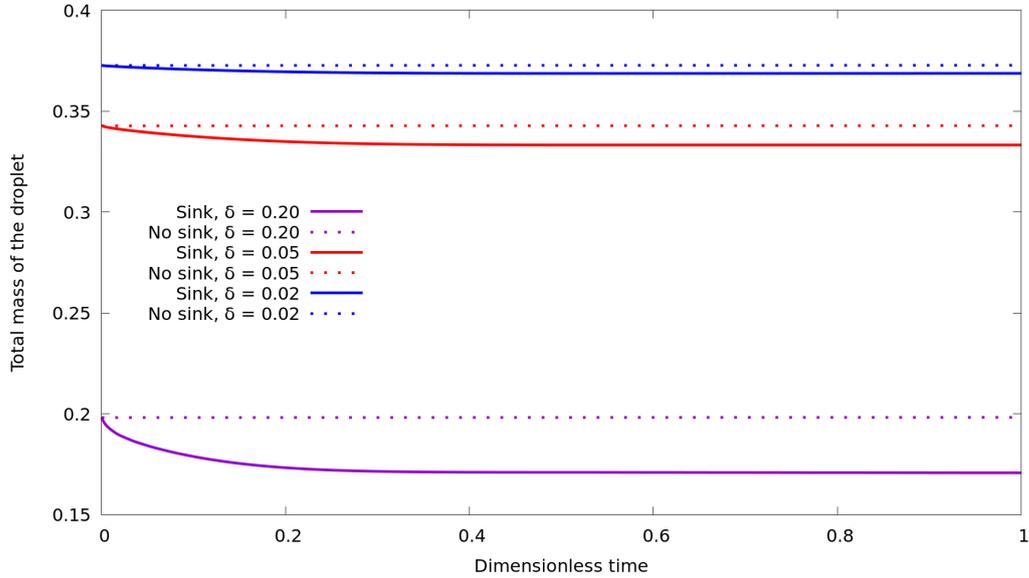
Determine the sink velocity

$$v_{\text{sink}} = \frac{1}{\tau} \frac{\mathcal{A}}{L}$$

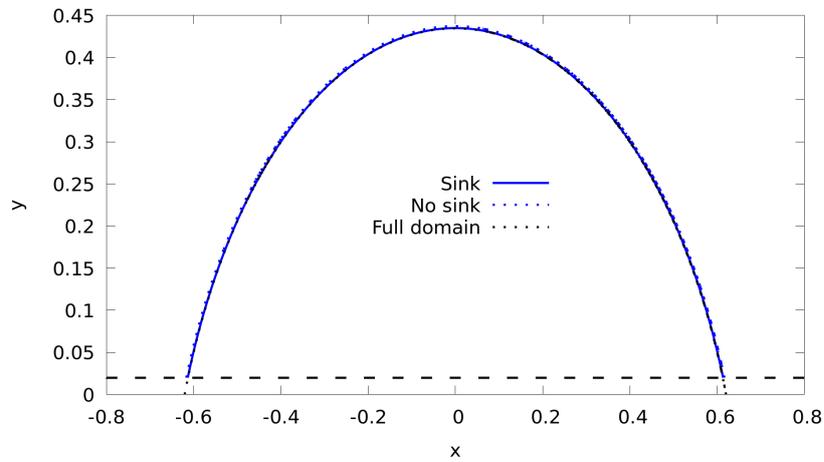
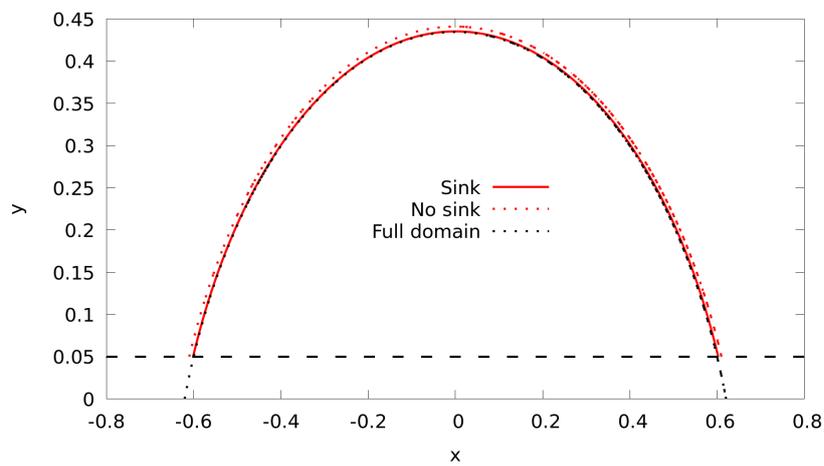
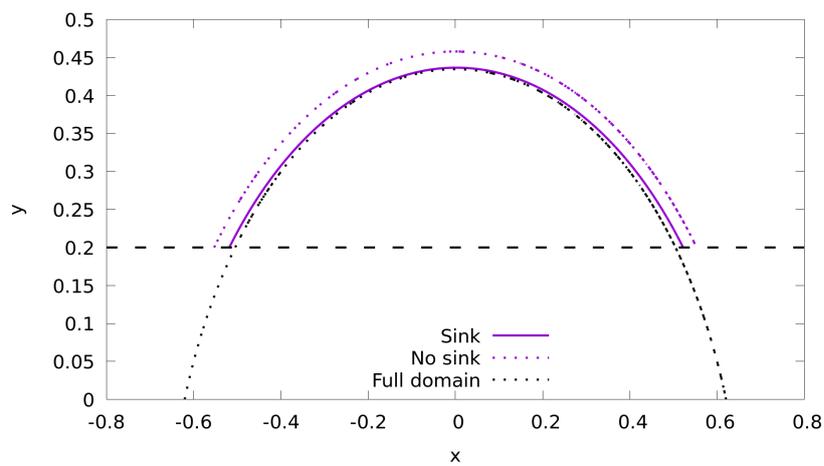
Store  $x_{\text{CL}}^{n-1} = x_{\text{CL}}^n$  and  $\tilde{x}_{\text{CL}}^{n-1} = \tilde{x}_{\text{CL}}^n$

**end**

---



**Figure G.1:** Total mass of the drop as a function of time with or without sink velocity for different computational boundaries placed at  $+\delta$  with  $\delta = 0.02, 0.05, 0.2$ .

(a)  $\delta = 0.02$ (b)  $\delta = 0.05$ (c)  $\delta = 0.2$ 

**Figure G.2:** Validation of the sink velocity imposed at the computational boundary placed at  $+\delta$  with  $\delta = 0.02, 0.05, 0.2$  denoted by the dotted black lines in the figures. The equilibrium angle is  $\theta_e = 70^\circ$  for the three cases considered.



# Bibliography

- [Abu-Al-Saud *et al.* 2018] M. O. Abu-Al-Saud, S. Popinet and H. A. Tchelepi. *A conservative and well-balanced surface tension model*. Journal of Computational Physics, vol. 371, pages 896–913, October 2018. (Cited on page 163.)
- [Afkhami & Bussmann 2008] S. Afkhami and M. Bussmann. *Height functions for applying contact angles to 2D VOF simulations*. Int. J. Numer. Meth. Fl., vol. 57, no. 4, pages 453–472, 2008. (Cited on pages 105, 106 and 111.)
- [Afkhami & Bussmann 2009] S. Afkhami and M. Bussmann. *Height functions for applying contact angles to 3D VOF simulations*. Int. J. Numer. Meth. Fl., vol. 61, no. 8, pages 827–847, 2009. (Cited on pages 105, 106 and 111.)
- [Afkhami *et al.* 2017] S. Afkhami, J. Buongiorno, A. Guion, S. Popinet, R. Scardovelli and S. Zaleski. *Transition in a numerical model of contact line dynamics and forced dewetting*. J. Comput. Phys., vol. 374, 2017. (Cited on pages 105, 109, 127, 130 and 145.)
- [Almgren 1993] R. Almgren. *Variational algorithms and pattern formation in dendritic solidification*. J Comp. Phys., vol. 106, no. 2, pages 337–354, 1993. (Cited on page 70.)
- [Amberg 2003] G. Amberg. *Semisharp Phase Field Method for Quantitative Phase Change Simulations*. Phys. Rev. Lett., vol. 91, Dec 2003. (Cited on pages 4 and 97.)
- [Arakawa & Lamb 1977] A. Arakawa and V. R. Lamb. *Computational Design of the Basic Dynamical Processes of the UCLA General Circulation Model*. In Methods in Computational Physics: Advances in Research and Applications, volume 17 of *Methods in Computational Physics: Advances in Research and Applications*, pages 173–265. Elsevier, 1977. (Cited on page 33.)
- [Arnold 1966] V. Arnold. *Sur la géométrie différentielle des groupes de Lie de dimension infinie et ses applications à l'hydrodynamique des fluides parfaits*. Annales de l'institut Fourier, vol. 16, pages 319–361, 1966. (Cited on page 154.)
- [Balay *et al.* 1997] S. Balay, W. D. Gropp, L. C. McInnes and B. F. Smith. *Efficient Management of Parallelism in Object Oriented Numerical Software Libraries*. In E. Arge, A. M. Bruaset and H. P. Langtangen, editors, Modern Software Tools in Scientific Computing, pages 163–202. Birkhäuser Press, 1997. (Cited on page 68.)
- [Bell *et al.* 1989] J. B. Bell, P. Colella and H. M. Glaz. *A second-order projection method for the incompressible navier-stokes equations*. Journal of Computational Physics, vol. 85, no. 2, pages 257–283, 1989. (Cited on page 108.)

- [Benzi *et al.* 2005] M. Benzi, G. H. Golub and J. Liesen. *Numerical solution of saddle point problems*. Acta Numerica, vol. 14, 5 2005. (Cited on page 157.)
- [Bernauer & Herzog 2011] M. K. Bernauer and R. Herzog. *Optimal Control of the Classical Two-Phase Stefan Problem in Level Set Formulation*. SIAM J. Sci. Comput., vol. 33, no. 1, pages 342–363, 2011. (Cited on pages 3, 10, 15, 22 and 151.)
- [Beyer & Leveque 2006] R. P. Beyer and R. J. Leveque. *Analysis of a One-Dimensional Model for the Immersed Boundary Method*. vol. 29, pages 332–364, 7 2006. (Cited on page 30.)
- [Bezanson *et al.* 2017] J. Bezanson, A. Edelman, S. Karpinski and V. B. Shah. *Julia: A Fresh Approach to Numerical Computing*. SIAM Review, vol. 59, no. 1, pages 65–98, 2017. (Cited on page 67.)
- [Blake & Shikhmurzaev 2002] T. D. Blake and Y. D. Shikhmurzaev. *Dynamic Wetting by Liquids of Different Viscosity*. Journal of colloid and interface science, vol. 253, pages 196–202, 2002. (Cited on pages 3, 103, 116 and 117.)
- [Blake *et al.* 1999] T. D. Blake, M. Bracke and Y. D. Shikhmurzaev. *Experimental evidence of nonlocal hydrodynamic influence on the dynamic contact angle*. Physics of Fluids, vol. 11, no. 8, pages 1995–2007, 1999. (Cited on pages 4, 95, 103, 113, 116 and 117.)
- [Bonn 2009] D. Bonn. *Wetting and spreading*. Reviews of Modern Physics, vol. 81, no. 2, pages 739–805, 2009. (Cited on page 135.)
- [Braman *et al.* 2015] K. Braman, T. A. Oliver and V. Raman. *Adjoint-based sensitivity analysis of flames*. Comb. Theo. Modelling, vol. 19, no. 1, pages 29–56, 2015. (Cited on page 2.)
- [Brattkus & Meiron 1992] K. Brattkus and D. I. Meiron. *Numerical simulations of unsteady crystal growth*. SIAM J. Appl. Math., vol. 52, page 1303, 1992. (Cited on page 1.)
- [Brown *et al.* 2001] D. L. Brown, R. Cortez and M. L. Minion. *Accurate Projection Methods for the Incompressible Navier–Stokes Equations*. Journal of Computational Physics, vol. 168, 4 2001. (Cited on page 157.)
- [Cahn & Hilliard 1958] J. W. Cahn and J. E. Hilliard. *Free Energy of a Nonuniform System. I. Interfacial Free Energy*. The Journal of Chemical Physics, vol. 28, no. 2, pages 258–267, 1958. (Cited on page 98.)
- [Calhoun & Leveque 2000] D. Calhoun and R. J. Leveque. *A Cartesian Grid Finite-Volume Method for the Advection-Diffusion Equation in Irregular Geometries*. Journal of Computational Physics, vol. 157, 1 2000. (Cited on page 30.)

- [Carlson 2012] A. Carlson. *Capillarity and dynamic wetting*. PhD thesis, KTH Royal Institute of Technology, 2012. (Cited on pages 4 and 97.)
- [Chen *et al.* 1997] S. Chen, B. Merriman, S. Osher and P. Smereka. *A Simple Level Set Method for Solving Stefan Problems*. J Comp. Phys., vol. 135, no. 1, pages 8–29, 1997. (Cited on pages 8, 10, 75, 76 and 88.)
- [Cheny & Botella 2010] Y. Cheny and O. Botella. *The LS-STAG method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties*. Journal of Computational Physics, vol. 229, pages 1043–1076, 2010. (Cited on pages 30 and 154.)
- [Chorin 1968] A. J. Chorin. *Numerical Solution of the Navier-Stokes Equations*. American Mathematical Society, vol. 22, 1968. (Cited on page 157.)
- [Cox 1986] R. G. Cox. *The dynamics of the spreading of liquids on a solid surface. Part 1. Viscous flow*. J. Fluid Mech., vol. 168, pages 169–194, 1986. (Cited on pages 4, 103 and 110.)
- [Crank 1987] J. Crank. *Free and Moving Boundary Problems*. Oxford science publications. Clarendon Press, 1987. (Cited on page 68.)
- [Devauchelle *et al.* 2007] O. Devauchelle, C. Josserand and S. Zaleski. *Forced dewetting on porous media*. Journal of Fluid Mechanics, vol. 574, page 343–364, 2007. (Cited on pages 4, 102 and 121.)
- [Dupont & Legendre 2010] J.-B. Dupont and D. Legendre. *Numerical simulation of static and sliding drop with contact angle hysteresis*. Journal of Computational Physics, vol. 229, no. 7, pages 2453–2478, 2010. (Cited on page 119.)
- [Duraisamy & Alonso 2012] K. Duraisamy and J. Alonso. *Adjoint-based techniques for uncertainty quantification in turbulent flows with combustion*. 42nd AIAA Fluid Dynamics Conference and Exhibit, pages 25–28, 2012. (Cited on page 2.)
- [Eggers & Evans 2005] J. Eggers and R. Evans. *Comment on "Dynamic Wetting by liquids of different viscosity", by T. D. Blake and Y. D. Shikhmurzaev*. Journal of colloid and interface science, vol. 280, pages 537–8; discussion 539, 2005. (Cited on pages 4, 103 and 117.)
- [Eggers 2004] J. Eggers. *Toward a description of contact line motion at higher capillary numbers*. Physics of Fluids, vol. 16, no. 9, pages 3491–3494, 2004. (Cited on pages 4, 95 and 103.)
- [Favier *et al.* 2019] B. Favier, J. Purseed and L. Duchemin. *Rayleigh–Bénard convection with a melting boundary*. J. Fluid Mech., vol. 858, pages 437–473, January 2019. (Cited on pages 3, 12 and 81.)

- [Fikl *et al.* 2020] A. Fikl, V. Le Chenadec and T. Sayadi. *Control and Optimization of Interfacial Flows Using Adjoint-Based Techniques*. Fluids, vol. 5, no. 3, 2020. (Cited on pages 2 and 10.)
- [Foures *et al.* 2014] D. Foures, C. Caulfield and P. Schmid. *Optimal mixing in two-dimensional plane Poiseuille flow at finite Péclet number*. J Fluid Mech., vol. 748, pages 241–277, 2014. (Cited on page 2.)
- [Frank 1950] F. C. Frank. *Radially symmetric phase growth controlled by diffusion*. Proc. R. Soc. Lond, 1950. (Cited on page 69.)
- [Fricke *et al.* 2018] M. Fricke, M. Köhne and D. Bothe. *On the Kinematics of Contact Line Motion*. Proc. Appl. Math. Mech., vol. 18, no. 1, 2018. (Cited on page 126.)
- [Fricke *et al.* 2019] M. Fricke, M. Köhne and D. Bothe. *A kinematic evolution equation for the dynamic contact angle and some consequences*. Physica D: Nonl. Phen., vol. 394, pages 26–43, 2019. (Cited on pages 103 and 126.)
- [Fricke 2020] M. Fricke. *Mathematical modeling and Volume-of-Fluid based simulation of dynamic wetting*. PhD thesis, 2020. (Cited on pages 4, 126, 173 and 174.)
- [Fujioka 1978] T. Fujioka. PhD thesis, Carnegie-MeBon University, 1978. (Cited on page 9.)
- [Fullana *et al.* 2020] T. Fullana, S. Zaleski and S. Popinet. *Dynamic wetting failure in curtain coating by the Volume-of-Fluid method - Volume-of-Fluid simulations on quadtree meshes*. Eur. Phys. J. Special Topics, vol. 229, no. 10, pages 1923–1934, 2020. (Cited on pages 4, 95, 102, 109, 111 and 145.)
- [Fullana *et al.* 2023] T. Fullana, V. Le Chenadec and T. Sayadi. *Adjoint-based optimization of two-dimensional Stefan problems*. Journal of Computational Physics, vol. 475, page 111875, 2023. (Cited on page 3.)
- [Fullana 2019] T. Fullana. Study of the dynamic contact line by volume of fluid method on hierarchical adaptive mesh. Master’s thesis, Sorbonne Université, 2019. (Cited on pages 144 and 145.)
- [Giles & Pierce 2000] M. B. Giles and N. A. Pierce. *An Introduction to the Adjoint Approach to Design*. Flow Turb. Combust., vol. 65, no. 3, pages 393–415, 2000. (Cited on page 2.)
- [Goldstein *et al.* 1993] D. Goldstein, R. Handler and L. Sirovich. *Modeling a No-Slip Flow Boundary with an External Force Field*. Journal of Computational Physics, vol. 105, 4 1993. (Cited on page 30.)

- [Harlow & Welch 1965] F. H. Harlow and J. E. Welch. *Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface*. Physics of Fluids, vol. 8, no. 12, pages 2182–2189, 1965. (Cited on page 33.)
- [Hassan *et al.* 2021] A. Hassan, T. Sayadi, V. Le Chenadec, H. Pitsch and A. Attili. *Adjoint-Based Sensitivity Analysis of Steady Char Burn out*. Com. Theo. Modeling, vol. 25, no. 1, pages 96–120, 2021. (Cited on pages 1 and 3.)
- [Hecht 2012] F. Hecht. *New development in FreeFem++*. J. Numer. Math., vol. 20, no. 3-4, pages 251–265, 2012. (Cited on page 141.)
- [Hinze & Ziegenbalg 2007] M. Hinze and S. Ziegenbalg. *Optimal control of the free boundary in a two-phase Stefan problem*. J Comput. Phys., vol. 223, no. 2, pages 657–684, 2007. (Cited on page 3.)
- [Hoffmann & Sprekels 1982] K. H. Hoffmann and J. Sprekels. *Real-time control of the free boundary in a two-phase Stefan problem*. Numerical Functional Analysis and Optimization, vol. 5, no. 1, pages 47–76, 1982. (Cited on page 3.)
- [Huh & Scriven 1971] C. Huh and L. E. Scriven. *Hydrodynamic model of steady movement of a solid/liquid/fluid contact line*. J. Colloid and Interface Sci., vol. 35, pages 85–101, 1971. (Cited on pages 3, 102 and 135.)
- [Jacqmin 1999] D. Jacqmin. *Calculation of two-phase Navier–Stokes flows using phase-field modeling*. J. Comput. Phys., vol. 155, no. 1, pages 96–127, 1999. (Cited on page 98.)
- [Jacqmin 2000] D. Jacqmin. *Contact-line dynamics of a diffuse fluid interface*. J. Fluid Mech., vol. 402, pages 57–88, 2000. (Cited on pages 4, 97 and 98.)
- [Jameson *et al.* 1998] A. Jameson, L. Martinelli and N. A. Pierce. *Optimum Aerodynamic Design Using the Navier-Stokes Equations*. Theor. Comp. Fluid Dyn., vol. 10, no. 1, pages 213–237, 1998. (Cited on page 2.)
- [Jameson 1988] A. Jameson. *Aerodynamic design via control theory*. J Scientific Computing, vol. 3, no. 3, pages 233–260, 1988. (Cited on page 2.)
- [Javierre *et al.* 2006] E. Javierre, C. Vuik, F. J. Vermolen and S. van der Zwaag. *A comparison of numerical models for one-dimensional Stefan problems*. J Comp. Applied Mathematics, vol. 192, no. 2, pages 445–459, 2006. (Cited on page 1.)
- [Johansen & Colella 1998] H. Johansen and P. Colella. *A Cartesian Grid Embedded Boundary Method for Poisson’s Equation on Irregular Domains*. J Comp. Phys., vol. 147, no. 1, pages 60–85, 1998. (Cited on page 53.)

- [Juniper 2010] M. Juniper. *Triggering in the horizontal Rijke tube: non-normality, transient growth and bypass transition*. J. Fluid Mech., vol. 667, pages 272–308, 2010. (Cited on page 2.)
- [Juric & Tryggvason 1996] D. Juric and G. Tryggvason. *A Front-Tracking Method for Dendritic Solidification*. J Comp. Phys., vol. 123, no. 1, pages 127–148, 1996. (Cited on pages 1, 9 and 76.)
- [Kang & Zabaras 1995] S. Kang and N. Zabaras. *Control of the freezing interface motion in two-dimensional solidification processes using the adjoint method*. International J Numerical Methods in Engineering, vol. 38, pages 63–80, 1995. (Cited on page 3.)
- [Knabner 1985] P. Knabner. *Control of Stefan problems by means of linear-quadratic defect minimization*. Numerische Mathematik, vol. 46, no. 3, pages 429–442, 1985. (Cited on page 3.)
- [Lācis *et al.* 2020] U. Lācis, P. Johansson, T. Fullana, B. Hess, G. Amberg, S. Bagheri and S. Zaleski. *Steady moving contact line of water over a no-slip substrate: Challenges in benchmarking phase-field and volume-of-fluid methods against molecular dynamics simulations*. Eur. Phys. J. Spec. Top., vol. 229, no. 10, pages 1897–1921, September 2020. (Cited on pages 4, 95, 102, 109, 110 and 145.)
- [Lācis *et al.* 2022] U. Lācis, M. Pellegrino, J. Sundin, G. Amberg, S. Zaleski, B. Hess and S. Bagheri. *Nanoscale sheared droplet: volume-of-fluid, phase-field and no-slip molecular dynamics*. Journal of Fluid Mechanics, vol. 940, page A10, 2022. (Cited on pages 4, 95 and 109.)
- [Lai & Peskin 2000] M. C. Lai and C. S. Peskin. *An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity*. Journal of Computational Physics, vol. 160, 5 2000. (Cited on page 30.)
- [Langer 1980] J. S. Langer. *Instabilities and pattern formation in crystal growth*. Reviews of Modern Physics, 1980. (Cited on pages 1, 8 and 72.)
- [Lavi & Marmur 2004] B. Lavi and A. Marmur. *The exponential power law: partial wetting kinetics and dynamic contact angles*. Colloids and Surfaces A: Physicochemical and Engineering Aspects, vol. 250, no. 1, pages 409–414, 2004. (Cited on pages 4 and 119.)
- [Legendre & Maglio 2015] D. Legendre and M. Maglio. *Comparison between numerical models for the simulation of moving contact lines*. Computers & Fluids, vol. 113, pages 2–13, May 2015. (Cited on pages 4, 103, 109, 119 and 145.)
- [Legendre 2013] D. Legendre. *Numerical simulation of spreading drops*. Colloids Surf, vol. 432, page 29–37, 09 2013. (Cited on page 109.)

- [Lemke *et al.* 2013] M. Lemke, J. Reiss and J. Sesterhenn. *Adjoint-based analysis of thermoacoustic coupling*. ICNAAM, pages 2163–2166, 2013. (Cited on page 2.)
- [Lemke *et al.* 2019] M. Lemke, L. Cai, J. Reiss, H. Pitsch and J. Sesterhenn. *Adjoint-based sensitivity analysis of quantities of interest of complex combustion models*. Comb. Theo. Modelling, vol. 23, no. 1, pages 180–196, 2019. (Cited on page 2.)
- [Limare *et al.* 2023] A. Limare, S. Popinet, C. Josserand, Z. Xue and A. Ghigo. *A hybrid level-set / embedded boundary method applied to solidification-melt problems*. Journal of Computational Physics, vol. 474, page 111829, 2023. (Cited on pages 1, 3, 10, 62, 68, 78 and 81.)
- [Liu. & Nocedal 1989] D. C Liu. and J. Nocedal. *On the Limited Memory Method for Large Scale Optimization*. Mathematical Programming, vol. 3, no. 45, page 503–528, 1989. (Cited on page 26.)
- [Liu *et al.* 2016] C.-Y. Liu, E. Vandre, M. S. Carvalho and S. Kumar. *Dynamic wetting failure and hydrodynamic assist in curtain coating*. Journal of Fluid Mechanics, vol. 808, page 290–315, 2016. (Cited on pages 102, 111, 113, 115 and 116.)
- [Liu *et al.* 2018] C.-Y. Liu, M. S. Carvalho and S. Kumar. *Dynamic wetting failure in curtain coating: Comparison of model predictions and experimental observations*. Chemical Engineering Science, vol. 195, 2018. (Cited on pages 102 and 117.)
- [Marsden *et al.* 2008] A. L. Marsden, J. A. Feinstein and C. A. Taylor. *A computational framework for derivative-free optimization of cardiovascular geometries*. Computer Methods in Applied Mechanics and Engineering, vol. 197, no. 21-24, pages 1890–1905, 2008. (Cited on page 2.)
- [Marston *et al.* 2009] J. O. Marston, V. Hawkins, S. P. Decent and M. J. H. Simmons. *Influence of surfactant upon air entrainment hysteresis in curtain coating*. Experiments in Fluids, vol. 46, no. 3, pages 549–558, Mar 2009. (Cited on pages 3, 113, 116 and 117.)
- [Mavriplis 1995] D. J. Mavriplis. *Unstructured mesh generation and adaptivity*. Technical report, 1995. (Cited on page 30.)
- [Mikula & Ohlberger 2010] K. Mikula and M. Ohlberger. *A New Level Set Method for Motion in Normal Direction Based on a Semi-Implicit Forward-Backward Diffusion Approach*. SIAM J. Sci. Comput., vol. 32, no. 3, pages 1527–1544, 2010. (Cited on pages 56 and 58.)

- [Mikula *et al.* 2014] K. Mikula, M. Ohlberger and J. Urbán. *Inflow-implicit/outflow-explicit finite volume methods for solving advection equations*. Applied Numerical Mathematics, vol. 85, pages 16–37, 2014. (Cited on pages 56 and 57.)
- [Min 2010] C. Min. *On reinitializing level set functions*. J Comp. Phys., vol. 229, no. 8, pages 2764–2772, April 2010. (Cited on pages 62 and 63.)
- [Mogensen & Riseth 2018] P. K. Mogensen and A. N. Riseth. *Optim: A mathematical optimization package for Julia*. Journal of Open Source Software, vol. 3, no. 24, page 615, 2018. (Cited on page 68.)
- [Morinishi *et al.* 1998] Y. Morinishi, T. S. Lund, O. V. Vasilyev and P. Moin. *Fully Conservative Higher Order Finite Difference Schemes for Incompressible Flow*. Journal of Computational Physics, vol. 143, no. 1, pages 90–124, 1998. (Cited on pages 30 and 31.)
- [Morinishi 2010] Y. Morinishi. *Skew-symmetric form of convective terms and fully conservative finite difference schemes for variable density low-Mach number flows*. Journal of Computational Physics, vol. 229, no. 2, pages 276–300, 2010. (Cited on pages 30, 155 and 156.)
- [Mullins & Sekerka 1964] W. W. Mullins and R. F. Sekerka. *Stability of a Planar Interface During Solidification of a Dilute Binary Alloy*. Journal of Applied Physics, vol. 35, no. 2, pages 444–451, 1964. Publisher: American Institute of Physics. (Cited on pages 1, 70 and 88.)
- [Nocedal & Wright 2006] J. Nocedal and S. J. Wright. Numerical optimization. Springer series in operation research and financial engineering. Springer, 2nd édition, 2006. (Cited on page 157.)
- [Osher & Sethian 1988] S. Osher and J. A. Sethian. *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*. J Comput. Phys., vol. 79, no. 1, pages 12–49, 1988. (Cited on page 1.)
- [Ou & Jameson 2011] K. Ou and A. Jameson. *Unsteady Adjoint Method for the Optimal Control of Advection and Burger’s Equations Using High-Order Spectral Difference Method*. American Institute of Aeronautics and Astronautics, pages 1–18, 2011. (Cited on page 2.)
- [Pandey *et al.* 2018] A. Pandey, J. D. Scheel and J. Schumacher. *Turbulent superstructures in Rayleigh-Bénard convection*. Nat Commun, vol. 9, no. 1, page 2118, May 2018. Number: 1 Publisher: Nature Publishing Group. (Cited on page 12.)
- [Peng *et al.* 1999] D. Peng, B. Merriman, S. Osher, H. Zhao and M. Kang. *A PDE-Based Fast Local Level Set Method*. J Comp. Phys., vol. 155, no. 2, pages 410–438, 1999. (Cited on page 54.)

- [Perot 2011] J. B. Perot. *Discrete Conservation Properties of Unstructured Mesh Schemes*. Annual Review of Fluid Mechanics, vol. 43, pages 299–318, jan 2011. (Cited on page 154.)
- [Peskin 1972] C. S. Peskin. *Flow patterns around heart valves: A numerical method*. Journal of Computational Physics, vol. 10, 10 1972. (Cited on page 30.)
- [Peskin 1977] C. S. Peskin. *Numerical analysis of blood flow in the heart*. Journal of Computational Physics, 11 1977. (Cited on page 30.)
- [Peskin 2003] C. S. Peskin. *The immersed boundary method*. Acta Numerica, vol. 11, 7 2003. (Cited on page 30.)
- [Pierret *et al.* 2007] S. Pierret, R. F. Coelho and H. Kato. *Multidisciplinary and multiple operating points shape optimization of three-dimensional compressor blades*. Structural and Multidisciplinary Optimization, vol. 33, no. 1, pages 61–70, 2007. (Cited on page 2.)
- [Pironneau 1974] O. Pironneau. *On optimum design in fluid mechanics*. J Fluid Mech., vol. 64, no. 1, pages 97–110, 1974. (Cited on page 2.)
- [Popinet & Zaleski 1999] S. Popinet and S. Zaleski. *A front-tracking algorithm for accurate representation of surface tension*. Int. J. Numer. Methods Fluids, vol. 30, no. 6, pages 775–793, 1999. (Cited on pages 105 and 107.)
- [Popinet 2003] S. Popinet. *Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries*. Journal of Computational Physics, vol. 190, no. 2, pages 572–600, September 2003. (Cited on page 108.)
- [Popinet 2009] S. Popinet. *An Accurate Adaptive Solver for Surface-tension-driven Interfacial Flows*. J. Comput. Phys., vol. 228, no. 16, pages 5838–5866, 2009. (Cited on pages 64, 105 and 107.)
- [Popinet 2015] S. Popinet. *A quadtree-adaptive multigrid solver for the Serre–Green–Naghdi equations*. J. Comput. Phys., vol. 302, pages 336–358, 2015. (Cited on pages 105 and 107.)
- [Popinet 2018] S. Popinet. *Numerical Models of Surface Tension*. Annu. Rev. Fluid Mech., vol. 50, no. 1, pages 49–75, 2018. (Cited on pages 95, 105, 107 and 163.)
- [Qian *et al.* 2003a] T.-Z. Qian, X.-P. Wang and P. Sheng. *Generalized Navier Boundary Condition for the Moving Contact Line*. Commun. Math. Sci., vol. 1, no. 2, pages 333–341, 06 2003. (Cited on page 117.)
- [Qian *et al.* 2003b] T.-Z. Qian, X.-P. Wang and P. Sheng. *Molecular scale contact line hydrodynamics of immiscible flows*. Phys. Rev. E, vol. 68, no. 1, page 016306, 2003. (Cited on pages 4, 98, 103 and 126.)

- [Qian *et al.* 2006] T.-Z. Qian, X.-P. Wang and P. Sheng. *A variational approach to moving contact line hydrodynamics*. *J. Fluid Mech.*, vol. 564, pages 333–28, 2006. (Cited on pages 4, 103 and 126.)
- [Quirós Rodríguez *et al.* 2022] A. Quirós Rodríguez, T. Fullana, V. Le Chenadec and T. Sayadi. *A Conservative Cartesian Cut Cell Method for the Solution of the Incompressible Navier-Stokes Equations on Staggered Meshes*. in press, 2022. (Cited on pages 3 and 153.)
- [Rabin *et al.* 2014] S. Rabin, C. Caulfield and R. Kerswell. *Designing a more nonlinearly stable laminar flow via boundary manipulation*. *J Fluid Mech.*, vol. 738, pages R1:1–12, 2014. (Cited on page 2.)
- [Rose 1993] M. E. Rose. *An enthalpy scheme for Stefan problems in several dimensions*. *App. Numerical Math.*, vol. 12, page 229, 1993. (Cited on page 1.)
- [Scardovelli & Zaleski 1999] R. Scardovelli and S. Zaleski. *Direct numerical simulation of free-surface and interfacial flow*. *Annu. Rev. Fluid Mech.*, vol. 31, no. 1, pages 567–603, 1999. (Cited on pages 95, 105 and 107.)
- [Schmidt *et al.* 2013] S. Schmidt, C. Ilic, V. Schulz and N. R. Gauger. *Three-dimensional large-scale aerodynamic shape optimization based on shape calculus*. *AIAA J*, vol. 51, no. 11, pages 2615–2627, 2013. (Cited on page 2.)
- [Segal *et al.* 1998] G. Segal, C. Vuik and F. Vermolen. *A conserving discretization for the free boundary in a two-dimensional Stefan problem*. *J Comput. Phys.*, vol. 141, pages 1–21, 1998. (Cited on page 1.)
- [Sherer & Scott 2005] S. E. Sherer and J. N. Scott. *High-order compact finite-difference methods on general overset grids*. *Journal of Computational Physics*, vol. 210, pages 459–496, 12 2005. (Cited on page 30.)
- [Shikhmurzaev 1994] Y. D. Shikhmurzaev. *Mathematical modeling of wetting hydrodynamics*. *Fluid Dyn. Res.*, vol. 13, no. 1, pages 45–64, jan 1994. (Cited on pages 4, 95, 103 and 117.)
- [Snoeijer & Andreotti 2013] J. H. Snoeijer and B. Andreotti. *Moving Contact Lines: Scales, Regimes, and Dynamical Transitions*. *Annual Review of Fluid Mechanics*, vol. 45, no. 1, pages 269–292, 2013. (Cited on page 135.)
- [Stefan 1891] J. Stefan. *Über die Theorie der Eisbildung, insbesondere über die Eisbildung im Polarmeere*. *Ann. Physik Chemie*, vol. 42, pages 269–286, 1891. (Cited on page 1.)
- [Sui *et al.* 2014] Y. Sui, H. Ding and P. D. M. Spelt. *Numerical Simulations of Flows with Moving Contact Lines*. *Annu. Rev. Fluid Mech.*, vol. 46, no. 1, pages 97–119, 2014. (Cited on page 109.)

- [Tan & Zabaras 2006] L. Tan and N. Zabaras. *A level set simulation of dendritic solidification with combined features of front-tracking and fixed-domain methods*. Journal of Computational Physics, vol. 211, no. 1, pages 36–63, 2006. (Cited on page 78.)
- [Tryggvason *et al.* 2011] G. Tryggvason, R. Scardovelli and S. Zaleski. Direct numerical simulations of gas–liquid multiphase flows. Cambridge University Press, 2011. (Cited on pages 95 and 105.)
- [Voinov 1976] O. Voinov. *Hydrodynamics of wetting*. Fluid Dyn., vol. 11, page 714, 1976. (Cited on pages 4 and 110.)
- [Wilson *et al.* 2006] M. Wilson, J. Summers, Y. D. Shikhmurzaev, A. Clarke and T. D. Blake. *Nonlocal hydrodynamic influence on the dynamic contact angle: Slip models versus experiment*. Physical review. E, Statistical, nonlinear, and soft matter physics, vol. 73, page 41606, 2006. (Cited on pages 3, 95, 102 and 117.)
- [Woods 1992] A. W. Woods. *Melting and dissolving*. J. Fluid Mech., vol. 239, page 429, 1992. (Cited on page 1.)
- [Wright & Nocedal 2006] S. J. Wright and J. Nocedal. Numerical optimization. Springer, 2006. (Cited on page 26.)
- [Yang 1997] Z. Yang. *The Adjoint Method for the Inverse Design of Solidification Processes with Convection*. PhD thesis, Cornell University, 1997. (Cited on page 3.)
- [Yue & Feng 2011] P. Yue and J. J. Feng. *Wall energy relaxation in the Cahn–Hilliard model for moving contact lines*. Physics of Fluids, vol. 23, no. 1, 2011. (Cited on pages 135 and 141.)
- [Yue *et al.* 2007] P. Yue, C. Zhou and J. J. Feng. *Spontaneous shrinkage of drops and mass conservation in phase-field simulations*. Journal of Computational Physics, vol. 223, no. 1, pages 1–9, April 2007. (Cited on page 136.)
- [Zhan *et al.* 2009] Z.-H. Zhan, J. Zhang, Y. Li and H. S.-H. Chung. *Adaptive Particle Swarm Optimization*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 39, no. 6, pages 1362–1381, 2009. (Cited on page 93.)
- [Šarler 1995] B. Šarler. *Stefan’s work on solid-liquid phase changes*. Engineering Analysis with Boundary Elements, vol. 16, no. 2, pages 83–92, 1995. (Cited on page 1.)