ADJOINT-BASED OPTIMIZATION FOR HYPERBOLIC BALANCE LAWS IN THE PRESENCE OF DISCONTINUITIES

BY

ALEXANDRU FIKL

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Master’s Committee:

Assistant Professor Vincent Le Chenadec, Director of Research
Assistant Professor Taraneh Sayadi
Abstract

In this thesis, we are interested in optimization in multiphase flows using discrete adjoint-based methods. The main issues we will endeavor to study are the impact on the linearized and adjoint equations of discontinuous solutions, the behavior of the THINC-family of numerical schemes under linearization and the rigorous formulation of an optimization problem when the interface is represented by a Heaviside marker function.

We will study the two main discontinuous wave patterns that appear in hyperbolic balance laws: contact discontinuities and shocks. These two have very different characteristics in the forward problem and we will see that they require different treatments for the adjoint formulations as well. Specifically, shocks require additional information, in the form of the shock location, to properly define first-order variations necessary for gradient methods. Another important aspect in the numerical treatment of multiphase flows is the use of an anti-diffusive numerical scheme for the interface advection equation. The scheme we will investigate is the fairly recent THINC scheme, because it is one of the few numerical schemes in the field that is differentiable (with respect to the volume fraction). However, we will need to extend the classic formulation of the THINC scheme to force it to behave correctly in the linearized and adjoint regime. Even with these extensions, we will see that the adjoint converges everywhere except at the interface, due to the discontinuity. Finally, we define a simplified optimization problem for the volume fraction, where the velocity is given by a velocity potential instead of the Navier-Stokes equations. This allows us to specifically study the formulation of the cost functional and the viability of using a Heaviside function as a representation of the interface (as opposed to e.g. level set methods). We will show that such a formulation is indeed possible and leads to well-posed optimization problems.
I would like to express my sincere gratitude to my two advisers, Vincent Le Chenadec and Taraneh Sayadi, for all their guidance and insights during the past two years. I never fail to be impressed by Professor Le Chenadec’s knowledge in fluid dynamics and I am glad I can partake and learn from him at every corner. Similarly, the expertise in adjoint methods that Professor Sayadi has shared with me in the past couple of years has been truly invaluable for somebody entering the field with a very naive view. I am very grateful for the opportunity that they have bestowed upon me to study at a top university on complex and interesting problems and hope they had as much fun with it this far as I have.

In obvious order of importance, I would also like to thank many of my friends and family that have been there for me in spite of everything. Yong Yi Bay has been a constant positive influence upon me and graciously shared his controversial knowledge of Asian cuisine, that I fear may have had a deep effect on my future preferences. I would also like to thank Jeaniffer Lissette Vides for being there and always taking the time to answer my increasingly specific questions about a myriad of numerical methods. Last and seemingly least, I would like to thank all the people that have helped me proofread this thesis and sort out the many remaining kinks.
Table of Contents

List of Tables ........................................................................................................ vi

List of Figures ........................................................................................................ vii

Chapter 1 Introduction .............................................................................................. 1
  1.1 Organization ...................................................................................................... 3

Chapter 2 Adjoint-based Optimization ..................................................................... 5
  2.1 Calculus of Variations ....................................................................................... 6
  2.2 Differentiate-then-discretize ............................................................................. 9
  2.3 Discretize-then-differentiate .......................................................................... 14

Chapter 3 Adjoint Equations of Hyperbolic Balance Laws ..................................... 16
  3.1 The Transport Equation ................................................................................... 17
     3.1.1 Continuous Linearized and Adjoint Equations ......................................... 17
     3.1.2 Discrete Linearized and Adjoint Equations ............................................. 20
     3.1.3 Numerical Results .................................................................................... 22
  3.2 Generalized Tangent Vectors .......................................................................... 26
  3.3 The Inviscid Burgers’ Equation ....................................................................... 29
     3.3.1 Spatial Consistency ............................................................................... 33
     3.3.2 Temporal Consistency ........................................................................... 39
  3.4 Conclusions .................................................................................................... 43

Chapter 4 Adjoint Equations for Anti-diffusive Schemes ....................................... 45
  4.1 THINC: Tangent of Hyperbola for Interface Capturing .................................. 46
     4.1.1 THINC Scheme ...................................................................................... 47
     4.1.2 Blended THINC Scheme ...................................................................... 53
     4.1.3 Extended THINC Scheme ................................................................... 56
  4.2 COMIC: Compact Mollifier for Interface Capturing ........................................ 59
     4.2.1 Jump Location ....................................................................................... 60
     4.2.2 Linearized Equations .......................................................................... 66
     4.2.3 Adjoint Equations ............................................................................... 67
  4.3 Comparison: Optimization ............................................................................. 68
  4.4 Conclusions .................................................................................................... 71

Chapter 5 Motion Planning ....................................................................................... 73
  5.1 Problem Description ...................................................................................... 74
  5.2 Continuous Adjoint Equations ....................................................................... 77
     5.2.1 Formal Description ................................................................................ 78
  5.3 Discrete Adjoint Equations ............................................................................ 81
     5.3.1 Unsplit Scheme ..................................................................................... 82
     5.3.2 Dimensional Splitting Scheme ............................................................... 84
5.4 Steepest Descent Algorithm ........................................ 86
5.5 Numerical Results ..................................................... 87
  5.5.1 One-dimensional Tests ......................................... 87
  5.5.2 Two-dimensional Tests ......................................... 89
5.6 Conclusions .......................................................... 96

Chapter 6  Conclusions ...................................................... 98

Appendix A  Generalized Tangent Vectors: Analytical First Order Variation .......... 101
Appendix B  Burgers’ Equation: Derivation of Adjoint Equations ......................... 103
Appendix C  THINC: Flux Derivatives .................................... 105
Appendix D  THINC: First-order Approximations .................................... 107
Appendix E  COMIC: Fluxes and Flux Derivatives .................................... 113
Appendix F  Motion Planning: Derivation of the Continuous Adjoint ..................... 118
Bibliography ................................................................. 123
List of Tables

3.1 Boxcar Test: $L_1$ error log results for the adjoint variable $p$. .......................... 26
3.2 Modified Lax-Friedrichs Scheme: $L_1$ error log results for the state and adjoint variables. 34
3.3 Upwind Scheme: $L_1$ error log results for the adjoint variable $p$. .......................... 35
3.4 van Albada Limited Scheme: $L_1$ error log results for the adjoint variable $p$. .......... 38
3.5 Runge-Kutta: $L_1$ error log results for the adjoint variable $p$. ............................... 42
List of Figures

3.1 Forward solution at time $t = T$ (left) and adjoint solution at time $t = 0$ (right) for different number of cells $N$. .......................................................... 24
3.2 Adjoint solution using the incompletely differentiated flux with the minmod limiter (left) and a finite difference approximation (right). .................................................. 25
3.3 Adjoint solution using the incompletely (left) and completely (right) differentiated flux with the van Albada limiter. .......................................................... 25
3.4 Exact state solution at time $t = 0$ and exact adjoint solution at time $t = 0$ for Example 3.3. 32
3.5 Adjoint solution at $t = 0$ for (3.25) (left) and a zoom around the shock position at $x = 0$ (right). 35
3.6 Adjoint solution at time $t = 0$ using the Upwind scheme. ................................. 36
3.7 Adjoint solution at $t = 0$ for (3.25) using the modified Lax-Friedrichs scheme and the exact solution (3.24) for the state variable $u(t,x)$. .............................................. 37
3.8 Adjoint solution at $t = 0$ for (3.25) using complete differentiation (left) and incomplete differentiation (right) of the flux limited scheme. ........................................... 38
3.9 Adjoint solution at $t = 0$ for (3.25) zoomed around the shock position at $x = 0$. .... 41
3.10 Adjoint solution at $t = 0$ for (3.25) using the explicit Heun's method (left) and the Implicit Midpoint method (right) with consistent time-stepping. ......................... 43
3.11 Solution of (3.14) at time $t = 0$ for the adjoint and nonlinear THINC schemes with $\eta = 12.5$ (left) and $\eta = 2.5$ (right). ......................................................... 51
3.12 Solution of (3.14) at time $t = 0$ for the linearized and nonlinear THINC schemes with $\eta = 12.5$ (left) and $\eta = 2.5$ (right). ......................................................... 52
4.1 The jump location $\delta$ (4.6) for various values of the steepness $\beta$. ....................... 50
4.2 THINC flux (left) and flux derivative (right) with respect to $\alpha$ compared to the upwind flux for $c > 0, \lambda = 0.9$ and $\gamma = 1$. ......................................................... 51
4.3 Solution of (3.14) at time $t = 1$ for the linearized and nonlinear THINC schemes with $\eta = 12.5$ (left) and $\eta = 2.5$ (right). ......................................................... 52
4.4 Solution of (3.14) at time $t = 0$ for the adjoint and nonlinear THINC schemes with $\eta = 12.5$ (left) and $\eta = 2.5$ (right). ......................................................... 53
4.5 A sine-based (left) and a piecewise linear (right) blending function for a width of $w = 0.2$ and a cutoff value $\epsilon = 0.05$. ......................................................... 54
4.6 The flux (left) and flux derivative (right) for the blended flux given by (4.13), with a width of $w = 0.2$ and a cutoff value $\epsilon = 0.05$. ......................................................... 55
4.7 The linearized scheme at time $T = 1$ for a slope steepness $\eta = 12.5$ (left) and $\eta = 2.5$ (right). ......................................................... 55
4.8 The adjoint scheme at time $t = 0$ for a slope steepness $\eta = 12.5$ (left) and $\eta = 2.5$ (right). ......................................................... 56
4.9 The linearized scheme at time $T = 1$ for a slope steepness $\eta = 12.5$ (left) and $\eta = 2.5$ (right). ......................................................... 56
4.10 The adjoint scheme at time $t = 0$ for a slope steepness $\eta = 12.5$ (left) and $\eta = 2.5$ (right). ......................................................... 57
4.11 Jump location for various values of $\eta$ obtained by solving (4.21). ......................... 61
4.12 Mollifier (4.20) with $\delta + \eta < -0.5$ (left) and $\delta - \eta > 0.5$ (right). .................. 62
4.13 Mollifier (4.20) with $\delta - \eta > -0.5$ and $\delta + \eta < 0.5$ (left) and $\delta - \eta < -0.5$ and $\delta + \eta > 0.5$ (right). ......................................................... 62
4.14 Mollifier (4.20) with $\delta - \eta < -0.5$ and $\delta + \eta < 0.5$ (left) and $\delta - \eta > -0.5$ and $\delta + \eta > 0.5$ (right). ......................................................... 63
4.15 The two critical values of alpha from (4.23) for $\gamma = 1$. .................................. 64
4.16 The flux (left) and flux derivative (right) for the COMIC scheme with $c > 0$ and $\gamma = 1$. .... 65
4.17 The linearized COMIC scheme for $\eta = 12.5$ (left) and $\eta = 2.5$ (right). ............... 66
Chapter 1

Introduction

The present work is focused on developing the necessary prerequisites for the optimization of incompressible two-phase flows. Specifically, the studies we will perform have in mind future applications involving the commonly used one-fluid model of two-phase flows. We will focus on using discrete adjoint methods, instead of their continuous counterparts, partly due to their simplicity in formulation (we only need to model the forward problem) and also due to the consistency of the discrete gradients. However, this approach is not without its share of trade-offs. To better understand these trade-offs, we will study some of the issues that appear in these types of flows due to discontinuities (e.g. contact discontinuities), the convergence to weak adjoint solutions, the behavior of commonly used numerical schemes, with a focus on anti-diffusive schemes, etc. Finally, we will attempt to formulate and solve an optimization problem regarding the movement of an interface in an idealized scenario of an incompressible and irrotational velocity field.

The type of the problem we will attempt to tackle is a PDE-constrained optimization problem. This type of problem has gained a lot of interest in recent decades with the advent of a rigorous formalism involving the optimal control of dynamic systems, largely due to the work of Pontryagin and others [Pontryagin, 1962]. Maybe the most prevalent of the methods used to solve this type of problem on a large scale is the adjoint method. The advantage provided by the adjoint method is that it offers an efficient way to compute the gradient of a given cost functional, in a manner independent of the number of control variables. The first applications of optimal control and the adjoint method to the field of fluid mechanics were performed in the seminal papers by Olivier Pironneau [Pironneau, 1974], who applied it to a simple shape optimization problem of a body in Stokes flow, and Anthony Jameson [Jameson, 1988], who, several years later, extended the approach to inviscid compressible flows. Ever since, optimal control and adjoint methods have been one of the most important components of computational fluid dynamics and related fields.

One of the most successful applications of adjoint methods has been in topology optimization [Newman, 1999; Allaire, 2004; Bendsoe, 2013], where other methods usually fail due to the significantly large size of the problem. Topology optimization is also very prevalent in the field of aerospace engineering with the works of [Jameson, 1998; Anderson, 1999; Giles, 2000] and others. However, adjoint methods have
been used in a very wide variety of applications, such as the inverse problem in solid mechanics [Oberai, 2003; Bonnet, 2005], seismology [Fichtner, 2006a; Fichtner, 2006b], geophysics [Plessix, 2006], mesh adaptation [Fidkowski, 2011], etc. More general approaches to optimal control and adjoint-based optimization can be found in classic monographs such as [Lions, 1971], focusing on all major types of PDEs, notably hyperbolic equations, [Tröltzsch, 2010; Borzi, 2012], focusing on elliptic and parabolic equations, or [Gunzburger, 2003], which offers a wide selection of examples using different types of controls.

We are interested in extending approaches such as [Gunzburger, 1999], which study the optimal control of the velocity field described by the Navier-Stokes equations, to the case of multiphase flows. To our knowledge, work on such problems is in its very early stages in the context of incompressible two-phase Navier-Stokes flows. As such, there are very few documented studies that deal with the issues that can arise from optimizing multiphase flows and methods to mitigate or solve these issues. Some other applications to general multiphase media include the two-phase Stefan problem [Bernauer, 2011], various studies in porous media [Jansen, 2011], rigid body motion in a multiphase flow [Springer, 2015], etc. In the specific case of a two-phase one-fluid model, we can immediately see a series of complications that can arise. Continuously, we have to introduce new terms and equations into the classic Navier-Stokes system related to the interface, modeling surface tension, mass transfer, etc., usually through a series of Heaviside functions and Dirac delta distributions. Discretely, we have another set of issues related to an accurate representation of a sharp fluid-fluid interface, accurate computation of surface tension, etc. We will analyze some of these issues in very simplified scenarios, in an attempt to gain intuition when approaching more realistic models. For an introduction to the issues inherent in modeling multiphase flows see [Prosperetti, 2009; Brennen, 2005].

The first issue we will attempt to deal with is the rigorous definition of the adjoint equations in the presence of discontinuities. Most descriptions of adjoint methods make strong assumptions on the differentiability of all the involved quantities. However, a sharp interface model is inherently discontinuous and puts into question the well-posedness of the adjoint system and all the manipulations performed to obtain it. Discontinuous solutions have been thoroughly studied in the case of hyperbolic systems of conservation laws, specifically in the context of shocks. Some early applications involving such flows include [Iollo, 1999; Homescu, 2003; Oyama, 1997]. Formally, the problem was first approached by [Bressan, 1995], which defined a new calculus for formulating first-order optimality conditions using generalized tangent vectors. The problem was later extended multiple times, for example in [Bardos, 2003; Giles, 2002; Ulbrich, 2001] and others. A recent important result can be found in [Giles, 2010a; Giles, 2010b], where a new modified Lax-Friedrichs scheme was defined, which allowed rigorous proofs of convergence of the discrete linearized and adjoint systems in the presence of multiple shocks. In our case, we are also interested in the behavior of
contact discontinuities, since the volume fraction gives rise to one such wave pattern in the one-fluid model.

Besides discontinuities in the solutions to the two-phase problem, we also encounter difficulties on a discrete level. Applying the discretize-then-differentiate approach to adjoint methods (see [Nadarajah, 2000; Giles, 2002]) leads to differentiating highly nonlinear numerical schemes. In multiphase flows, there are many complex numerical schemes that are in common use, for example the VOF-family of methods [Hirt, 1981; Rider, 1998], the closely related THINC-family of methods [Xiao, 2005; Yokoi, 2007], the level set methods [Osher, 1988], the front-tracking methods [Unverdi, 1992], etc. All these methods are very complex and contain numerous non-differentiable terms, which implies that their behavior under linearization is not clear. Furthermore, rigorous results, such as the ones obtained using the modified Lax-Friedrichs scheme, are out of reach even for the simplest of the previously mentioned numerical schemes. Therefore, we will confine ourselves to numerical tests to show the convergence of the adjoint scheme to known exact solutions.

Finally, an important issue in optimization in general is finding a well-posed cost functional that faithfully represents the solutions we want to obtain and has beneficial qualities, such as convexity. For our intended use in the context of multiphase flows, we require formulating cost functionals that involve the position of the interface or the volume fraction. These are generally modeled using either Dirac delta functions or Heaviside functions, both of which pose some problems numerically. We will see such a formulation, the problems that come with it and how they can be solved. Furthermore, we will show that controlling the movement of an interface is possible using only the volume fraction. This is done on two levels: continuously, where a rigorous approach to the existence and uniqueness of solutions to state and adjoint equations is provided, and discretely, where we will show through a series of tests that we can obtain a desired interface motion, even in non-trivial cases.

1.1 Organization

This thesis is organized into 6 chapters, out of which the current introduction is the first. Chapter 6 contains the conclusions of our analysis of discontinuous solutions to hyperbolic balance laws and applications to interface motion planning.

In Chapter 2, we will give a short introduction to adjoint methods, with the goal of introducing common notation and current approaches to optimization with adjoints. The most important part of adjoint methods is the development of a calculus of variations in infinite dimensional vector spaces. We will follow [Borzi, 2012; Tröltzsch, 2010] and introduce the ideas behind directional derivatives and first-order optimality conditions in such spaces. This will allow us to delve, in a rigorous manner, into the two different approaches to
adjoint methods: the *differentiate-then-discretize* approach and the *discretize-then-differentiate* approach. Finally, we will shortly investigate how the adjoint equations are obtained in these two scenarios, under the assumption that all involved quantities are sufficiently smooth.

In Chapter 3, we will define the adjoint equations for linear and nonlinear hyperbolic balance laws in the presence of discontinuities. We will investigate the linear advection equation and Burgers’ equation as two representatives of the main issues that may arise from hyperbolic equations. The main two problematic wave patterns exhibited by each of them are contact discontinuities and shocks, respectively. The investigation will be performed on two levels: continuous, where the definition of first-order variations of discontinuous solutions is not trivial, and discrete, where convergence to weak adjoint solutions and the non-differentiability of commonly used numerical schemes are problematic.

In Chapter 4, we will move closer to two-phase flows by analyzing the THINC family of schemes, a widely used set of numerical schemes with very good anti-diffusive qualities. The THINC schemes are highly nonlinear because of the use of a hyperbolic tangent in the definition of the numerical flux. These nonlinearities cause many issues under linearization. We will investigate the behavior of the THINC schemes and propose a series of improvements that make them a viable choice when used in the context of the discretize-then-differentiate approach to adjoint methods.

In Chapter 5, we will study the optimization of a simplified two-phase flow problem. The problem consists in optimizing the movement of an interface described by the advection of a volume fraction in an incompressible and irrotational velocity field. Under these assumptions, the velocity field is fully described by a velocity potential. To reduce the dimensionality of the problem even further, we will control the system using the boundary conditions of the Laplace equation satisfied by the velocity potential. While this is not a full two-phase model, it will allow us investigate issues related to the formulation of such systems in the context of sharp interface models, where the interface is represented by a Heaviside function. Numerical results are given to prove the viability of the presented approach.
Chapter 2

Adjoint-based Optimization

In this chapter we will attempt to give a very broad view on the optimization and optimal control of flows with PDE-based constraints with a focus on adjoint-based methods. The problems discussed here take a very general form:

\[
\begin{align*}
\min_{u,g} & \quad J(u, g), \\
\mathcal{F}(u, g) &= 0
\end{align*}
\]  

(2.1)

where we can identify the main components of a constrained optimization problem:

- \( u \), known as the state variables. In the setting of fluid mechanics, state variables usually represent the velocity field (alternatively represented by a velocity potential or stream function), pressure, density, temperature, etc.

- \( g \), known as the control variables or design parameters that are commonly part of a constraint set \( G_{\text{ad}} \). There are also many examples of control variables, of which maybe the most well known ones are parameters determining the shape of an object in design optimization. Other examples include the heat flux or inflow mass rate through a wall, etc.

- \( \mathcal{F}(u, g) \) is the set of (PDE-based) constraints that both the state variable \( u \) and the control variable \( g \) must satisfy. For the type of flow optimization problems we are interested in, the constraints will usually be the Euler or Navier-Stokes equations or simpler models like the linear advection equation or Burgers' equation.

- \( J \) is the cost or objective functional. Common types of cost functionals include tracking-type functionals, which match the solution to a desired result at a certain time (or for all time), lift maximization or drag minimization functionals, etc.

Issues regarding the existence and uniqueness of a solution \( \hat{g} \) to such problems are very complex and will not be discussed here. For an in-depth analysis we direct the reader to classic materials such as [Lions, 1971] or more recent works such as [Gunzburger, 2003], [Tröltzsch, 2010] or [Borzi, 2012].
In the rest of this chapter we will focus on defining a set of approaches to solve the above optimization problem. The problem will be kept abstract as much as possible with explicit examples given in the chapters to come. Furthermore, for now we will assume that both the cost functional $J$ and the constraints $F$ are differentiable (in a certain sense), as issues regarding non-differentiability will be discussed later.

2.1 Calculus of Variations

To avoid any ambiguity, we will first give an incomplete introduction to the ideas of the calculus of variations, with the main aim of introducing the notation and definitions we will be using for the rest of the chapter.

The most important component of the calculus of variations is the idea of a directional derivative, of which there are numerous types throughout the literature. We will focus here on two of them: the Gâteaux derivative and the Fréchet derivative. Firstly, consider a mapping $f$ from $X$ to $Y$, two affine vector spaces:

$$ f : X \longrightarrow Y $$

If it exists, the Gâteaux derivative of $f$ at a point $x$ in the direction $y$ is given by the limit:

$$ \frac{df}{dx}(x; y) = \lim_{\epsilon \to 0} \frac{f(x + \epsilon y) - f(x)}{\epsilon}, $$

in which case we call $f$ Gâteaux-differentiable. Note that the Gâteaux derivative defined in this way is not necessarily linear in $y$, as we would expect from a derivative operator. However, it must still be homogeneous of the first degree, in the sense that $df(x; \alpha y) = \alpha df(x; y)$.

Many of the applications we will be looking at will require a slightly stronger notion of differentiability than that given by Gâteaux derivatives, namely the implication that if the directional derivative exists at a given point, the mapping must be continuous at that point. To allow for this, we must impose further structure on our vector spaces $X$ and $Y$: they must be complete normed vector spaces, i.e. Banach spaces.

Consider now the mapping $f : X \to Y$ defined on $X, Y$ Banach spaces endowed with the norms $\| \cdot \|_X$ and $\| \cdot \|_Y$, respectively. If, for a fixed $x \in X$, there exists a bounded linear operator $df(x)$ such that:

$$ \lim_{y \to 0} \frac{\|f(x + y) - f(x) - df(x; y)\|_Y}{\|y\|_X} = 0, $$

then $df(x; y)$ is called the Fréchet-derivative of $f$ at $x$ in the direction $y$. As mentioned, this is a stronger condition on $f$ and, as such, a Fréchet-differentiable mapping is necessarily Gâteaux-differentiable, in the
sense that we have defined them. From now on we will only be using the term directional derivative to refer to the Fréchet derivative of a mapping. To reiterate, we will be using the following notation to denote a Fréchet derivative of a mapping with one variable at a point \( x \) in the direction \( y \):

\[
\frac{df}{dx}(x; y) \quad \text{or} \quad df(x; y),
\]

often forsaking the mention of the direction to refer to the derivative operator itself. If the mapping takes multiple variables, for example in a product space \( X_1 \times X_2 \), we denote its Fréchet derivatives with the following common notation for partial derivatives:

\[
\frac{\partial f}{\partial x_1}(x_1, x_2; y_1, 0) \quad \text{and} \quad \frac{\partial f}{\partial x_1}(x_1, x_2; 0, y_2)
\]

at a point \((x_1, x_2)\) in the direction \((y_1, 0)\) and \((0, y_2)\), respectively. Note that these are not what we understand by partial derivatives in the classical sense, but simply denote the usual Frechét derivative in a particular set of directions.

A very important special case is that of functionals, i.e. mappings to the real line with \( Y = \mathbb{R} \). This case allows us to return to the familiar ground of derivatives of multivariate mappings by allowing the definition of a gradient. To define the gradient we will also assume that \( X \) is a Hilbert space (inner product spaces could also be used, but they are not sufficient for uniqueness) and use the Riesz representation theorem which gives us:

\[
\frac{df}{dx}(x; y) = (\nabla f(x), y),
\]

where \((\cdot, \cdot)\) is the inner product on \( X \) and \( \nabla f(x) \) is the gradient of \( f \) at \( x \). The spaces we are usually interested in, such as \( L^2 \) or \( H^k \) (i.e. the Sobolev spaces \( W^{k,2} \)), are all Hilbert spaces, so we will usually be able to define the gradient in such a way. Note that, in the case of Hilbert spaces, the Riesz Representation theorem basically gives the equivalence \( X^* = X \), but this is not to be understood as set equality. The equality is in the sense that there exists a norm-preserving isomorphism between the two spaces, which is what the theorem tells us.

Note, however, that computing the directional derivative of \( f \) is usually quite straightforward, but finding the gradient can prove to be more challenging depending on the choice of Hilbert space. Another important consequence of defining the gradient in such a way is that the definition invariably depends on the choice of inner product. This can have a very large impact on the convergence of an optimization algorithm and the resulting solution. For example, using the standard inner product on \( H^1 \) will introduce derivatives into the
inner product which can have a smoothing effect on the result. For a deeper discussion and examples of this phenomenon, see [Borzi, 2012].

In the case of functionals on a Hilbert space, we can also make use of dual space $X^*$ of $X$ and note that the directional derivative is a mapping into the dual space:

$$\frac{df}{dx} : X \rightarrow X^*,$$

which, when applied to a given point, becomes an element of that dual space:

$$\frac{df}{dx}(x; \cdot) : X \rightarrow \mathbb{R}$$

and, similarly, the gradient is an endomorphism on $X$:

$$\nabla f : X \rightarrow X,$$

such that the gradient at a point is an element of the space itself $\nabla f(x) \in X$. These are very important considerations, since the differences between directional derivatives and gradients are often a matter of confusion.

Getting back to our optimization problem (2.1), we will assume that the state variables $u \in U$ are elements of a given Hilbert space and, similarly, the constraints $g \in G_{ad} \subset G$ are at least in a convex subset of a given Hilbert space $G$ ($G_{ad}$ can also be a subspace or the full space, when there are no constraints on $g$). These assumptions will apply to most, if not all, of the practical examples we are interested in. Given these definitions our cost functional is defined as the mapping:

$$J : U \times G \rightarrow \mathbb{R}, \quad (2.2)$$

and the constraints are given by a mapping:

$$F : U \times G \rightarrow P, \quad (2.3)$$

where $P$ is a Hilbert space as well, whose use will become important in defining the adjoint equations. Since $J$ is a functional on a Hilbert space (product space of two Hilbert spaces), all the definitions of the
directional derivatives from above will apply. More importantly, we can also apply the chain rule to obtain:

\[
\frac{dJ}{dg} (u(g), g) = \frac{\partial J}{\partial u} (u, g) \frac{du}{dg} (g) + \frac{\partial J}{\partial g} (u, g).
\] (2.4)

We note that we can consider \( J \) as a single variable functional by using the implicit function theorem to motivate the existence of \( u(g) \) such that \( F(u, g) = 0 \). In the chain rule above, we have mentioned the following directional derivatives which belong to their respective dual spaces:

\[
\frac{\partial J}{\partial g} (u, g) \in G^* \quad \text{and} \quad \frac{\partial J}{\partial u} (u, g) \in U^*,
\]

as well as

\[
\frac{du}{dg} (g) : G \rightarrow U,
\]

which is a bounded linear operator, but it does not belong to a dual space. However, the composition of the two operators does:

\[
\frac{\partial J}{\partial u} \frac{du}{dg} \in G^*,
\]

as it must, since the left-hand side is part of the dual space of \( G \). We also introduce the following gradient notation corresponding to the various directional derivatives:

- \( \nabla_u J(u, g) \) is the gradient with respect to \( u \), while keeping \( g \) constant and
- \( \nabla_g J(u, g) \) is the gradient with respect to \( g \), while keeping \( u \) constant,

with the gradients being obtained using the Riesz representation theorem, as defined in the more general setting.

### 2.2 Differentiate-then-discretize

Having defined all the necessary terms, we can proceed to derive a generic formulation of the so-called optimality system and the adjoint method. The first step in this derivation is making use of Lagrange relaxation (also known as the Lagrange multiplier method) to transform the constrained optimization problem (2.1) into an unconstrained optimization problem. This can be accomplished by introducing the Lagrangian (or Lagrange functional):

\[
L(u, g, p) = J(u, g) - (p, F(u, g))_p,
\] (2.5)
where $(\cdot, \cdot)_P$ is the inner product of $P$, introduced as the image of $(2.3)$, and $p \in P$ is called a Lagrange multiplier, which is used to weakly impose the constraints as part of the functional. Note that the existence of the Lagrange multiplier $p$ is not guaranteed and is usually dependent on the surjectivity of $F$, which we will assume from now on. Now that we have an unconstrained problem, we can define the usual first-order necessary conditions for a minimum:

\[
\begin{align*}
\nabla_u L(u, g, p) &= 0, \\
\nabla_g L(u, g, p) &= 0, \\
\nabla_p L(u, g, p) &= 0.
\end{align*}
\]

We have separated the derivatives of the Lagrangian with respect to each of the variables of interest because each of them will give an important part of the optimality system, as we will see. We will start with the last condition, since it is the simplest one. We have the following directional derivative:

\[
(\nabla_p L, \tilde{p})_P = \left( \frac{\partial L}{\partial p}(u, g, p; \tilde{p}) \right)_P = - (\tilde{p}, F(u, g))_P = - (\tilde{p}, F(u, g))_P,
\]

for an arbitrary admissible direction $\tilde{p}$, so our first-order optimality condition reads:

\[
\nabla_p L = - F(u, g) = 0,
\]

i.e. the derivative with respect to the Lagrange multiplier $p$ simply retrieves the original constraints. Next we will look at the derivative with respect to the state variable $u$. We have that:

\[
(\nabla_u L, \tilde{u})_U = \left( \frac{\partial L}{\partial u}(u, g, p; \tilde{u}) \right)_U = \left( \frac{\partial J}{\partial u}(u, g; \tilde{u}) - \left( p, \frac{\partial F}{\partial u}(u, g; \tilde{u}) \right)_P \right)_U,
\]

where we can use the Riesz representation theorem for the first directional derivative:

\[
\frac{\partial J}{\partial u}(u, g; \tilde{u}) = (\nabla_u J, \tilde{u})_U.
\]

However, transforming the inner product into a similar form is slightly more complicated, since we have that:

\[
\frac{\partial F}{\partial u}(u, g) : U \rightarrow P
\]

is a bounded linear operator and not a functional. In this scenario, we make use of the definition of an
adjoint operator:
\[
\left( \frac{\partial F}{\partial u}(u, g) \right)^*: P \rightarrow U
\]
which must satisfy:
\[
\left( p, \frac{\partial F}{\partial u}(u, g; \tilde{u}) \right)_P = \left( \left( \frac{\partial F}{\partial u}(u, g) \right)^* p, \tilde{u} \right)_U.
\]

The existence and uniqueness of the adjoint operator in a Hilbert space is a consequence of the Riesz Representation theorem as well. Putting the two terms back together, we obtain the gradient of the Lagrangian:
\[
\nabla_u L = \nabla_u J - \left( \frac{\partial F}{\partial u} \right)^* p = 0.
\]

Finally, we will look at the derivatives with respect to the control variable \( g \). Very similarly, we have that:
\[
\left( \nabla_g L, \tilde{g} \right)_G = \frac{\partial L}{\partial g}(u, g, p; \tilde{g}) = \frac{\partial J}{\partial g}(u, g; \tilde{g}) - \left( p, \frac{\partial F}{\partial g}(u, g; \tilde{g}) \right)_P.
\]
Using the same reasoning as before, we obtain:
\[
\nabla_g L = \nabla_g J - \left( \frac{\partial F}{\partial g} \right)^* p = 0.
\]

The optimality system of the given optimization problem (2.1) is given by the explicit form of the three first-order conditions that we have derived this far. It is formed out of:

- The constraints, or state equations:
  \[
  F(u, g) = 0. \tag{2.6}
  \]

- The adjoint equations:
  \[
  \left( \frac{\partial F}{\partial u} \right)^* p = \nabla_u J \tag{2.7}
  \]

- The optimality condition:
  \[
  \left( \frac{\partial F}{\partial g} \right)^* p = \nabla_g J \tag{2.8}
  \]

There are at least two ways to solve the above system (see [Gunzburger, 2003]). The naive method is using what is known as a one-shot method. In the case of the system (2.6)-(2.8), we have 3 unknowns \( u, g, p \) and as many equations, so the existence of a solution is indeed possible. However, our constraints can be complex nonlinear PDEs which are very hard to solve, so we would expect the coupled system to be even more formidable. This makes one-shot methods intractable in the case of PDE-based constraints.
An alternative to using one-shot methods is using an iterative method to solve the system. Such a method would require the gradient of the cost functional as given in (2.4). However, this poses its own set of problems since the term
\[
\frac{du}{dg}
\]
will have to be computed for every control variable \(g\). In many applications, such as shape optimization, we have a very large amount of control variables, so finding the above derivative would require solving the constraint PDE system for each variation. This is a valid approach, known as computing the gradient through \textit{sensitivities} (of \(u\) with respect to \(g\)). An alternative method is the \textit{adjoint method}, which sidesteps the requirement of solving a system of PDEs for each \(g\) by instead solving the adjoint system given by (2.7).

To obtain the adjoint formulation of the gradient, we will first look at the derivatives of the constraints from (2.1) using the chain rule as in (2.4):

\[
\frac{dF}{dg}(u(g),g) = \frac{\partial F}{\partial u} \frac{du}{dg}(u,g) + \frac{\partial F}{\partial g}(u,g) = 0
\]

\[
\Rightarrow \frac{du}{dg} = - \left( \frac{\partial F}{\partial u} \right)^{-1} \frac{\partial F}{\partial g},
\]

which we can actually replace into (2.4) to obtain:

\[
\frac{dJ}{dg}(u(g),g) = \frac{\partial J}{\partial g}(u,g) - \frac{\partial J}{\partial u} \left( \frac{\partial F}{\partial u} \right)^{-1} \frac{\partial F}{\partial g}(u,g)
\]

We have a choice in the above formulation about how to associate the products in the second term. Associating right to left:

\[
\frac{\partial J}{\partial u} \left( \frac{\partial F}{\partial u} \right)^{-1} \frac{\partial F}{\partial g}
\]

gives the sensitivity method, while associating left to right:

\[
\left\{ \frac{\partial J}{\partial u} \left( \frac{\partial F}{\partial u} \right)^{-1} \right\} \frac{\partial F}{\partial g},
\]

gives the adjoint method. For the adjoint method, we define \(p^* \in P^*\) such that:

\[
p^* = \frac{\partial J}{\partial u} \left( \frac{\partial F}{\partial u} \right)^{-1} : P \rightarrow \mathbb{R}
\]

which gives:

\[
\frac{dJ}{dg}(u(g),g) = \frac{\partial J}{\partial g}(u,g) - p^* \frac{\partial F}{\partial g}(u,g).
\]
We will now apply the Riesz representation theorem on this formulation where we have denoted the Lagrange multiplier \( p \in P \) as the representer of \( p^* \):

\[
(\nabla J, \tilde{g}) = (\nabla g J, \tilde{g}) - (p, \frac{\partial F}{\partial g}(u, g, \tilde{g})) = (\nabla g J, \tilde{g}) - \left( \left( \frac{\partial F}{\partial g} \right)^* p, \tilde{g} \right)
\]

or:

\[
\nabla J = \nabla g J - \left( \frac{\partial F}{\partial g} \right)^* p.
\]

where \( p \) still satisfies (2.7). We can see that we have now eliminated the directional derivative of \( u(g) \) from the system. Not only this, but, to compute the gradient as given in (2.9), we only need to solve the adjoint system once to obtain \( p \). The gradient we have obtained here is exactly the same as the derivative of the Lagrangian \( L \) with respect to \( g \) that gives the optimality condition from (2.8). This is to be expected since incorporating the constraints into the Lagrangian and incorporating them directly into the derivative of the cost functional is completely equivalent.

See [Borzi, 2012] for further details about these results as well as how to obtain second order conditions using the Hessian and [Gunzburger, 2003] for details on the one-shot method and obtaining the gradient using sensitivities (as well as plenty of examples).

Since it is unlikely we will find an analytical solution to our optimization problem, the next step will be discretizing the optimality system and the gradient from (2.9) and using any of the well-known gradient descent methods to solve the optimization problem.

The method we have presented here is known as the differentiate-then-discretize approach to adjoint-based optimization, referring to the order of the operations: we have first derived the continuous optimality system which we will then discretize accordingly. This method has its advantages:

- We can use different numerical schemes to solve the state and adjoint equations;
- We can use different meshes;
- We can avoid differentiating numerical artifacts such as nonlinearities in the numerical scheme (e.g. limiters in MUSCL schemes), moving meshes, turbulence models, shock-capturing schemes, etc.

However, there is at least one significant downside to this approach. Since we discretize the state and adjoint equations separately, there is no guarantee that the discrete gradient will actually correspond to the continuous one that we expect. Examples are given in [Gunzburger, 2003] in which the approximate gradient doesn’t only not point in the direction of steepest descent, it actually points completely in the
opposite direction. This can be remedied with regularization or other methods, but it is generally a very important issue.

### 2.3 Discretize-then-differentiate

As a counterpart to the differentiate-then-discretize approach we have the discretize-then-differentiate approach, which does the two steps in reverse order (of course the differentiation and discretization operators do not commute). In this case it is perhaps more intuitive to look at the problem from a linear algebra point of view (see also [Giles, 2000]) where \( u \in \mathbb{R}^N \) is the discretized \( u \) on a space-time grid with a total of \( N = N_t \times N_s \) discrete “points”. Similarly \( g \in \mathbb{R}^M \) is the set of discrete constraints. Unlike the state variables, the constraints could already be a discrete set or we could have infinite dimensional constraints which require discretization. The discrete PDE-based constraints are then denoted by \( F^h(u, g) \) and the discrete cost functional is denoted by \( J^h(u, g) \). In this case, we can talk about derivatives in a much more familiar setting, i.e. in \( \mathbb{R}^n \) for some \( n \), but the same rules apply as we have seen in the first section of this chapter.

We still have the derivative of the cost functional obtained with the chain rule as in (2.4):

\[
\frac{dJ^h}{dg} = \frac{\partial J^h}{\partial u} \frac{du}{dg} + \frac{\partial J^h}{\partial g}
\]

and the derivative of the discretized constraints:

\[
\frac{dF^h}{dg} = \frac{\partial F^h}{\partial u} \frac{du}{dg} + \frac{\partial F^h}{\partial g} = 0.
\]

Discretely, we also have a choice of space that we can choose from (generally this will be the Euclidean space) and an associate inner product (generally given by a weighted inner product \( (x, y) = x^T My \), where \( M \) is a positive-definite matrix). This choice will also influence the resulting formulation and gradient. To use the adjoint method, we will once again reduce the two above equations to a single one:

\[
\frac{dJ^h}{dg} = \frac{\partial J^h}{\partial g} + h^T v,
\]
where \( v \) solves \( Lv = f \) with the following notation:

\[
\begin{align*}
  v &= \frac{du}{dg} \in M_{N \times M}, \\
  h^T &= \frac{\partial J^h}{\partial u} \in \mathbb{R}^N, \\
  L &= \frac{\partial F^h}{\partial u} \in M_{N \times N}, \\
  f &= -\frac{\partial F^h}{\partial g} \in M_{N \times M}.
\end{align*}
\]

As before, we note that to solve the system \( Lv = f \), we actually have to solve \( M \) different systems with \( L \) as the operator. This corresponds to computing the gradient using sensitivities in the continuous case. We can now introduce the adjoint variable \( p \in \mathbb{R}^N \) which solves \( Lp = h \) so that our gradient becomes:

\[
\frac{dJ^h}{dg} = \frac{\partial J^h}{\partial g} + p^T f.
\]

The main advantage of this approach is that the gradient above is guaranteed to be the gradient of the discrete cost functional, since the approximations have been made beforehand. Because of the consistency of the gradient, this can also lead to considerably improved convergence of any descent algorithm. However, all the advantages of the \textit{differentiate-then-discretize} method have disappeared here. Using different numerical schemes or meshes for the state and adjoint equations is, of course, no longer possible and we are left with linearizing complex numerical schemes, which can be a daunting task on its own.

Because of these difficulties, a very important tool for adjoint-based optimization is \textit{automatic differentiation}. It can automatically create the linearized and adjoint code from the state equations and can save large amounts of time. However, it does have its own downsides: as any automatic process, it is unlikely to be as optimized as a manual implementation and will use both more storage and more processing power in most non-trivial cases. A thorough comparison between the two approaches can be found in [Nadarajah, 2000].
Chapter 3

Adjoint Equations of Hyperbolic Balance Laws

In the previous chapter we have seen a very generic derivation of the adjoint equations and possible approaches to solving a given constrained optimization problem. A very important assumption made in all the previous results was that of differentiability: both the variables $u, g$ themselves as well as the cost functional $\mathcal{J}$ and the constraints $\mathcal{F}$ were assumed to be Fréchet-differentiable. However, discontinuous solutions naturally appear in most hyperbolic equations with the development of different wave patterns, such as shocks, even when initial conditions are smooth.

In this chapter we will attempt to focus more on the issue of discontinuous solutions and how the adjoint equations would be derived and defined in such a case. To simplify our analysis, we will be looking at a simplified optimization problem using the well-studied [Ulbrich, 2001; Giles, 2002; Giles, 2010a] tracking-type cost functionals:

$$\mathcal{J}(u) = \int_\Omega G(u(T, x)) \, dx,$$

where $\Omega = [a, b]$ is a 1D domain and $G$ is assumed to be differentiable. The constraints are given by:

$$\begin{align*}
\mathcal{F}(u(t, x)) &= 0, \quad (t, x) \in [0, T] \times \Omega, \\
u(0, x) &= g(x), \quad x \in \Omega,
\end{align*}$$

where $\mathcal{F}$ defines the PDE with some appropriate boundary conditions. The control is given as the initial condition for our PDE and does not appear anywhere else in the constraints. The quantity of interest is the variation of $\mathcal{J}$ with respect to $g$, which we can then use to obtain the gradient. In this case, since the cost functional does not depend explicitly on $g$, we will be looking at the part of (2.4) corresponding to the variation in $u$:

$$\frac{d\mathcal{J}}{du} (u; \tilde{u}) = \int_a^b G'(u(T, x))\tilde{u} \, dx,$$

where $\tilde{u}$ is the variation of $u$, which itself depends on $g$. If either $u$ or $g$ or both are discontinuous, this variation of $\tilde{u}$ is not necessarily well-defined in a classical sense, as we will see. For a similar study, but with
a focus on continuous solutions, see [Ou, 2011] or [Liu, 2008].

In this chapter we will look at two types of discontinuities that appear in fluid mechanics:

- **contact discontinuities**, which are one of the basic wave patterns in hyperbolic systems. We are interested in contact discontinuities because they appear in models of two-phase flow as a result of the sharp interface. To study contact discontinuities, we will look at the transport equation with piecewise constant profiles that mimic the Heaviside function used to define an interface.

- **shocks**, which are stronger discontinuities and pose more problems, as we will see. To study shocks, we will look at the inviscid Burgers’ equation with similar piecewise constant profiles.

### 3.1 The Transport Equation

#### 3.1.1 Continuous Linearized and Adjoint Equations

We will start by looking at weak solutions to the 1D advection equation:

\[
\begin{align*}
\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} &= 0, \quad (t, x) \in [0, T] \times \Omega, \\
u(0, x) &= g(x), \quad x \in \Omega,
\end{align*}
\]

(3.3)

with appropriate boundary conditions at \(a\) or \(b\) and for a constant velocity field \(c\). Generally (for non-constant velocity \(c\)), a weak solution \(u\) to the advection equation belongs to some measure space (see [Le Floch, 1990]). To study the issue of whether (3.2) is well defined we will be looking at a perturbation of the control variable \(g\):

\[g'(x) = g(x) + \epsilon g'(x) + o(\epsilon).\]

Under some regularity assumptions, the solution \(u\) to the advection equation (3.3) can also be expressed as:

\[u'(t, x) = u(t, x) + cu'(t, x) + o(\epsilon),\]

where the perturbation \(u'(x)\) also satisfies (3.3), since the advection equation is linear.

**Example 3.1.** To see that the above expansion actually holds without any additional information, we will look at a simple initial condition:

\[u(0, x) = g(x) = \chi_{[0,1]}(x),\]
with the perturbed initial condition given by:

\[ u^\epsilon(0, x) = g^\epsilon(x) = (1 + \epsilon)\chi_{[0,1]}(x), \]

where \( \chi \) is the characteristic function. For constant \( c \), this initial condition gives the solution:

\[ u^\epsilon(t, x) = (1 + \epsilon)\chi_{[0,1]}(x - ct) = (1 + \epsilon)\chi_{[ct,ct+1]}(x). \]

The variation is then given by:

\[ u'(x) = \lim_{\epsilon \to 0} \frac{1}{\epsilon} (u^\epsilon(t, x) - u(t, x)), \]

but this limit can be exactly computed:

\[ \lim_{\epsilon \to 0} \frac{1}{\epsilon} (u^\epsilon(t, x) - u(t, x)) = \lim_{\epsilon \to 0} \frac{1}{\epsilon} ((1 + \epsilon)\chi_{[ct,ct+1]}(x) - \chi_{[ct,ct+1]}(x)) = \chi_{[ct,ct+1]}(x). \]

We have obtained that \( u'(t, x) = u(t, x) \), which is not surprising, given the advection equation is linear. In actuality, the limit we have computed here is the directional derivative of \( u \) with respect to \( g \). The limit above can be rewritten as:

\[ \frac{du}{dg} (g; g') = u'(g) = \lim_{\epsilon \to 0} \frac{1}{\epsilon} (u(g + \epsilon g') - u(g)) = u, \]

which is well-defined and belongs to the same space as \( u \) (trivially).

We have seen that, in the case of the advection equation, a perturbation in the control variable \( g \) gives a well-defined perturbation in the state variable \( u \), even in the case of discontinuous solutions. This is mainly due to the linearity of the advection equation, which will no longer hold in the second section for Burgers' equation.

To find the adjoint equation, we introduced the Lagrangian:

\[ \mathcal{L}(u, g, p) = \mathcal{J}(u) - \int_0^T \int_{\Omega} p(u_t + cu_x) \, dx \, dt - \int_{\Omega} q(u(0, x) - g(x)) \, dx, \]

with the two Lagrange multipliers \((p, q)\) used to enforce the constraints. Without going into details, we can derive the adjoint of the operator:

\[ \frac{\partial}{\partial t} + c \frac{\partial}{\partial x} \]
by integration by parts, which leads to the following adjoint equation:

\begin{equation}
\begin{cases}
-\frac{\partial p}{\partial t} - c \frac{\partial p}{\partial x} = 0, & (t, x) \in [0, T] \times \Omega, \\
p(T, x) = G'(u(T, x)), & x \in \Omega,
\end{cases}
\end{equation}

(3.4)

with appropriate boundary conditions. Note that we can meaningfully take variations of the Lagrangian with respect to $u$ and $g$, since we have shown in Example 3.1 that they are well-defined. Having determined the state equations (3.3) and the adjoint equations (3.4), we are left with the optimality condition to complete our optimality system (2.6)-(2.8). It is given by:

\begin{equation}
 p(0, x) = 0,
\end{equation}

(3.5)

which implies that the gradient of our cost functional is in fact:

\begin{equation}
\nabla_g J = p(0, x).
\end{equation}

(3.6)

Note that the adjoint equations (3.4) have not been derived in a very rigorous manner. In fact, if we assume that $u$ and $p$ both belong to the same measure space (which is a reasonable assumption since they solve similar equations), then the term:

\[
\int_0^T \int_\Omega p \left( \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} \right) \; dx \; dt,
\]

in the Lagrangian is not well-defined. For the integral above to be well-defined we would need additional regularity in the state variable $u$ or the adjoint variable $p$. However, this is undesirable and would unnecessarily limit the possible solutions. Another way to formulate the problem would be to sidestep the need for a Lagrangian. We have seen in Chapter 3 that we can also obtain the relevant adjoint equations by manipulating the cost functional. Using this alternative method, we can derive the above equations, which are well-defined in the usual distribution sense. The only problem with this alternative is that finding the adjoint to the operator $\partial_t + c\partial_x$, which appeared naturally in the Lagrangian formulation, must be done through other methods.
3.1.2 Discrete Linearized and Adjoint Equations

Having completely defined the continuous optimality system, we will now continue to the discrete case. We will attempt to solve the advection equation given by (3.3) using a Finite Volume method and study the convergence of the adjoint numerical scheme, obtained by linearizing a chosen Finite Volume scheme.

Before continuing, we will swiftly introduce the common notation used in Finite Volume methods that will be used through this thesis. We define a discretization of the domain \( \Omega = [a, b] \) as the union of a set of cells \( T_i = [x_{i-1/2}, x_{i+1/2}] \) of size \( \Delta x_i \) and centered at \( x_i \), where:

\[
\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \quad \text{and} \quad x_i = \frac{1}{2} \left( x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}} \right),
\]

so that the distance between two cell centers becomes:

\[
\Delta x_{i-\frac{1}{2}} = x_i - x_{i-1} = \frac{1}{2} (\Delta x_i + \Delta x_{i-1}).
\]

We also introduce the following cell-averaged state variables:

\[
u_i^m = \frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} u(t^m, x) \, dx
\]

and the cell-averaged control variables:

\[
g_i = \frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} g(x) \, dx.
\]

The Finite Volume method is based on approximating the values at the cell centers \( x_i \) by integrating over the domain \( [t^n, t^{n+1}] \times T_i \) and computing the fluxes over the cell boundaries using the cell-averaged values defined above. Applying this methodology to the advection equation in (3.3), we obtain the following formulation:

\[
\begin{cases}
    u^{m+1} = u^m + \Delta t A(u^m), \\
    u^0 = g.
\end{cases}
\]

where we have used a Forward Euler time discretization and a space discretization given by the operator \( A \), which can be written in the usual conservative form:

\[
A_i(u^m) = \frac{1}{\Delta x_i} (f_{i+\frac{1}{2}}^m - f_{i-\frac{1}{2}}^m),
\]
for some suitable definition of the fluxes $f_{i-1/2}^m$. Unlike the continuous advection equation, the discrete variants need not be linear (and it isn’t in most cases). For this reason, we need to separately define the linearized scheme for a perturbation $v$:

$$
\begin{cases}
    v^{m+1} = v^m + \Delta t \left( \frac{\partial A}{\partial u}(u^m) \right) v^m, \\
    v = h,
\end{cases}
$$

(3.9)

where the initial condition is given by a perturbation $h$ in the control variable $g$. The adjoint scheme can be obtained by transposing the time integrator and space discretization in the linearized scheme:

$$
\begin{cases}
    p^m = p^{m+1} + \Delta t \left( \frac{\partial A}{\partial u}(u^m) \right)^T p^{m+1}, \\
    p^M = G'(u^M),
\end{cases}
$$

(3.10)

We can also discretize the cost functional (3.1) as follows:

$$
J^h = \sum_{i=1}^{N} \Delta x_i G_i(u^M).
$$

(3.11)

To compute the gradient, we will have to choose a discrete space and an inner product on that space. The space we consider is the Euclidean space with the inner product $(x, y) = x^T Q y$, where $Q$ is a positive-definite weight matrix (in this case corresponding to the cell size $\Delta x_i$). Finally, the gradient is given by:

$$
\nabla J^h = p^0.
$$

Note that in the discrete case, differentiability issues come into play in two ways:

- Firstly, they are inherited from the continuous cases. Indeed, if the gradient of the cost function is not well-defined in the continuous case, we cannot hope to achieve convergence in the discrete case.

- Secondly, further issues are introduced by way of the numerical flux approximations. Most commonly used numerical schemes will have non-differentiable fluxes (due to upwinding, limiters, etc).

In the following, we will study the effect of differentiability of the fluxes, to what extent it can be ignored and if complete vs. incomplete differentiation of the fluxes improves the final result significantly.

Another issue that is not obvious is whether the discrete adjoint will converge to the exact solution of the continuous adjoint (which can be easily determined in 1D for piecewise constant profiles). This is an issue
that is only present in the discretize-then-differentiate approach to adjoint methods, since in the opposite case, a vast majority of numerical schemes will be able to solve the advection equation backwards in time to the same degree of accuracy as a forward solve.

### 3.1.3 Numerical Results

**Upwind Scheme**

For the numerical part, we will be testing two classic flux choices for Finite Volume methods. The first one is the upwind flux, which is given by:

\[ f_{i-1/2}^m(u_{i-1}^m, u_i^m) = (1 - s_{i-1/2}^m)(cu_{i-1}^m) + s_{i-1/2}^m(cu_i^m), \]

where \( s \) is an upwinding coefficient given by:

\[ s_{i-1/2}^m = \begin{cases} 
0, & \text{if } c > 0, \\
1, & \text{if } c \leq 0.
\end{cases} \]

or

\[ s_{i-1/2}^m = \frac{1}{1 + \exp(c/\delta)}, \quad (3.12) \]

where \( \delta \ll 1 \). This formulation can be directly generalized to cases where \( c \) is \( c(t, x) \), a function of both space and time, with an appropriate discretization. We note that with the second upwinding coefficient, the upwind flux can be made completely differentiable.

**Flux Limited Scheme**

The second scheme we will be looking at is a flux limited scheme of the form:

\[ f_{i-1/2}^m = f_{i-1/2}^{LO} + \phi(r_{i-1/2}^m)(f_{i-1/2}^{HI} - f_{i-1/2}^{LO}), \]

i.e. a convex combination of a high-order HO flux and a low-order LO flux. A classic choice for the low-order flux is the upwind flux, as defined above, which is a first-order flux. For the high-order flux, we will choose the second-order Lax-Wendroff flux:

\[ f_{i-1/2}^{HO} = \frac{1}{2}(u_i^m + u_{i-1}^m) + c\frac{\Delta t}{2\Delta x}(u_i^m - u_{i-1}^m). \]

The main ingredient in flux limited scheme is the limiter \( \phi \) itself. The limiter is constructed in such a way that it detects discontinuities and forces the limited flux to degenerate into the low-order, non-oscillatory flux.
in those regions, so as to avoid spurious oscillations. We will be looking at two limiters: the non-differentiable \textbf{minmod} limiter:

\[
\phi(r) = \max(0, \min(1, r))
\]

and the differentiable \textbf{van Albada} limiter:

\[
\phi(r) = \frac{r^2 + r}{r^2 + 1}, \tag{3.13}
\]

where the slope ratio \( r \) is given by:

\[
r \equiv r_{i-1/2}^{m}(u_{i-2}^{m}, \ldots, u_{i+1}^{m}) = (1 - s_{i-1/2}^{m}) \frac{u_{i}^{m} - u_{i-2}^{m}}{u_{i}^{m} - u_{i-1}^{m}} + s_{i-1/2}^{m} \frac{u_{i+1}^{m} - u_{i}^{m}}{u_{i}^{m} - u_{i-1}^{m}}.
\]

The limited family of fluxes allows for more experimentation with nonlinear numerical schemes. We will take advantage of this by looking at:

- complete differentiation, noting that the flux is a function of \( u_{i-2}^{m} \) to \( u_{i+1}^{m} \), some of which only appear in the limiter and slope ratio. This is, of course, only possible when using the van Albada limiter, since the minmod limiter is not differentiable.

- incomplete differentiation, where we will not differentiate the flux limiter (i.e. it will be considered “constant” in this case).

\textbf{Test Case: Boxcar Function}

The test case we will look at has an initial condition consisting of a simple boxcar function, defined as:

\[
g(x) = \begin{cases} 
1, & x \in [0.25, 0.75], \\
0, & \text{otherwise},
\end{cases} \tag{3.14}
\]

on the domain \( \Omega = [0, 1] \) with periodic boundary conditions. We will take a velocity \( c = 1 \), a final time \( T = 1 \) and a CFL condition of 0.9 for the following tests. Furthermore, we take a cost functional like in (3.1) with:

\[
G(u) = \frac{1}{2} u^2 \implies G'(u) = u,
\]

so that the starting condition at time \( t = T \) of the adjoint is simply the solution of the forward equation at the final time. Regarding this as an optimization problem, the optimal solution for the given \( G \) is trivially
Using the gradient from (3.6), we formulate the following steepest descent update:

\[ g^{(k+1)} = g^{(k)} - \rho \nabla g J(g^{(k)}) = g^{(k)} - \rho p^{(k)}(0, x). \]

Note that, for our boxcar function, if we advect the initial guess \( g^{(0)} \) exactly both forward and backward and take \( \rho = 1 \), we can converge to the optimal solution in exactly one iteration. We will judge the following numerical results with this in mind.

The results using the upwind scheme can be seen in Figure 3.1. For a constant \( c \), the upwind scheme is essentially linear, so we find that the forward and adjoint discrete schemes are exactly the same. Furthermore, even with a time and space dependent velocity, the upwind scheme will only have issues in cells which are sources or sinks (see [Liu, 2008]).

![Figure 3.1](image)

Figure 3.1: Forward solution at time \( t = T \) (left) and adjoint solution at time \( t = 0 \) (right) for different number of cells \( N \).

Next we will look at the flux limited scheme in the two following cases:

- Using the minmod limiter and performing an incomplete differentiation and a “complete” differentiation using finite difference. This can be seen in Figure 3.2.
- Using the van Albada limiter and performing an incomplete differentiation and a complete differentiation. This can be seen in Figure 3.3.

In Figure 3.2 we have results using the minmod limiter, that is a priori non-differentiable. We can see that, unlike the upwind scheme, the flux limited scheme develops spurious oscillations around the interface where the limiter is actually active. The main difference between a completely and incompletely differentiated minmod-based scheme is that with the completely differentiated scheme we can see oscillations on the other
Figure 3.2: Adjoint solution using the incompletely differentiated flux with the minmod limiter (left) and a finite difference approximation (right).

side of the discontinuity, reminiscent of typical oscillations we see with the Lax-Wendroff scheme.

Figure 3.3: Adjoint solution using the incompletely (left) and completely (right) differentiated flux with the van Albada limiter.

We can see the same results in the case of the van Albada limiter in Figure 3.3. One important difference is that the oscillations arising with the van Albada limiter seem to be larger than the ones from the minmod limiter, likely due to the fact that the minmod is exactly at the lower limit of the TVD region, while the van Albada limiter is larger for all $r$. In Table 3.1 we have order estimates for the schemes we have seen in this section. The error is computed using the discrete $L^1$ norm:

$$E = \sum \Delta x |u_i^{M} - u(t^M, x_i)|,$$

In the $L^1$ norm, the expected order of convergence for monotone schemes with discontinuous solutions is 0.5
(see [Tang, 1995], for example). The results seem to indicate that there are some subtle differences between complete and incomplete differentiation: complete differentiation seems to give a larger order, while having larger errors as well. However, the differences seem to be small enough to conclude there is no large impact that can be obtained by completely differentiating a nonlinear scheme. Note that it is important to carefully analyze the scheme to see which terms may lead to large gradients and have to be included in the end result. We will have further cause to analyze this problem in Chapter 4 for other highly nonlinear schemes.

Table 3.1: Boxcar Test: $L_1$ error log results for the adjoint variable $p$.

<table>
<thead>
<tr>
<th></th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>Adjoint Order</th>
<th>State Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upwind</td>
<td>-2.73941</td>
<td>-3.10127</td>
<td>-3.45567</td>
<td>0.514</td>
<td>0.570</td>
</tr>
<tr>
<td>minmod Incomplete</td>
<td>-3.28800</td>
<td>-3.72856</td>
<td>-4.16831</td>
<td>0.632</td>
<td>0.795</td>
</tr>
<tr>
<td>minmod Complete</td>
<td>-3.10148</td>
<td>-3.56381</td>
<td>-4.04178</td>
<td>0.675</td>
<td>0.795</td>
</tr>
<tr>
<td>van Albada Incomplete</td>
<td>-3.32044</td>
<td>-3.73063</td>
<td>-4.12307</td>
<td>0.576</td>
<td>0.854</td>
</tr>
<tr>
<td>van Albada Complete</td>
<td>-2.99763</td>
<td>-3.40805</td>
<td>-3.83301</td>
<td>0.600</td>
<td>0.854</td>
</tr>
</tbody>
</table>

3.2 Generalized Tangent Vectors

In the previous section, we have seen that, analytically, the discontinuities in the advection equation pose no particular problem. Similarly, numerically, the main problems that arise are due to linearizing highly nonlinear fluxes and not from a difficulty in capturing the discontinuity.

In this section we will look at the same problem in the context of scalar nonlinear conservation laws of the form:

\[
\begin{cases}
\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, & (t, x) \in [0, T] \times \Omega, \\
u(0, x) = g(x), & x \in \Omega.
\end{cases}
\] (3.15)

Similarly to the advection case, we will consider the following perturbation to the control variable $g$ providing the initial condition:

\[g'(x) = g(x) + \epsilon g'(x) + o(\epsilon),\]

which implies, under some regularity assumptions, that the solution at time $t$ can also be decomposed as follows:

\[u'(t, x) = u(t, x) + \epsilon u'(t, x) + o(\epsilon),\]
where \( u'(t, x) \) satisfies the linearized system:

\[
\begin{align*}
\frac{\partial u'}{\partial t} + \frac{\partial}{\partial x} (a(u)u') &= 0, \\
(t, x) &\in [0, T] \times \Omega, \\
\end{align*}
\]

\[u'(0, x) = g'(x), \quad x \in \Omega,
\]

\[3.16\]

where

\[a(u) = \frac{df}{du}(u).\]

Thus far, the steps we have taken are completely analogous to the advection equation. However, they are no longer valid after \( u \) develops shocks. Intuitively, this can be understood through the idea that the perturbation no longer occurs only in the solution itself, but also in the shock position and velocity (given by the Rankine-Hugoniot conditions) which depend on the solution. Formally, this implies that the approximation no longer converges in the standard \( L_1 \) norm, like it did for the advection equation.

Example 3.2. To showcase this issue, we will look at Burgers’ equation, where the flux is given by:

\[f(u) = \frac{1}{2} u^2 \quad \text{and} \quad a(u) = \frac{df}{du}(u) = u.
\]

\[3.17\]

For simplicity, we will not take the same initial condition as for the advection equation, since it will give rise to a rarefaction wave on the left-hand side besides the shock on the right-hand side. A simpler perturbed initial condition is given by:

\[u'(0, x) = (1 + \epsilon)x\chi_{[0,1]}(x),
\]

which gives the following solution at time \( t > 0 \):

\[u'(t, x) = \frac{(1 + \epsilon)x}{1 + \frac{t}{\sqrt{1 + (1 + \epsilon)^2}}} \chi_{[0, \sqrt{1 + (1 + \epsilon)t}]}(x),
\]

where \( \sqrt{1 + (1 + \epsilon)t} \) is the perturbed shock position. At time \( t = 0 \), we have that:

\[u'(0, x) = \lim_{\epsilon \to 0} \frac{1}{\epsilon} (u'(0, x) - u(0, x)) = x\chi_{[0,1]}(x),
\]

is well defined in \( L_1 \). However, for \( t > 0 \), the limit:

\[\lim_{\epsilon \to 0} \frac{1}{\epsilon} (u'(t, x) - u(t, x))
\]

is no longer well defined in \( L_1 \). A possible solution to the above can still be computed as a weak limit in...
a measure space whose absolutely continuous part (with respect to the Lebesgue measure) is given by the
measure:
\[ d\mu(t,x) = \frac{x}{(1+t)^2} \chi_{[0,\sqrt{1+t}]}(x) \, dx \]
and the additional part is a point mass at \( x = \sqrt{1+t} \) of magnitude:
\[ \| u \| \xi = \frac{t}{2(1+t)}, \]
where \( \| u \| \) is the jump in \( u \) across the shock and \( \xi \) is the limit of:
\[ \xi = \lim_{\epsilon \to 0} \frac{x^\epsilon(t) - x(t)}{\epsilon}, \]
where \( x(t) \) is the shock position at time \( t \). A full derivation of this result can be found in Appendix A.

Given the example above, we can see that a first order variation of \( u \) can only be defined in a weak sense
and it depends on both the actual solution and a perturbation of the shock position. The duo \( (\mu, \xi) \in L^1 \times \mathbb{R} \)
is what is known as a generalized tangent vector. This concept has been introduced by [Bressan, 1995] with
an application to the more general case of systems of conservation laws. The work from [Bressan, 1995] is
still constrained to one dimensional problems, but the formalism has been extended to 2D conservation laws
in [Baeza, 2009].

We will now proceed to give a formal definition and characterization of generalized tangent vectors,
following [Bressan, 1995] closely. First, we need to introduce the parametrization \( \gamma : [0,\epsilon_0] \to L^1 \), where
\( \gamma(\epsilon) = u^\epsilon \) is known as a path. Two such paths are called equivalent if:
\[ \lim_{\epsilon \to 0} \frac{1}{\epsilon} \| \gamma(\epsilon) - \gamma'(\epsilon) \|_{L^1} = 0. \]

**Theorem 3.1** (Definition 1 in [Bressan, 1995]). Let \( u : \Omega \to \mathbb{R} \) be piecewise Lipschitz continuous with jumps
at points \( x_i \), for \( i \in [1,N] \). We define the space of generalized vectors to \( u \) as \( T_u = L^1 \times \mathbb{R}^N \). A continuous
path \( \gamma \) is said to generate the tangent vector \((v, \xi) \in T_u\) if it is equivalent to the path defined by:
\[ \gamma(v;\xi;u) = u + \epsilon v + \sum_{\xi_i < 0} \| u \| \chi_{[x_i, x_i + \epsilon \xi_i]} - \sum_{\xi_i > 0} \| u \| \chi_{[x_i, x_i + \epsilon \xi_i]} \].

The generalized tangent vectors obtained this way are necessarily unique. We have seen this in our exam-
ple as well, which allowed for a single limit in some measure space. A similar definition for a special class of
functions with simple discontinuities, meaning discontinuities where the jump is either a contact discontinu-


ity or a single shock (this distinction is more important for systems of conservation laws, see [Bressan, 1995]), is given:

**Theorem 3.2** (Lemma 3.1 in [Bressan, 1995]). The path \( \gamma \) generates the tangent vector \( (v, \xi) \) if and only if:

\[
\lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{\sigma}^{\eta} |u'(\xi_i^\epsilon - y) - u(x_i + y) - v(x_i + y) - \xi_i \partial_x u(x_i + y)| \, dy = 0,
\]

where the interval \([x_i - \sigma, x_i + \eta]\) for \( \sigma, \eta > 0 \) does not contain any other discontinuities.

Using these two theorems we can properly derive the linearized and adjoint equations for a scalar conservation law like Burgers’ equation. We will see how this can be done in the next section.

### 3.3 The Inviscid Burgers’ Equation

In this section we will focus specifically on Burgers’ equation in the simple case where we only have one shock. The assumption that we only have a single shock is done to simplify the derivation of the linearized and adjoint equations, since we have seen that generalized tangent vectors are defined for any number of shocks [Bressan, 1995; Ulbrich, 2001].

Given their additional complexity and practical use, conservation laws have been extensively studied in the literature. Older papers such as [Bardos, 2003; Giles, 2002; Homescu, 2003] have laid the foundations of how we approach these problems formally. A very exhaustive study, both analytical and numerical, is given in [Ulbrich, 2001], where a calculus was developed using concepts of *shift-differentiability*, that has been used to prove the Fréchet differentiability of cost functionals for conservation laws with source terms. Furthermore, conditions for the correct convergence of the discrete equations to their continuous counterparts were given. A recent extension from the same author can be found in [Giles, 2010a; Giles, 2010b]. This paper formally proves the convergence of a simple modified Lax-Friedrichs for all the quantities of interest in adjoint-based optimization: the linearized equations, the adjoint equations and the cost functional. We will describe here some of these developments in the case of Burgers’ equation with a single shock.

In the following, we will analyze Burgers’ equation as defined in (3.15) with the flux given by (3.17). Additionally, at the shock position \( x_s \), we have the Rankine-Hugoniot condition:

\[
\dot{x}_s \llbracket u \rrbracket = \llbracket f(u) \rrbracket, \quad \text{on } \Gamma,
\]

where \( \Gamma = \{(t, x_s(t)) \mid t \in [0, T]\} \) is the shock trajectory.
The linearized equations are also given by (3.16) with the flux derivative from (3.17). We are left with linearizing the Rankine-Hugoniot condition. Since we know we are in the vicinity of a shock, we can use Theorem 3.2 to obtain:

\[
\dot{\xi}_s [u] + \dot{x}_s [v + \xi_s u_x] = [a(u)v + \xi_s f(u)_x], \quad \text{on } \Gamma,
\]

(3.19)

where \((v, \xi_s)\) represent the small perturbation to the state variables \((u, x_s)\). More explicitly, on the left-hand side we have applied the product rule and then we have the variation of \(u\) given by \(v + \xi_s u_x\) and on the right-hand side we have applied Theorem 3.2 to obtain a first order approximation for \(f(u)\) in the same way as we did for \(u\). The linearized Rankine-Hugoniot condition can be manipulated into a simpler form of a ODE (see [Giles, 2010a]):

\[
\frac{d}{dx} (\xi_s [u]) = [a(u) - \dot{x}_s]v.
\]

Next we will look at the linearized cost functional from (3.1). Since we know the shock position, we can split the cost functional as follows:

\[
\mathcal{J}(u, g) = \int_a^{x_s-} G(u(T, x)) \, dx + \int_{x_s+}^b G(u(T, x)) \, dx.
\]

As such, the directional derivative, in the direction of a tangent vector \((v, \xi_s)\) is given by:

\[
\frac{d\mathcal{J}}{du}(u; v, \xi_s) = \int_a^{x_s-} G'(u(T, x)) v \, dx + \int_{x_s+}^b G'(u(T, x)) v \, dx - \xi_s [G(u(T, x_s))],
\]

(3.20)

with an application of the Reynolds transport theorem to get the last term. Next, we will construct the Lagrangian so that we can derive the adjoint equations. It is obtained by putting together our PDE, its initial condition and the Rankine-Hugoniot condition to form:

\[
\mathcal{L}((u, x_s), (p, y_s), g) = \mathcal{J}(u) - \int_{[0, T] \times \Omega \setminus \Gamma} p \left( \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) \right) \, dx \, dt - \int_{\Omega} p(0, x)(u(0, x) - g) \, dx - \int_{\Gamma} y_s(\dot{x}_s [u] - [f(u)]) \, dt,
\]

where \((p, y_s)\) are the Lagrange multipliers or the adjoint variable and the adjoint shock position. A complete derivation of the adjoint equations can be found in Appendix B. They are:

\[
\begin{cases}
- \frac{\partial p}{\partial t} - a(u) \frac{\partial p}{\partial x} = 0, & (t, x) \in ([0, T] \times \Omega) \setminus \Gamma, \\
p(T, x) = G'(u(T)), & x \in \Omega \setminus \{x_s(T)\}.
\end{cases}
\]

(3.21)
At the shock position we have the following equations:

\[
\frac{dy_s}{dt} = 0, \quad p = -y_s, \quad \text{and} \quad y_s(T) = -\frac{\|G(u(T, x_s))\|}{\|u(T, x_s)\|}. \tag{3.22}
\]

The adjoint equation from (3.21) is also a backward in time advection equation with the velocity \(-a(u)\). Unlike in the case of the advection equation and its adjoint, the coefficient in this case is \(a(u) = u\) which is both time and space dependent. Furthermore, it is discontinuous since \(u\) has shocks. The problem of advection with discontinuous coefficients is also very difficult to treat formally. Some successful attempts have been made in [Bouchut, 1998; Gosse, 2000], which proved the existence and uniqueness of solutions to such equations.

The equations obtained for the adjoint Rankine-Hugoniot conditions are also very important. We notice that at the shock position \(x_s\) we actually have that \(p = -y_s\) and that \(y_s = 0\), which implies that the adjoint equations will always be constant along the shock. Also note that the terminal condition for the adjoint shock position tends to the derivative of \(G\) when the shock is infinitesimal:

\[
\frac{\|G(u(T, x_s))\|}{\|u(T, x_s)\|} \rightarrow \frac{\partial G}{\partial u}.
\]

These analytical results give us very important pointers as to how a numerical scheme would have to behave to achieve convergence to the continuous solution. Namely, it would need to either impose \(p = -y_s\) at the shock position or make sure that the gradient of \(G\) is correctly captured. The second condition could be achieved by smearing the shock across an increasing number of cells as proved in [Giles, 2010a; Giles, 2010b]. The idea of directly imposing the value at the shock position has also been used in practice, for example in [Castro, 2008], to develop a new alternating descent algorithm (alternating between the two components of the tangent vectors) to great success.

**Example 3.3.** Before going to some numerical results, we will give a small example of how an analytical solution for the adjoint could be derived using the equations we have obtained. We take the following solution to Burgers’ equation (see Figure 3.4):

\[
u(t, x) = \begin{cases} 
1 & x < 0.5t, \\
0 & x \geq 0.5t,
\end{cases}
\]

with \(G(u) = f(u)\) and a final time \(T = 1\). The shock position is initially at \(x_s = 0\) and advances with a
velocity:
\[
\dot{x}_s = \frac{[f(u)]}{\|u\|} = \frac{u(t,x_s(t)_+) + u(t,x_s(t)_-)}{2} = \frac{1}{2},
\]
which gives an ODE we can solve analytically to get the shock position as a function of time:

\[
x_s(t) = \frac{1}{2} t.
\]

The adjoint to the shock position is given by:

\[
y_s(t) = -\frac{\|G\|}{\|u\|} = -\frac{\|f\|}{\|u\|} = -\frac{1}{2},
\]
because it is constant in time. Using this as a boundary condition for the adjoint equation at \(x_s(t)\), we can compute the exact solution to (3.21) with the final state \(p(T,x) = u(T,x)\). We notice that the adjoint equation is an advection equation with a velocity of \(-1\) to the left of the shock and a velocity of \(0\) to the right. Normally, this would leave an unknown region as the two sides separate, but we know that \(p = 0.5\) on all the characteristics going into the shock. Finally, we can find the solution (see Figure 3.4):

\[
p(t,x) = \begin{cases} 
1 & x < -0.5 + t, \\
0.5 & -0.5 + t < x < 0.5, \\
0 & x > 0.5.
\end{cases}
\]

Figure 3.4: Exact state solution at time \(t = 1.0\) and exact adjoint solution at time \(t = 0\) for Example 3.3.
3.3.1 Spatial Consistency

In this section we will proceed to analyze a number of spatial discretizations with respect to some of the same metrics we have used for the advection equation: convergence of the adjoint to known exact solutions, complete vs. incomplete differentiation, etc. In the derivation of the adjoint equations (3.21) and the adjoint Rankine-Hugoniot conditions (3.22), we have seen that there are two ways to approach the numerical resolution for conservation laws. Firstly, we can impose a boundary condition at the shock position and, secondly, we can correctly capture the adjoint shock position at time $T$, i.e. capture:

$$y_s = \frac{G}{\|u\|}$$

In this section we will investigate the second case. That is to say, we will study the convergence of different well-known schemes without prior modification to see under which conditions convergence is achieved. A more thorough investigation can be found in [Fikl, 2016].

Burgers’ equation can be discretized using a Finite Volume method in the exact same way as the advection equation and we will preserve all the previous notation from (3.7)-(3.10) and before, without going into detail.

Modified Lax-Friedrichs Scheme

The first scheme we will look at is the modified Lax-Friedrichs scheme from [Giles, 2010a; Giles, 2010b]. This scheme will act as a baseline for comparison since it has been proven to correctly converge to the weak solution of the adjoint equation, i.e. correctly approximate $p = y_s$ at the shock position $x_s$. The flux for this scheme may be written as:

$$f_{i-\frac{1}{2}}^m = \frac{1}{2}(f(u_i^m) + f(u_{i-1}^m)) - \frac{\epsilon}{\Delta x}(u_i^m - u_{i-1}^m), \quad (3.23)$$

where $\epsilon = \Delta x^\alpha$, with $\alpha \in (2/3, 1)$, is a grid-dependent parameter that will smear the shock over an increasing number of cells as $\Delta x \to 0$. The proposed scheme has an order of $O(\alpha)$ and is stable under the CFL condition:

$$\max_i |u_i^m| \epsilon \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}.$$

We will now briefly look at the convergence and the order of accuracy for the case of the modified Lax-Friedrichs scheme (3.23) and its adjoint. For testing purposes, we take a cost functional with $G(u) = u^2/2$
and the following exact solution with a single initial shock at \( x_s(0) = 0 \):

\[
    u(t,x) = \begin{cases} 
        1.5, & x < 0.5t, \\
        -0.5, & x > 0.5t 
    \end{cases} 
\]  

(3.24)

and a known solution to the adjoint equation (3.21) for \( T = 1 \):

\[
    p(t,x) = \begin{cases} 
        1.5, & x < -1 + 1.5t, \\
        0.5, & -1 + 1.5t < x < 1 - 0.5t, \\
        -0.5, & x > 1 + 0.5t, 
    \end{cases} 
\]  

(3.25)

where the middle state in the adjoint is computed from \( \|G(u)\|/\|u\| \) (see [Giles, 2002]) as in Example 3.3.

We also take \( T = 1.0 \), \( \Omega = [-1.5,1.5] \), and a CFL of 0.9. We can see in Table 3.2 that the forward scheme converges with the expected order \( \alpha \), while the adjoint equation converges with order \( \alpha/2 \). This corresponds to the expected order of convergence of a linear conservation law with discontinuous coefficients, i.e. \( k/k + 1 \), where \( k \) is the order of the scheme obtained assuming continuous solutions and coefficients.

<table>
<thead>
<tr>
<th></th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State</strong> ( \alpha = 0.999 )</td>
<td>-2.77028</td>
<td>-3.42418</td>
<td>-4.15565</td>
<td>0.999</td>
</tr>
<tr>
<td><strong>Adjoint</strong> ( \alpha = 0.999 )</td>
<td>-1.37710</td>
<td>-1.49982</td>
<td>-1.85148</td>
<td>0.514</td>
</tr>
<tr>
<td><strong>State</strong> ( \alpha = 0.9 )</td>
<td>-2.37230</td>
<td>-2.98675</td>
<td>-3.61683</td>
<td>0.897</td>
</tr>
<tr>
<td><strong>Adjoint</strong> ( \alpha = 0.9 )</td>
<td>-0.91325</td>
<td>-1.21177</td>
<td>-1.52599</td>
<td>0.441</td>
</tr>
<tr>
<td><strong>State</strong> ( \alpha = 0.8 )</td>
<td>-1.99073</td>
<td>-2.54112</td>
<td>-3.09571</td>
<td>0.797</td>
</tr>
<tr>
<td><strong>Adjoint</strong> ( \alpha = 0.8 )</td>
<td>-0.72734</td>
<td>-0.96850</td>
<td>-1.24296</td>
<td>0.372</td>
</tr>
</tbody>
</table>

We have used the following discrete \( L^1 \) norm for the error estimates:

\[
    ||e_\ast||_1 = \sum \Delta x |u^M_i - u(T,x)|, 
\]

which aggregates both time and space errors. The result of the simulation for various number of cells can be seen in Figure 3.5. Most importantly, we can see that it correctly captures the boundary condition at the shock position.
Figure 3.5: Adjoint solution at $t = 0$ for (3.25) (left) and a zoom around the shock position at $x = 0$ (right).

**Upwind Scheme**

We will now look at another first-order scheme which is widely used in practice: the upwind scheme. For Burgers’ equation, the upwinded flux is given by:

$$f^m_{i-\frac{1}{2}} = (1 - s^m_{i-\frac{1}{2}}) f(u^m_i) + s^m_{i-\frac{1}{2}} f(u^m_{i-1}),$$  (3.26)

where the upwinding coefficient is given by (3.12) using $u^m_{i-\frac{1}{2}}$. The approximation of the face velocity we have used is:

$$u^m_{i-\frac{1}{2}} = \frac{u^m_i + u^m_{i-1}}{2}.$$

The test case is the same simple example we have used for the modified Lax-Friedrichs scheme from (3.24)-(3.25) with the differentiable upwinding coefficient from (3.12) and a CFL of 0.4.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>81</th>
<th>243</th>
<th>729</th>
<th>Adjoint Order</th>
<th>State Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta = 1.0$</td>
<td>-1.23690</td>
<td>-1.61394</td>
<td>-1.68173</td>
<td>0.202</td>
<td>1.000</td>
</tr>
<tr>
<td>$\delta = 0.1$</td>
<td>-1.25341</td>
<td>-1.80489</td>
<td>-1.78244</td>
<td>0.299</td>
<td>0.999</td>
</tr>
<tr>
<td>$\delta = 0.01$</td>
<td>-1.13373</td>
<td>-1.83637</td>
<td>-1.75990</td>
<td>0.294</td>
<td>0.999</td>
</tr>
<tr>
<td>$\delta = 0.001$</td>
<td>-1.12231</td>
<td>-1.80167</td>
<td>-1.75728</td>
<td>0.293</td>
<td>0.999</td>
</tr>
<tr>
<td>$\delta = 0.0$</td>
<td>-1.12238</td>
<td>-1.80161</td>
<td>-1.75728</td>
<td>0.293</td>
<td>0.999</td>
</tr>
</tbody>
</table>

The results can be seen in Table 3.3 for various values of $\delta$. For values of $\delta \geq 1$, the upwind scheme degenerates to a centered (unstable) scheme, and the adjoint scheme becomes unstable as well. However, for
values < 0.1 (even the discontinuous case of δ = 0) there are no significant differences in convergence order and error. Results can also be seen in Figure 3.6 for δ = 0.001 and compare qualitatively well with those obtained in [Alauzet, 2012]. Unlike the modified Lax-Friedrichs scheme, the upwind scheme does not seem to exhibit mesh-convergence and only correctly captures the boundary conditions at the shock position for a very specific combination of mesh points, CFL condition, δ, etc. An example of this behavior is also given in [Giles, 2002], proving that it is easy to construct cost functionals, by choosing G, such that the numerical scheme does not capture the jump correctly and, thus, the adjoint does not converge.

Figure 3.6: Adjoint solution at time t = 0 using the Upwind scheme.

The importance of smearing the shock across an increasing number of cells can even be seen with the modified Lax-Friedrichs scheme that normally converges to the correct solution. For a simple test, we will use the known exact solution (3.24) to solve the adjoint equation. Using the exact solution means that the shock is not smeared at all during the forward solve and the result can be clearly seen in Figure 3.7. So, even for a converging scheme as the modified Lax-Friedrichs scheme, if the shock is not sufficiently smeared during the forward solve, the adjoint will not converge to the correct solution for the same reasons as the upwind flux.

Flux Limited Schemes

We have seen that for the Upwind scheme, having a completely differentiable flux (for δ > 0) made a very small difference in the convergence and convergence order of the adjoint solution. To further investigate this issue, we will look at a family of flux limited schemes, like we have in the case of the advection equation.
Figure 3.7: Adjoint solution at $t = 0$ for (3.25) using the modified Lax-Friedrichs scheme and the exact solution (3.24) for the state variable $u(t, x)$.

The flux is given by a combination of a low-order and a high-order flux:

$$f_{i-1/2}^m(u_i^m, u_{i-1}^m, s_{i-1/2}^m, r_{i-1/2}^m) = f_{i-1/2}^{LO} + \phi \left( r_{i-1/2}^m \right) \left( f_{i-1/2}^{HI} - f_{i-1/2}^{LO} \right),$$

(3.27)

where we choose the upwind flux from (3.26) as the low-order flux and the Lax-Wendroff flux as the high-order (second-order, to be exact) flux given by:

$$f_{i-1/2}^{HI} = \frac{1}{2} \left( |f(u_i^m) + f(u_{i-1}^m)| + \frac{\Delta t}{\Delta x} u_{i-1/2}^m [f(u_i^m) - f(u_{i-1}^m)] \right).$$

(3.28)

For testing purposes, we select the differentiable van Albada limiter given by (3.13). As in the case of the advection equation, we will investigate completely and incompletely differentiating the flux. Incomplete differentiation will be done assuming that the upwinding coefficient $s_{i-1/2}^m$ and the slope ratio $r_{i-1/2}^m$ do not depend explicitly on the solution $u_i^m$. A previous study [Alauzet, 2012] suggests that the discrepancies obtained by incomplete differentiation are expected to remain small. However, the tests in [Alauzet, 2012] have been performed on the more complicated 2D Euler equations and may hide some subtleties.

The results for our test can be seen in Table 3.4. It is clear that the incomplete differentiation of the van Albada limiter has essentially the same problem as we have seen with the upwind scheme and does not converge to the exact solution (see also Figure 3.8). However, the completely differentiated scheme does not seem to behave any better. Finally, neither exhibits grid convergence properties at the intermediate state.
around the shock. In a way, we can consider this as validation of the results from [Alauzet, 2012] which also mention that complete vs. incomplete differentiation does not have a large impact on the end result.

A few points need to be made regarding the convergence of flux limited schemes of this type. One of the main ingredients in convergence proofs regarding nonlinear conservation laws with shocks is Oleinik’s One-Sided Lipschitz Condition (OSLC):

\[
\frac{u(t,x) - u(t,y)}{x - y} \leq \frac{1}{t}
\]

and its discrete equivalent. See, for example, [Bouchut, 1998; Gosse, 2000; Castro, 2008; Ulbrich, 2001] for general proofs that heavily involve the OSLC. The only limiter that is known to satisfy the OSLC is the maxmod limiter [Brenier, 1988], which, unfortunately, does not remove spurious oscillations from the solution (specifically, it allows undershoots), rendering it unacceptable in practice. In [Ulbrich, 2001], an iterative method is proposed to ensure that a given slope, initially limited with any commonly used limiter, is modified in such a way that the scheme satisfies the OSLC. However, this makes the limiter itself overly complicated

| Table 3.4: van Albada Limited Scheme: $L_1$ error log results for the adjoint variable $p$. |
|----------------------------------|---|---|---|---|---|
|                                 | 81 | 243 | 729 | Adjoint Order | State Order |
| Complete. CFL = 0.4             | -1.32960 | -1.66087 | -2.24946 | 0.418 | 0.999 |
| Complete. CFL = 0.2             | -1.47540 | -1.54951 | -1.89077 | 0.338 | 0.999 |
| Incomplete. CFL = 0.4           | -1.02991 | -1.58054 | -1.57674 | 0.248 | 0.999 |
| Incomplete. CFL = 0.2           | -0.89631 | -0.80670 | -1.38451 | 0.222 | 0.999 |

Figure 3.8: Adjoint solution at $t = 0$ for (3.25) using complete differentiation (left) and incomplete differentiation (right) of the flux limited scheme.
(both implementation-wise and from a computation overhead point of view) and, thus, not practical. Even though there are no formal proofs, [Nessyahu, 1994] suggests that TVD MUSCL schemes are bounded in $Lip^+$, even though they are not monotonically decreasing, as required by the OSLC, leading to convergence in certain cases.

Another important consideration to be made in practice is the size of the shocks. Since the oscillations present in, e.g., Figure 3.8 are always bounded and generally close to the correct value $p = y_s$, for weak shocks the error suffered from these inconsistencies is unlikely to matter much and still lead to acceptable gradients.

### 3.3.2 Temporal Consistency

In this section we will investigate how to properly derive the adjoint of a time integration scheme and what effects different choices in the discretization will have. Work on adjoint integration schemes has been done in the past, notably for Runge-Kutta methods in [Hager, 2000].

**First-Order Forward Euler Method**

The first scheme we will investigate is the Forward Euler method we have been using this far. The spatial discretization can be given by any of the previous scheme, but we choose the modified Lax-Friedrichs scheme from (3.23) to make sure that any issues we will see must come from the time integration.

To better understand the problem, we will look at a scalar ODE of the form:

$$\frac{dy}{dt} = f(t),$$

which, discretized using an explicit forward Euler integration method, is:

$$\frac{y^{m+1} - y^m}{\Delta t} = f(t^m).$$

We can also write the above equation as $Ly = f$. In the continuous case, the adjoint of $\partial_t$ is $-\partial_t$. However,
in the discrete case the time derivative operator \( L \) and its adjoint \( L^T \) are given by:

\[
L = \frac{1}{\Delta t} \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
-1 & 1 & 0 & \cdots & 0 \\
0 & -1 & 1 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & -1 & 1 \\
0 & \cdots & 0 & -1 & 1
\end{bmatrix} \quad \Rightarrow \quad L^T = \frac{1}{\Delta t} \begin{bmatrix}
1 & -1 & 0 & \cdots & 0 \\
0 & 1 & -1 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & 0 & 1
\end{bmatrix}
\]

We will now look at the \( m \)-th term of the dot products \( \langle x, Ly \rangle = \langle L^Tx, y \rangle \), where \( y \) is our variable and \( x \) its adjoint. We have:

\[
x^m \left( -\frac{1}{\Delta t} y^{m-1} + \frac{1}{\Delta t} y^m \right) = y^m \left( \frac{1}{\Delta t} x^m - \frac{1}{\Delta t} x^{m+1} \right),
\]

or

\[
x^m \left( \frac{y^m - y^{m-1}}{\Delta t} \right) = y^m \left( -\frac{x^{m+1} - x^m}{\Delta t} \right),
\]

which is also consistent with the property of the continuous operators. This equality shows that the \( n \)th term from \( \langle L^T x, y \rangle \) matches the “state” variable \( y \) at time \( t^m \) with the adjoint variable \( x \) advancing backwards in time from \( t^{m+1} \) to \( t^m \). A properly constructed adjoint time integrator is very important for convergence. If we were, for example, discretizing the continuous equations, the time at which the state variable is evaluated during the adjoint solve is more of a choice, but we see that discretely there is only one correct option. This can be seen in Figure 3.9, where the boundary condition at the shock position is no longer correctly captured if using an incorrect time discretization.

**Second-Order Runge-Kutta Methods**

While the issues regarding time integration of the discrete adjoint are rather obvious for the Forward Euler method used above, they are more subtle for high-order methods, such as the Runge-Kutta family of methods. In this section we will analyze two second-order Runge-Kutta methods and their adjoints, one explicit and one implicit method. Generally, a time integrator \( \phi_{\Delta t} \) has a classic definition of its adjoint [Hairer, 2006]:

\[
\phi_{\Delta t}^* = \phi_{-\Delta t}^{-1}, \quad (3.29)
\]
i.e. the adjoint is the inverse of the forward method with a negative time step. Using this definition, we can
deduce that any explicit scheme will have an implicit adjoint. This is the case for the results we have obtained
with the explicit Euler scheme in the previous section, albeit in a slightly convoluted way. Specifically, if the
original scheme is explicit, the adjoint scheme will be implicit when integrated forward in time, but explicit
when integrated backwards in time (see [Hager, 2000] for how to reverse time in a Runge-Kutta method).

To take advantage of different time integration methods, we first write the semi-discrete variant of (3.7)
as a coupled system of ODEs:

\[ \frac{d\mathbf{u}}{dt} = \mathcal{A}(\mathbf{u}), \]

where \( \mathcal{A} \) is the same as in (3.8), with the modified Lax-Friedrichs flux (3.23). The interdependence of the
ODEs originates from the fact that \( \mathcal{A} \) contains a non-local discrete spatial derivative. We can write the
system more explicitly, for the case of the 3 cell stencil in the modified Lax-Friedrichs scheme, as:

\[ \frac{du_i}{dt} = \mathcal{A}_i(u_{i-1}, u_i, u_{i+1}). \quad (3.30) \]

First we will look at the second-order explicit Runge-Kutta method known as Heun’s method (or the
explicit trapezoidal rule):

\[
\begin{align*}
\mathbf{u}^{m+1} &= \mathbf{u}^m + \frac{\Delta t}{2}(k_1^m + k_2^m), \\
k_1^m &= \mathcal{A}(\mathbf{U}_1), \quad \text{where } \mathbf{U}_1 = \mathbf{u}^m, \\
k_2^m &= \mathcal{A}(\mathbf{U}_2), \quad \text{where } \mathbf{U}_2 = \mathbf{u}^m + \Delta t k_1^m.
\end{align*}
\]

(3.31)
which corresponds to the Butcher tableau with \( b_1 = b_2 = \frac{1}{2}, c_1 = 0, c_2 = 1, a_{11} = a_{12} = a_{22} = 0 \) and \( a_{21} = 1 \).

The adjoint system we obtain (see [Hager, 2000]) is another explicit Runge-Kutta method, when integrated backwards in time, with a Butcher tableau where the \( a_{ij} \) matrix of coefficients has been transposed. The resulting system can be written as:

\[
\begin{cases}
\displaystyle p^m = p^{m+1} + \frac{\Delta t}{2} (l_1^{m+1} + l_2^{m+1}), \\
\displaystyle l_1^{m+1} = \left( \frac{\partial A}{\partial u^m}(U_1) \right)^T P_1, & \text{where } P_1 = p^{m+1} + \Delta t l_2^{m+1}, \\
\displaystyle l_2^{m+1} = \left( \frac{\partial A}{\partial u^m}(U_2) \right)^T P_2, & \text{where } P_2 = p^{m+1},
\end{cases}
\] (3.32)

For general formulations of the adjoint of an arbitrary discrete \( s \)-stage Runge-Kutta method see [Hager, 2000]. A very important property of the adjoint, as defined in [Hager, 2000], is that the order of the adjoint method is the same as that of the original method, even when the original method is explicit.

For comparison with Heun’s Method (3.32), we will investigate an implicit method defined using the same methodology from [Hager, 2000], namely the implicit midpoint method. The implicit midpoint method is a 1-stage second-order symmetric Runge-Kutta defined as:

\[
\begin{cases}
\displaystyle u^{m+1} = u^m + \Delta t k_1^m, \\ k_1^m = A(U_1), & \text{where } U_1 = u^m + \Delta t \frac{1}{2} k_1^m,
\end{cases}
\] (3.33)

and its adjoint is similarly given by:

\[
\begin{cases}
\displaystyle p^m = p^{m+1} + \Delta t l_1^m, \\
\displaystyle l_1^m = \left( \frac{\partial A}{\partial u^m}(U_1) \right)^T P_1, & \text{where } P_1 = p^{m+1} + \Delta t \frac{1}{2} l_1^m.
\end{cases}
\] (3.34)

Table 3.5: Runge-Kutta: \( L_1 \) error log results for the adjoint variable \( p\).

<table>
<thead>
<tr>
<th></th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>Adjoint Order</th>
<th>State Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heun’s Method</td>
<td>-1.04299</td>
<td>-1.39669</td>
<td>-1.74686</td>
<td>0.507</td>
<td>0.999</td>
</tr>
<tr>
<td>Implicit Midpoint Method</td>
<td>-1.04384</td>
<td>-1.39697</td>
<td>-1.74701</td>
<td>0.507</td>
<td>0.999</td>
</tr>
</tbody>
</table>

We are mainly interested in comparing the two methods in the context of time consistency. We have seen in the previous section (and Figure 3.9) that an inconsistent time stepping method may compromise the convergence of the numerical approximation to the correct solution. Error estimates for the two Runge-
Kutta methods we have presented can be seen in Table 3.5. Note that the space-time order of the method remains \( \approx 1 \) for the state equations and \( \approx 0.5 \) for the adjoint since the space discretization dominates at order \( \alpha \). This is not a major issue because we are simply interested in the consistency and the convergence of the adjoint Runge-Kutta methods and not in improving the accuracy.

We can see the same results in Figure 3.10 where both methods still converge to the exact solution around the shock position \( x = 0 \) at time \( t = 0 \). Similar convergence issues as in Figure 3.9 appear if the state variable is used at the inconsistent time \( u^{n+1} \) in the adjoint Runge-Kutta methods (3.32) and (3.34).

![Figure 3.10: Adjoint solution at \( t = 0 \) for (3.25) using the explicit Heun’s method (left) and the Implicit Midpoint method (right) with consistent time-stepping.](image)

### 3.4 Conclusions

This concludes our analysis of classical numerical schemes in the context of adjoint methods in the presence of discontinuities. The first issue we have looked at is the linear advection equation and the convergence of the discrete adjoint equations for various schemes. As expected, the upwind scheme performed very well for this case, since it is essentially linear for a constant velocity, like the continuous equations. Analyzing the nonlinear flux limited schemes has shown that this is not always the case. We have seen that when linearized, the limited flux is no longer “limited”, so the usual oscillations from the Lax-Wendroff scheme make their appearance around discontinuities. Even so, since the discontinuities were correctly advected by the adjoint scheme, we believe that better performing schemes can be developed, that do not exhibit such oscillations. Indeed, we will see some such non-oscillatory results in the next chapter.

The analysis of Burgers’ equation is considerably more involved than that of the advection equation. The main complication comes from the formation of shocks in finite time. We have seen that perturbation
of solutions which shocks cannot be accurately represented just by the variation of the solution itself, but we must also consider variations in the shock position to get a complete first-order approximation. This approximation is given by the theory of generalized tangent vectors. Using these generalized tangent vectors, we have shown that the adjoint equations have additional boundary conditions at the shock position, even though solutions are continuous for all characteristics going into the shock. We have studied in what scenarios these boundary conditions can be implicitly captured by numerical schemes. The main requirement for implicit convergence is given in [Giles, 2002] and refers to smearing the shock across an increasing number of cells as the grid gets refined (in the spirit of artificial viscosity methods).

A specially developed modified Lax-Friedrichs was shown in [Giles, 2010a; Giles, 2010b] to correctly capture the adjoint boundary conditions and converge to the correct solution. However, other classical schemes such as the previously mentioned upwind and flux limited schemes do not exhibit such qualities. Even if they do not converge, classical schemes still manage to capture the correct behaviour to a certain degree. We have found that for weak shocks, usage of such schemes can be acceptable and lead to small errors.

We have also looked at the time integrators used when solving Burgers’ equation. We have seen that, given a converging spatial discretization, the time integrator will not negatively influence the convergence. Even so, special care must be taken when defining the adjoint time integrator, since a definition that is discretely inconsistent will lead to a diverging scheme. The theory developed in [Hager, 2000] and others is complete for Runge-Kutta methods and gives the expected results, but no similar analysis seems to have been performed for multistep methods like Adams-Bashforth.
Chapter 4

Adjoint Equations for Anti-diffusive Schemes

We have seen in the previous chapter that adjoint convergence is possible, when dealing with conservation laws with certain types of discontinuities (contact discontinuities, shocks). However, the schemes we have used this far (upwind, MUSCL, Lax-Friedrichs) are not suited for interface capturing because of their diffusive nature. We are mainly interested in two-phase flows where the interface is identified by a marker function on a structured grid, in the vein of the advection tests performed in Chapter 3. We will shortly discuss here a few of the methods available for dealing with sharp interfaces. For a thorough rundown through possible approaches to dealing with such problems see, for example, [Prosperetti, 2009] for the numerical aspects and [Brennen, 2005] for more general modeling of multiphase flows.

One of the first methods that was developed for use in two-phase flows was the Volume-of-Fluid (VOF) method. In one dimension, the VOF principle is very simple. We know that our interface is a Heaviside function, which limits the values in a cell to the following averages: 1, if the cell is completely full, 0 if it is completely empty and an intermediate value if it contains the interface. This implies that, given a cell containing the interface and knowledge of which of its two neighbors is full and which is empty, we can exactly determine the jump position using only the average cell value. Once the exact position of the interface is determined, it can be advected exactly, with no diffusion. One of the first applications of this principle is given by the SLIC (Simple Line Interface Calculation) method [Noh, 1976], which is generalized to multiple dimensions using dimensional splitting. A first truly multidimensional VOF method was presented in [Hirt, 1981]. The method from Hirt and Nichols used a single interface orientation for all dimensions (determined using an approximation of the interface normal), so dimensional splitting was not needed. Since, many extensions to and variants of the VOF approach have appeared. Some notable ones are the PLIC (Piecewise Linear Interface Calculation) method from [Youngs, 1982], the fully multidimensional geometric VOF method of [Rider, 1998], the LVIRA (Least-squares Volume-of-Fluid Interface Reconstruction Algorithm) method from [Pilliod, 2004], etc. In the rest of this chapter we will analyze another type of VOF method which uses a smoothed Heaviside function to represent the interface, called the THINC method [Xiao, 2005].

VOF methods are by no means the only type of methods used for interface modeling. Some of their
inherent difficulties (complex geometrical algorithms, flotsam and jetsam, etc.) have led others to research alternatives. One of the notable alternatives is the family of Level Set methods developed in [Osher, 1988; Sussman, 1994] and subsequent papers. Unlike VOF-type methods, which attempt to directly advect the marker function of the interface, level set methods take a different route by modeling the interface using the 0 level set of a signed distance function (with the property that $|\nabla \phi| = 1$). The notable quality of such methods is that the level set function is continuous, so classical high-order methods can be used to advect it, without any spurious oscillations that naturally appear at the discontinuity of a Heaviside function. However, even with high-order methods, additional steps (such as reinitialization) need to be taken to maintain the properties of the signed distance function.

Another type of widely used methods are so-called front-tracking methods. To track the interface, these methods make use of explicit marker points that are not part of the usual computational grid, but layered on top of it and advected using Lagrangian methods. The ideas of front-tracking lead back to the work of Peskin on immersed boundary methods and Glimm [Glimm, 1982] on tracking shocks. Applications to two-phase flows are more recent, with the work from [Unverdi, 1992]. Much like the VOF methods, topological complexities (e.g. fusing boundaries) make front-tracking methods difficult to implement for many flows types.

A non-exhaustive list of other notable mentions in the field of anti-diffusive methods are the CIP (Constrained Interpolation Profile or Cubic Interpolated Propagation) methods [Yabe, 2001], phase-field methods [Jacqmin, 1999], the Vofire method [Després, 2010], etc.

4.1 THINC: Tangent of Hyperbola for Interface Capturing

Most of the methods we have mentioned previously, while very successful in the representation of sharp interfaces, have many features (e.g. geometrical reconstruction, high-order nonlinear schemes) that do not easily lend themselves to linearization, which is necessary for the discretize-then-differentiate approach to adjoint optimization. For this reason, we choose to investigate another family of VOF-type methods called THINC (Tangent of Hyperbola for Interface Capture). THINC schemes use a smoothed Heaviside function with a fixed thickness to approximate the interface (if desired, a sharp interface can then be obtained by taking the 0.5 level set). Besides the obvious advantage of their algebraic simplicity, THINC schemes also lead to less flotsam and jetsam, making them very successful in practice (see for example [Yokoi, 2013]).

The first introduction to THINC methods for interface capturing was given by [Xiao, 2005] and described the algebraic derivation of the THINC scheme. Since, several extensions have been developed that improve
its accuracy, for example the THINC/SW scheme [Xiao, 2011] with *slope weighing*, the THINC/WLIC scheme [Yokoi, 2007] with *weighted line interface calculation*, multidimensional reconstruction [Xie, 2014], extensions to unstructured meshes and others.

### 4.1.1 THINC Scheme

We will now proceed to define the THINC scheme as given in [Xiao, 2005] with some elements from the THINC/SW scheme from [Xiao, 2011]. Generally, the cell averaged values are given by:

\[
\alpha^m_i = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \alpha(t^m, x) \, dx,
\]

where \( \alpha \) denotes the volume fraction (using the notation from the previous chapter). In this case, we know that the volume fraction is a Heaviside function, so a valid approximation would be to use a hyperbolic tangent-based mollifier as follows:

\[
\alpha^m_i \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \phi^m_i(x) \, dx,
\]

where \( \phi^m_i \) is given by:

\[
\phi^m_i(x) = \frac{1}{2} \left( 1 + \frac{\gamma^m_i \tanh(\beta \left[ x - x_i \Delta x - \delta^m_i \right])}{\gamma^m_i + \beta} \right).
\]

Typically, in Finite Volume methods the approximations come into play when we define the various face fluxes. We will see that, in the THINC scheme, the approximation we have made above is the only one necessary, since the fluxes can be computed exactly afterwards. For simplicity, we perform a change of variables:

\[
\xi \rightarrow \frac{x - x_i}{\Delta x}
\]

and redefine the cell average as:

\[
\alpha^m_i = \int_{-0.5}^{0.5} \phi^m_i(\xi) \, d\xi,
\]

where the mollifier is now given by:

\[
\phi^m_i(\xi) = \frac{1}{2} \left( 1 + \frac{\gamma^m_i \tanh(\beta(\xi - \delta^m_i))}{\gamma^m_i + \beta} \right).
\]

The parameters in the mollifier have a special meaning that also defines how they can be computed:

- \( \beta \) is the non-dimensional *slope steepness*, which defines how sharp the interface is (larger \( \beta \) implies a sharper interface). However, it is not necessarily a very intuitive quantity to work with. For this
reason, we redefine it (see [Xiao, 2011]) using another quantity \( \eta \), which represents half the number of cells of size \( \Delta x \) over which the interface spans. Then, the steepness \( \beta \), as a function of the thickness \( \eta \), is given by:

\[
\beta = \frac{1}{\eta} \tanh^{-1}(1 - 2\epsilon),
\]

(4.3)

where \( \epsilon \) is a cutoff value for the hyperbolic tangent. Defining a cutoff value is necessary because the hyperbolic tangent only goes to \(-1\) or 1 at infinity, which would imply that none of the cells are completely full or empty. The enhancement given by the THINC/SW scheme consists in modifying \( \beta \) in each dimension as follows:

\[
\beta = \frac{|n_x|}{\eta} \tanh^{-1}(1 - 2\epsilon),
\]

where \( n = (n_x, n_y, n_z) \) is the normal in the current cell.

- \( \gamma \) is the interface direction given by:

\[
\gamma^m_i = \begin{cases} 
1, & \alpha^m_i - 1 < \alpha^m_i - 1 \\
-1, & \alpha^m_i - 1 > \alpha^m_i - 1.
\end{cases}
\]

(4.4)

- Finally, \( \delta \) is the jump location, i.e. it defines the location of the 0.5 level set of \( \phi^m_i \). The jump location is the only real unknown and can be computed by imposing volume conservation in the cell, i.e. we must find \( \delta \) such that the equality from (4.1) holds. The integration can be done exactly, since we can compute a primitive of \( \phi^m_i \):

\[
\Phi^m_i(\xi) = \frac{1}{2} \left( \xi + \frac{\gamma^m_i}{\beta} \log(\cosh(\beta(\delta^m_i - \xi))) \right).
\]

(4.5)

Therefore, the solution is given by:

\[
\delta^m_i = \frac{1}{2\beta} \log \left( \frac{\sinh \frac{\beta}{2} (1 - \omega^m_i)}{\sinh \frac{\beta}{2} (1 + \omega^m_i)} \right),
\]

(4.6)

where:

\[
\omega^m_i = \gamma^m_i (2\alpha^m_i - 1).
\]

The flux for the THINC scheme is given by simply integrating over an upstream segment around a given face:
• if \(c > 0\):
\[
f_{m,-}^{i-\frac{1}{2}} = \int_{0.5-\lambda}^{0.5} \phi_i^{m-1}(\xi) \, d\xi
\]
\[
= \frac{1}{2} \left[ \lambda + \frac{\gamma_i^{m-1}}{\beta} \log \left( \frac{\cosh(\beta(\delta_i^{m-1} - 0.5))}{\cosh(\beta(\delta_i^{m-1} - 0.5 + \lambda))} \right) \right].
\] (4.7)

• if \(c \leq 0\):
\[
f_{m,+}^{i-\frac{1}{2}} = -\int_{-0.5-\lambda}^{-0.5} \phi_i^{m}(\xi) \, d\xi
\]
\[
= \frac{1}{2} \left[ \lambda - \frac{\gamma_i^{m}}{\beta} \log \left( \frac{\cosh(\beta(\delta_i^{m} + 0.5 + \lambda))}{\cosh(\beta(\delta_i^{m} + 0.5))} \right) \right],
\] (4.8)

where
\[
\lambda = c \frac{\Delta t}{\Delta x}.
\]

For completeness, we denote the final THINC flux by (using (3.12)):
\[
f_{THINC}^{i-\frac{1}{2}}(\alpha_i^{m-1/2}, \gamma_i^{m-1/2}) = (1 - s_i^{m-1/2})f_{i-\frac{1}{2}}^{m-1/2} + s_i^{m-1/2}f_{i-\frac{1}{2}}^{m+1/2},
\]

and note that it is a function of only the upwind quantities:
\[
\begin{align*}
\alpha_i^{m-1/2} &= (1 - s_i^{m-1/2})\alpha_i^{m-1} + s_i^{m-1/2}\alpha_i^{m}, \\
\gamma_i^{m-1/2} &= (1 - s_i^{m-1/2})\gamma_i^{m-1} + s_i^{m-1/2}\gamma_i^{m}.
\end{align*}
\] (4.9)

Finally, the update formula for a THINC scheme defined as above is:
\[
\alpha_i^{m+1} = \alpha_i^m - (f_{i+1/2}^{THINC} - f_{i-1/2}^{THINC}).
\] (4.10)

However, there is an issue with the THINC scheme as we have defined it. When the volume fraction approaches a pure state, i.e. \(\alpha \to 0\) or \(\alpha \to 1\), the jump location \(\delta\) will go to infinity, since that is where the hyperbolic tangent achieves its limit. This can be seen in Figure 4.1 for various values of \(\beta\).

Given that we cannot compute \(\delta\) in pure regions, a complete definition of the THINC scheme must include another flux that is well defined in those regions:
\[
f_{UT}^{i-\frac{1}{2}}(\alpha_i^{m-1/2}, \gamma_i^{m-1/2}) = \begin{cases} f_{i-1/2}^{THINC}, & \alpha_i^{m-1/2} \in (\epsilon, 1 - \epsilon), \\
& \alpha_i^{m-1/2} \notin (\epsilon, 1 - \epsilon), \\
& \text{otherwise},
\end{cases}
\] (4.11)

where we have chosen the cutoff value \(\epsilon\) to be the same as the one from the definition of \(\beta\) in (4.3), but this

49
is not required by the scheme. As an example, we have chosen the upwind flux as the alternative flux. Even if it is very diffusive, it will only activate in regions where \( \alpha < \epsilon \) and \( \alpha > 1 - \epsilon \), for \( \epsilon \ll 1 \), so it has a very small impact on the overall result. Higher-order methods can be used if deemed necessary.

Having completely defined the THINC scheme, we can proceed to defining the linearized and adjoint equations. The generic formulation is the same as the one given in (3.9) and (3.10). As we have seen, the THINC flux is highly nonlinear compared to that of the upwind scheme (and even the flux limited scheme) and warrants careful investigation of the nature and behavior of the derivatives required by the linearized and adjoint schemes. A complete set of derivatives of the THINC fluxes is given in Appendix C, so we will proceed with analyzing the results. Note that, when computing the derivatives, we have assumed that \( \gamma \) is not a function of the volume fraction \( \alpha \), resulting in an incomplete differentiation of the flux.

In Figure 4.2 we can see how the THINC flux compares to the same results using the upwind flux. One important note is that, at a cursory glance, it seems that the THINC flux converges to the upwind flux for limit values of \( \alpha \) (see also Appendix D), thus justifying our previous choice to a certain degree. However, there is a large jump between the derivative of the THINC flux and the upwind flux in limit values of \( \alpha \). This implies that the derivatives of the combined flux (4.11) will have discontinuities at \( \epsilon \) and \( 1 - \epsilon \) that may cause issues. Note that the THINC flux is itself already not completely differentiable since \( \gamma \) is defined as a piecewise function. Furthermore, extensions such as the THINC/SW scheme add more non-differentiable terms.

To test the exact effect of this discontinuity we will reprise the test case from Chapter 3 with a boxcar
function (3.14). However, we cannot use the example as is, since the THINC scheme will smear any sharp discontinuities and will sharpen any smooth regions until they have a slope thickness corresponding to \( \eta \). To avoid any artificial smearing or sharpening, we define the following signed distance function to the interface represented by the boxcar initial condition:

\[
d(t, x) = \begin{cases} 
    x - (0.25 + ct), & x < 0.5 + ct, \\
    (0.75 + ct) - x, & x \geq 0.5 + ct,
\end{cases}
\]

The actual initial condition is then defined using a simplified version of the hyperbolic tangent mollifier (4.2), with \( \delta = 0 \) and \( \gamma = 1 \), so that it has the desired interface thickness:

\[
\alpha(0, x) = \frac{1}{2} \left( 1 + \tanh \left( \frac{\beta d(0, x)}{\Delta x} \right) \right).
\]

The rest of the problem parameters are the same: final time is \( T = 1 \), velocity is \( c = 1 \), CFL condition is 0.9 and boundary conditions are periodic. For the linearized scheme, we linearize around the approximated nonlinear solution and use the same initial condition. The results at time \( T = 1 \) using both the nonlinear and the linearized scheme can be seen in Figure 4.3.

There are multiple issues present in the linearized scheme as given in Figure 4.3. First, we can see there are two pillars forming at the interface, reminiscent of Dirac delta functions. They are completely expected since we are differentiating a smoothed Heaviside function, the results of which will be a smoothed delta function. The second issue is related the the dips that form on the interior around the \( \alpha = 1 \) phase. These
Figure 4.3: Solution of (3.14) at time \( t = 1 \) for the linearized and nonlinear THINC schemes with \( \eta = 12.5 \) (left) and \( \eta = 2.5 \) (right).

phenomena are entirely due to the fact that we are switching to the upwind scheme at a certain point. To see exactly what is happening, we take the case of \( \gamma = 1 \) (as in Figure 4.2) and look at a cell on the left interface, where the flux on the face to the left is still the THINC flux and on the right face side we have the upwind flux:

\[
- \left( \frac{\partial f_{UPWIND}^{i+1/2}}{\partial \tilde{\alpha}_m^i} \tilde{\alpha}_m^i - \frac{\partial f_{THINC}^{i-1/2}}{\partial \tilde{\alpha}_m^{i-1}} \tilde{\alpha}_m^{i-1} \right),
\]

where \( \tilde{\alpha}_m^i \) is the perturbed solution at time \( m \). Given the results from Figure 4.2 and knowing that \( \tilde{\alpha}_i \approx \tilde{\alpha}_m^i \approx 1 - \epsilon \), we can deduce that the quantity above must be negative. Furthermore, in regions where \( \alpha \) is approximately constant, we would expect the difference to be close to 0. Since this is not the case, this larger term then causes the apparition and growth of the dips we see. The case of \( \gamma = -1 \) is completely symmetric and causes the dip on the other side of the full phase.

Next we will study the adjoint equation. For the forward solve, we will use the same test case as before and for the adjoint we will take the cost function defined in (3.1) using:

\[
G(\alpha(T)) = \frac{1}{2} \alpha(T)^2,
\]

so that the starting condition for the adjoint is the forward solution at the final time \( T = 1 \). Note that this is the exact same problem we have studied in Chapter 3 and has \( \alpha^0 = 0 \) as a trivial optimal solution. The optimal solution can be obtained in one step if the adjoint equation is solved exactly. The results of the adjoint scheme can be seen in Figure 4.4.

The adjoint scheme seems to be behaving a lot better than the linearized scheme we have seen before, in the sense that it is closer to the exact solution. In the case of a wider interface, the results compare...
Figure 4.4: Solution of (3.14) at time $t = 0$ for the adjoint and nonlinear THINC schemes with $\eta = 12.5$ (left) and $\eta = 2.5$ (right).

favorably against the exact solution at time $t = 0$. For steeper transitions, we see plateaus appearing at the interface at 0.5. Unfortunately, unlike for the linearized scheme, there is very little intuition to be had for the adjoint, so we cannot say what the reason behind this behavior is exactly.

### 4.1.2 Blended THINC Scheme

We have seen that the THINC scheme, as defined in [Xiao, 2005], poses some problems in the linearized regime as the volume fraction approaches one of the pure states. We will attempt to rectify the artificial dips present in Figure (4.3) by providing a smooth transition from the THINC flux to the alternative upwind flux. Without making any other modification to either scheme, we define a new combined flux as:

$$ f_{\text{BT}}^{i-1/2} = f_{\text{UPWIND}}^{i-1/2} + \psi(\alpha_i) \left( f_{\text{THINC}}^{i-1/2} - f_{\text{UPWIND}}^{i-1/2} \right), \quad (4.13) $$

where $\psi$ is a given blending function that provides a gradual transition between two fluxes and also ensures that the THINC scheme is not used when $\alpha < \epsilon$ or $\alpha > 1 - \epsilon$. Possible candidates for such a function are a
differentiable sine-based blending function:

\[
\psi(\alpha) = \begin{cases} 
0, & \alpha < \epsilon, \\
\frac{1}{2} \left( 1 + \sin \left( \frac{\pi}{w}(\alpha - \epsilon - \frac{w}{2}) \right) \right), & \alpha \in [\epsilon, \epsilon + w), \\
1, & \alpha \in (\epsilon + w, 1 - (\epsilon + w)), \\
\frac{1}{2} \left( 1 + \sin \left( \frac{\pi}{w}(1 - \alpha - \epsilon - \frac{w}{2}) \right) \right), & \alpha \in [1 - (\epsilon + w), 1 - \epsilon], \\
0, & \alpha > 1 - \epsilon.
\end{cases}
\]

or a piecewise linear function such as:

\[
\psi(\alpha) = \begin{cases} 
0, & \alpha < \epsilon, \\
\frac{1}{w} \alpha - \epsilon, & \alpha \in [\epsilon, \epsilon + w), \\
1, & \alpha \in [\epsilon + w, 1 - (\epsilon + w)), \\
\frac{1}{\epsilon} (1 - \alpha) - \frac{\epsilon}{\epsilon}, & \alpha \in [1 - (\epsilon + w), 1 - \epsilon], \\
0, & \alpha > 1 - \epsilon.
\end{cases}
\]

where \( w \) is the width of the transition area. The two blending functions can be seen in Figure 4.5.

Figure 4.5: A sine-based (left) and a piecewise linear (right) blending function for a width of \( w = 0.2 \) and a cutoff value \( \epsilon = 0.05 \).

The resulting flux and flux derivative can be seen in Figure 4.6 for \( c > 0 \). As expected, both the flux and the flux derivative smoothly transition from the THINC values to the upwind values.

Using the same initial condition and parameters as in the previous case, we will investigate the linearized
behavior of this new extension to the THINC scheme. Note that the linearization of this scheme is incomplete as well, in the sense that we have not differentiated the blending function $\psi$ in the following results. The linearized scheme at time $T = 1$ can be seen in Figure 4.7.

The results from Figure 4.7 indicate that both the nonlinear and the linearized schemes are significantly more diffusive. Furthermore, the smooth transition we have introduced does not seem to also remove the non-physical dips due to the transition to the upwind scheme. Overall, a blending of this type can be effective for thick interfaces and large transition areas in the blending function, but, ultimately, it only delays the apparition of the same effects we have seen in the original THINC scheme.

For completeness, we also give the analogue in terms of the adjoint equations to the results in the previous section. The results can be seen in Figure 4.8. In the same manner, we see that the apparition of the plateaus...
we have seen previously in Figure 4.4 is considerably delayed, but their other characteristics are not changed at all (can be seen even better for larger time scales).

![Figure 4.8: The adjoint scheme at time $t = 0$ for a slope steepness $\eta = 12.5$ (left) and $\eta = 2.5$ (right).](image)

### 4.1.3 Extended THINC Scheme

We have seen that having a smooth flux does not have a large impact on the overall behaviour of the linearized and adjoint THINC schemes. The main culprit remains the change to the upwind scheme in pure regions, which causes the dips we have seen repeatedly in Figure 4.3 and Figure 4.7. The reason we have initially used the upwind scheme was that the jump location $\delta$ is not well-defined in pure regions. On the other hand, Figure 4.2 clearly shows that even for large $\delta$, the flux stays bounded. This leads us to believe that a first order approximation of the THINC flux can be used in pure regions, which could solve some of the artifacts that appear due to the upwind scheme. The derivation of the first-order approximation is given in Appendix D and we will only give the main results here.

A first-order approximation as $\alpha \to 0$ is given by:

- if $c > 0$:

$$f^\text{ZERO}.^-_{i-1/2} = \alpha_{i-1}^m \exp(\gamma_{i-1} \beta (1 - \lambda)) \frac{\sinh(\beta \lambda)}{\sinh(\beta)}$$

- if $c \leq 0$:

$$f^\text{ZERO}.^+_{i-1/2} = \alpha_i^m \exp(-\gamma_i \beta (1 + \lambda)) \frac{\sinh(\beta \lambda)}{\sinh(\beta)}$$

A first-first order approximation as $\alpha \to 1$ is given by:
• if \( c > 0 \):
\[
f_{\text{ONE},-}^{i-1/2} = \lambda + (\alpha_{i-1}^m - 1) \exp(-\gamma_{i-1} \beta (1 - \lambda)) \frac{\sinh(\beta \lambda)}{\sinh(\beta)}. \]

• if \( c \leq 0 \):
\[
f_{\text{ONE},+}^{i-1/2} = \lambda + (\alpha_i^m - 1) \exp(\gamma_i \beta (1 + \lambda)) \frac{\sinh(\beta \lambda)}{\sinh(\beta)}. \]

The full extended THINC flux is then given by:
\[
f_{\text{ET}}^{i-1/2}(\alpha_{i-1/2}^m, \gamma_{i-1/2}^m) = \begin{cases} 
  f_{\text{THINC}}^{i-1/2}, & \alpha_{i-1/2}^m \in (\epsilon, 1 - \epsilon), \\
  (1 - s_{i-1/2}^m) f_{\text{ZERO},-}^{i-1/2} + s_{i-1/2}^m f_{\text{ZERO},+}^{i-1/2}, & \alpha_{i-1/2}^m \leq \epsilon, \\
  (1 - s_{i-1/2}^m) f_{\text{ONE},-}^{i-1/2} + s_{i-1/2}^m f_{\text{ONE},+}^{i-1/2}, & \alpha_{i-1/2}^m \geq 1 - \epsilon.
\end{cases}
\]

The linearized scheme is given by (for \( c > 0 \)):
\[
\tilde{\alpha}_{i+1}^m = \tilde{\alpha}_i^m - \left[ \left( \frac{\partial f_{\text{ET}}^{i+1/2}}{\partial \alpha_i^m} \right) \tilde{\alpha}_i^m \right] - \left( \frac{\partial f_{\text{ET}}^{i+1/2}}{\partial \alpha_{i-1}^m} \right) \tilde{\alpha}_{i-1}^m, \tag{4.17}
\]

where \( \tilde{\alpha} \) is the perturbation. There are several issues that can arise using this formulation that we did not have before. They appear in the limits of the pure regions as \( \alpha \rightarrow 0 \) or \( \alpha \rightarrow 1 \) and reveal some new issues with the THINC scheme. They are:

• Firstly, given the definition of \( \gamma \) from (4.4), we can see that \( \gamma \) is in fact not well defined in pure regions where \( \alpha_{i-1}^m = \alpha_{i+1}^m \). Especially numerically, this can cause \( \gamma \) to oscillate between faces. While this is not a problem for the nonlinear flux since its limits do not depend on \( \gamma \), it is an issue for the flux derivative that appears in the linearized scheme and whose limits do depend on \( \gamma \) (see Appendix D).

This was not an issue before because, of course, the upwind flux does not depend on \( \gamma \) at all and the THINC scheme was never used close enough to the pure states to reveal numerical accuracy problems.

• Secondly, there is also the possibility that we have a different \( \gamma \) on the left and right face. The value of the THINC flux derivative for \( \gamma = \pm 1 \) are similar, but mirrored around a vertical line. This implies that in the case where \( \gamma \) is different on the two faces of a given cell, we will have a large discrepancy between the left and the right term in the update formula (4.17), which will cause spurious oscillations in the linearized scheme.

To mitigate these issues, we will first define \( \gamma \) per-cell instead of per face, so that the update formula (4.17)
where each flux has been changed to use the same slope direction. Furthermore, we must redefine \( \gamma \) to treat the pure states separately and avoid any numerical accuracy issues:

\[
\gamma_i^m = \begin{cases} 
-\text{sign}(c), & \alpha_i^m < \epsilon, \\
\text{sign}(c), & \alpha_i^m > 1 - \epsilon, \\
\text{sign}(\alpha_{i+1}^m - \alpha_{i-1}^m), & \text{otherwise},
\end{cases}
\]

(4.18)

where the value of \( \gamma \) for \( \alpha < \epsilon \) and \( \alpha > 1 - \epsilon \) was chosen empirically as the one that gave good results in practice.

Using the same scenario as before, Figure 4.9 gives us the results of the linearized scheme at final time \( T = 1 \). The first notable feature of this new scheme is that we now only have the smooth Dirac delta functions at the interface, as expected initially. However, in comparison to previous tests, the height of the delta functions will be a lot larger because we can no longer rely on the upwind scheme to dampen them.

Figure 4.9: The linearized scheme at time \( T = 1 \) for a slope steepness \( \eta = 12.5 \) (left) and \( \eta = 2.5 \) (right).

Before proceeding to the adjoint scheme, we note that some of the same problems with the definition of the interface direction \( \gamma \) appear. In this case, we must choose \( \gamma \) per-face to avoid oscillations and define it as:

\[
\gamma_i^m = \begin{cases} 
\text{sign}(\alpha_{i+1}^m - \alpha_{i-1}^m), & \alpha_i \in (\epsilon, 1 - \epsilon), \\
-\text{sign}(\alpha_{i+1}^m - \alpha_{i-1}^m), & \text{otherwise},
\end{cases}
\]

(4.19)

where the value in the limit of a pure state was chosen empirically as the one providing the best results.
Note that the nonlinear scheme must also change its definition of $\gamma$ based on whether the desired end result is simulating the linearized scheme or the adjoint scheme.

The results of the adjoint scheme at time $t = 0$ can be seen in Figure 4.10. In the case of a thick interface, the results are comparable to the ones obtained using the blended scheme and in the case of a sharper interface, they are comparable to the results obtained with the original THINC scheme, but with plateaus of a smaller width.

![Figure 4.10: The adjoint scheme at time $t = 0$ for a slope steepness $\eta = 12.5$ (left) and $\eta = 2.5$ (right).](image)

Overall, the THINC scheme using a first-order approximation should be the preferred one for a number of reasons. Firstly, it does not require the use of another flux and has consistent derivatives that give the correct linearized results. Secondly, using the redefined interface direction $\gamma$, the results for the adjoint are very similar to the ones obtained by the blended scheme without any of the additional diffusion.

### 4.2 COMIC: Compact Mollifier for Interface Capturing

In the previous section we have seen a complete description of the THINC scheme and its application to both the linearized and adjoint equations, to various degree of success. The main culprit in the problems concerning the THINC scheme was the fact that the hyperbolic tangent-based mollifier never gave us a definite pure state of either 0 or 1, so we had to work around it with various methods. In this chapter we propose a new class of numerical schemes, similar in spirit to the THINC scheme, which use a compact mollifier, in the sense that the pure states are achieved in a finite interval.

For this purpose we require a set of compact approximations to the Heaviside function. A very important source of inspiration for such functions is the work done on immersed boundary methods, e.g. in [Roma, 1999]. Immersed boundary methods usually require a smoothed Dirac delta function to define a stencil around the
boundary. These functions are usually compact and have many important properties (given in [Roma, 1999] and earlier works). Any of them can be converted into a smooth approximation to a Heaviside function by finding a primitive.

The mollifier we will be looking at is a sine-based mollifier (based on a cosine-based smooth delta function from [Roma, 1999]) like the following:

\[
\phi_m^i(\xi) = \begin{cases} 
  a, & \xi < \delta^m_i - \eta, \\
  a + \frac{b - a}{2} \left[ 1 + \sin \left( \frac{\pi}{2\eta} (\xi - \delta^m_i) \right) \right], & \xi \in [\delta^m_i - \eta, \delta^m_i + \eta], \\
  b, & \xi > \delta^m_i + \eta.
\end{cases}
\]

where \(\eta\) describes the slope thickness and \(\delta\) is the jump location, which we will try to determine. The other two variables, \(a\) and \(b\) describe the minimum and maximum value that our function can achieve and, in our case, are given by:

\[
\begin{align*}
  a &= 1 - \max(\gamma^m_i, 0), \\
  b &= \max(\gamma^m_i, 0).
\end{align*}
\]

### 4.2.1 Jump Location

All the variables in the COMIC scheme have the exact same significance as in the THINC scheme. The problematic part of the scheme is finding \(\delta\), for which we will need to impose volume conservation as before:

\[
\Psi^m_i(\delta^m_i) = \int_{-0.5}^{0.5} \phi^m_i(\xi) d\xi = \alpha^m_i.
\]

The primitive of the given mollifier is:

\[
\Psi^m_i(\delta^m_i) = \begin{cases} 
  a, & \delta^m_i - \eta > 0.5, \\
  b, & \delta^m_i + \eta < -0.5, \\
  \Phi^m_i(\delta^m_i), & \text{otherwise}
\end{cases}
\]

where:

\[
\Phi^m_i(\delta^m_i) = a \max (\delta^m_i - \eta + 0.5, 0) + b \max (0.5 - \delta^m_i - \eta, 0) + 0.5 \left[ \xi - \gamma^m_i \frac{2\eta}{\pi} \cos \left( \frac{\pi}{2\eta} (\xi - \delta^m_i) \right) \right]_{\min(\delta^m_i + \eta, 0.5)}^{\max(\delta^m_i - \eta, -0.5)}.
\]

Clearly, we cannot analytically find the solution to \(\Psi^m_i(\delta^m_i) = \alpha^m_i\). However, if we assume that the interval
to which $\delta$ belongs to is known, we can simplify the above equations into the following cases (forgoing the use of indices):

0. $\delta - \eta > 0.5$ or $\delta + \eta < -0.5$. This is the case where the entire cell is in one of the pure states (see Figure 4.12). Furthermore, any value of $\delta$ that satisfies the above conditions for a given $\eta$ will be valid, even infinite ones. However, we will choose the smallest possible one (in magnitude). This gives:

$$\delta = \begin{cases} 
\gamma(\eta + 0.5 + \epsilon), & \alpha < \epsilon, \\
-\gamma(\eta + 0.5 + \epsilon), & \alpha > 1 - \epsilon,
\end{cases}$$

where $\epsilon \ll 1$ can be used to avoid numerical accuracy issues and should be around machine precision.

1. $\delta - \eta > -0.5$ and $\delta + \eta < 0.5$. This is the case where the entire interface is contained in one cell (see Figure 4.13) and can only occur if $\eta < 0.5$. In this case, we have that:

$$\Psi(\delta) = (a - b)\delta + (a + b)(0.5 - \eta) + \eta = (a - b)\delta + 0.5 = \alpha,$$

since $a + b = 1$. We then get:

$$\delta = \frac{1}{a - b}(\alpha - 0.5).$$

2. $\delta - \eta < -0.5$ and $\delta + \eta > 0.5$. This is the case where the mollifier never goes to 0 or 1 in the current
cell (see Figure 4.13). We have that:

\[
\Psi(\delta) = \frac{1}{2} \left[ \frac{1}{2} - \gamma \frac{2\eta}{\pi} \cos \left( \frac{\pi}{2\eta}(\delta - 0.5) \right) \right] - \frac{1}{2} \left[ \frac{1}{2} - \gamma \frac{2\eta}{\pi} \cos \left( \frac{\pi}{2\eta}(\delta + 0.5) \right) \right] \\
= \frac{1}{2} \left[ 1 - \gamma \frac{2\eta}{\pi} \left( \cos \left( \frac{\pi}{2\eta}(\delta - 0.5) \right) - \cos \left( \frac{\pi}{2\eta}(\delta + 0.5) \right) \right) \right] \\
= \frac{1}{2} \left[ 1 - \gamma \frac{4\eta}{\pi} \sin \left( \frac{\pi}{4\eta} \right) \sin \left( \frac{\pi}{2\eta} \delta \right) \right] = \alpha.
\]

Which we can invert to get:

\[
\delta = -\frac{2\eta}{\pi} \arcsin \left( \frac{\pi}{4\eta} \left( \sin \left( \frac{\pi}{4\eta} \right) \right)^{-1} (2\alpha - 1) \right).
\]

Figure 4.13: Mollifier (4.20) with \( \delta - \eta > -0.5 \) and \( \delta + \eta < 0.5 \) (left) and \( \delta - \eta < -0.5 \) and \( \delta + \eta > 0.5 \) (right).
3. \(\delta - \eta < -0.5\) and \(\delta + \eta < 0.5\). This is the case where the pure state on the left (whether this is the empty or full state depends on \(\gamma\)) is in the current cell (see Figure 4.14). In this case we have that:

\[
\Psi(\delta) = \left(b - \frac{1}{2}\right)\left(\frac{1}{2} - \delta - \eta\right) + \frac{1}{2}\left[1 + \gamma \frac{2\eta}{\pi} \cos\left(\frac{\pi}{2\eta}(\delta + 0.5)\right)\right] = \alpha
\]

4. \(\delta - \eta > -0.5\) and \(\delta + \eta > 0.5\). This is the case where the pure state on the right is in the current cell (see Figure 4.14). In this case we have that:

\[
\Psi(\delta) = \left(a - \frac{1}{2}\right)\left(\delta - \eta + \frac{1}{2}\right) + \frac{1}{2}\left[1 - \gamma \frac{2\eta}{\pi} \cos\left(\frac{\pi}{2\eta}(\delta - 0.5)\right)\right] = \alpha
\]

Figure 4.14: Mollifier (4.20) with \(\delta - \eta < -0.5\) and \(\delta + \eta < 0.5\) (left) and \(\delta - \eta > -0.5\) and \(\delta + \eta > 0.5\) (right).

Given the results so far, we have two main ways to compute \(\delta\):

- We can use a bracketed root finding algorithm to find \(\delta\) satisfying (4.21).
- Find the interval to which \(\delta\) belongs to, based on \(\alpha\), and solve according to the cases outlined above.

We can see here that cases 0 – 2 can be solved analytically and cases 3 – 4 still require a root finding algorithm.

We would want to use the second approach because it is more computationally efficient to only use a root finding algorithm in a small subset of the full domain. We can find the limits of the intervals above based on \(\alpha\) if we solve:

\[
\delta(\alpha_1) - \eta = -0.5 \quad \text{and} \quad \delta(\alpha_2) + \eta = 0.5,
\]

where we consider \(\delta\) as a function of \(\alpha\). By the implicit function theorem, we can see that \(\alpha_i\) is a unique
function of $\eta$, so it would only need to be computed once at the beginning of the simulation. If $\eta$ would itself depend on $\alpha$, like in the case of an extension similar to the THINC/SW scheme, more advanced interpolation methods could be used to approximate $\alpha_i(\eta)$.

Figure 4.15: The two critical values of alpha from (4.23) for $\gamma = 1$.

Solving the equations from (4.23) can be seen in Figure 4.15. The middle range of $\alpha \in [\alpha_1, \alpha_2]$ represents case 2 above, while $\alpha < \alpha_1$ represents case 4 and $\alpha > \alpha_2$ represents case 3, for a value of $\gamma = 1$. The advantage of separating the computation of $\delta$ per cases is obvious in this case: as $\eta$ increases, the intervals on which we would need an expensive root finding algorithm decrease significantly. With some loss in accuracy, we could even do a Taylor expansion of the cosine in case 3 and case 4 for $\delta$ and solve them analytically as well.

Having defined multiple ways in which we can obtain the jump location $\delta$, all of the parameters of the COMIC scheme are now known. Finally, the fluxes are given by:

- if $c > 0$:

  \[
  f_{i-\frac{1}{2}}^{m,-} = \int_{0.5-\lambda}^{0.5} \phi_{i-1}^m(\xi) \, d\xi = \begin{cases} 
  a\lambda, & \delta_{i-1}^m - \eta > 0.5, \\
  b\lambda, & \delta_{i-1}^m + \eta < 0.5 - \lambda, \\
  \Phi_{i-1}(\delta_{i-1}^m), & \text{otherwise} 
  \end{cases}
  \]
where
\[ \Phi_{i-1}(\delta_{i-1}^{m}) = a \max(\delta_{i-1}^{m} - \eta - 0.5 + \lambda, 0) + b \max(-0.5 - \delta_{i-1}^{m} - \eta, 0) + \]
\[ \frac{1}{2} \left[ \xi - \gamma_{i-1}^{m} \frac{2\eta}{\pi} \cos \left( \frac{\pi}{2\eta} (\xi - \delta_{i-1}^{m}) \right) \right] \left| \min(\delta_{i-1}^{m} + \eta, -0.5) \right|_{\max(\delta_{i-1}^{m} - \eta, 0.5 - \lambda)} . \]

\[ f_{i-1/2}^{m,+} = \int_{i-0.5}^{0-0.5} \psi_{i}^{m}(\xi) d\xi \]
\[ = \begin{cases} \alpha \lambda, & \delta_{i}^{m} - \eta > -0.5 - \lambda, \\ b \lambda, & \delta_{i}^{m} + \eta < -0.5, \\ \Phi_{i}^{m,+}(\delta_{i}^{m}), & \text{otherwise} \end{cases} \]

\[ \Phi_{i}^{m,+}(\delta_{i}^{m}) = a \max(\delta_{i}^{m} - \eta + 0.5, 0) + b \max(-0.5 - \lambda - \delta_{i}^{m} - \eta, 0) + \]
\[ \frac{1}{2} \left[ \xi - \gamma_{i}^{m} \frac{2\eta}{\pi} \cos \left( \frac{\pi}{2\eta} (\xi - \delta_{i}^{m}) \right) \right] \left| \min(\delta^{m} + \eta, -0.5 - \lambda) \right|_{\max(\delta^{m} - \eta, 0.5)} . \]

The complete flux is then:
\[ f_{i-1/2}^{\text{COMIC}}(\alpha_{i-1/2}^{m}, \gamma_{i-1/2}^{m}) = \begin{cases} f_{i-1/2}^{m,-}, & c > 0, \\ f_{i-1/2}^{m,+}, & c \leq 0, \end{cases} \] (4.24)

using the definitions from (4.9). It may be advantageous to also split the flux computation into several cases, as we have done for the jump location \( \delta \). This has been done in Appendix E for all the above scenarios. We can also see in Figure 4.16 the flux and the flux derivative of the COMIC scheme for various values of \( \eta \).

Figure 4.16: The flux (left) and flux derivative (right) for the COMIC scheme with \( c > 0 \) and \( \gamma = 1 \).
The flux obtained by the COMIC scheme in Figure 4.16 is very similar to the flux we have obtained for the THINC scheme in Figure 4.2, for equivalent values of the slope thickness $\beta$ (that can be computed from $\eta$). The main difference comes in the way of the flux derivative: the compact support of the COMIC scheme allows the derivative to actually go completely to 0 or 1 in limit cases for all values of $\eta$, while the THINC scheme never achieved this, even for larger $\beta$. We will see in the next sections how this change affects the behavior of the linearized schemes.

### 4.2.2 Linearized Equations

Now that we have computed all the necessary parts of the COMIC scheme, we can look at the linearized scheme. Using the same parameters and initial condition as in the case of the various THINC schemes, we can see the results of the new linearized COMIC scheme in Figure 4.17. Note that, like in the case of the THINC scheme with a first order approximation, the use of the modified $\gamma$ from (4.18) is crucial.

![Figure 4.17: The linearized COMIC scheme for $\eta = 12.5$ (left) and $\eta = 2.5$ (right).](image)

The linearized COMIC scheme, as seen in Figure 4.17 seems to behave significantly worse than the THINC scheme with a first-order approximation in Figure 4.9. The main culprit for this behavior is that the COMIC derivatives achieve larger values as $\alpha \to 1$, so the resulting approximations of the Dirac delta functions have significantly higher peaks. For this reasons, even the case of $\eta = 12.5$, where the interface is very smooth, we still see significant growth.

It may be interesting to study the growth rate of the two different schemes. For this reason, we will plot the quantity:

$$\frac{1}{\eta} \max_i \alpha_i^m,$$

for all time steps. The results can be seen in Figure 4.18. Indeed, the COMIC scheme grows faster than the
extended THINC scheme by an order of magnitude for the same values of $\eta$.

![Graph showing growth over time of the COMIC scheme (left) and the extended THINC scheme (right).]

**Figure 4.18:** Growth over time of the COMIC scheme (left) and the extended THINC scheme (right).

### 4.2.3 Adjoint Equations

![Graph showing the adjoint COMIC scheme for $\eta = 12.5$ (left) and $\eta = 2.5$ (right).]

**Figure 4.19:** The adjoint COMIC scheme for $\eta = 12.5$ (left) and $\eta = 2.5$ (right).

We will now look at the adjoint scheme in the same scenario. The results can be seen in Figure 4.19. Perhaps surprisingly, the adjoint scheme seems to perform quite well compared to the other THINC schemes. The plateaus that have always been present in the THINC schemes (see Figure 4.10) are also present here, but they are smaller and closer to the interface.

An important takeaway from the results in this section is that the adjoint COMIC scheme seems to converge perfectly well in all pure regions. This can be seen clearly on the sharp interface in Figure 4.10. In the case of the smoother interface, the pure regions are already converged, but we need a longer time frame...
to see the complete plateaus forming over the whole width of the interface.

Figure 4.20: The COMIC scheme (left) and the extended THINC scheme (right) for different interface thicknesses.

We have performed this test for various interface thicknesses in Figure 4.20. We can see that both the extended THINC scheme and the COMIC scheme do not converge to the correct interface, for any value of the thickness $\eta$. One important difference between the two schemes is that the pure full state is maintained at its original width for the COMIC scheme, while in the extended THINC it becomes smaller for all $\eta$.

4.3 Comparison: Optimization

We have seen this far that the linearized and adjoint equations seem to converge everywhere except at the interface. The issues that arise at the interface are different between the schemes, but all of them exhibit some sort of a smoothed Dirac delta function in the linearized regime and some plateaus in the adjoint. The size and severity of these issues generally depends on the thickness of the interface. In the case of the linearized scheme, a thinner interface implies larger and faster growth in the size of the smooth delta functions. On the other hand, in the adjoint scheme, a thinner interface seems to imply a thinner plateau and, thus, a closer qualitative resemblance to the exact solution.

In this section we will look at a slightly more complicated cost functional and see how the following schemes perform in the optimization of the initial condition: the upwind scheme (3.26), the extended THINC scheme (4.16) and the COMIC scheme with the sine-based mollifier (4.24). The new cost functional we propose uses the following function:

$$G(\alpha(T)) = \frac{1}{2}(\alpha(T) - \alpha_d)^2,$$

68
where $\alpha_d$ represents a desired final state and is given by mollifying the following signed distance function:

$$\phi_d = \begin{cases} 
  x - 0.4, & x < 0.65, \\
  0.9 - x, & x > 0.65, 
\end{cases}$$

like in (4.12). The rest of the parameters of the numerical scheme will stay the same: the domain $\Omega = [0, 1]$, the final time $T = 1$, the CFL condition of 0.9. For the THINC and COMIC scheme we will use a interface half-thickness of $\eta = 12.5$ and $\epsilon = 0.01$ as a cutoff value.

The optimization algorithm we have used is a simple projected gradient descent with a fixed step size of $\rho = 0.5$. The stopping criteria used for the iteration are: a gradient norm tolerance $\epsilon_{\text{grad}} = 10^{-5}$, a cost functional tolerance $\epsilon_J = 10^{-5}$ and a cost functional difference between two iteration $\epsilon_{\text{rel}} = 10^{-5}$. We have also projected the initial condition to the $[0, 1]$ interval after each update as follows:

$$g^{(k+1)} = \min(1, \max(0, g^{(k)} - \rho \nabla g J^h)).$$

The initial condition for the optimization algorithm will $g(x) = \alpha(0, x)$ from (4.12). As such, the goal of the optimization is to move the boxcar initial condition from being centered on $x = 0.5$ to being centered at 0.65.

The results of the optimization using the upwind scheme can be seen in Figure 4.21. We can see that the interface of the approximated solution is obviously steeper than that of the desired solution. This is likely due to the excessive diffusion of the upwind scheme that flattened any small features appearing in the final state of the adjoint in the process of advecting them backwards.
In Figure 4.22, we can see the results for the extended THINC scheme. We can see that in this case the interface is slightly closer to the desired interface thickness. However, unlike in the case of the upwind scheme, the steepest descent algorithm we have used did not converge in the maximum of 30 iterations we have set for it. Analyzing the evolution of the optimal solution, we have seen that there are oscillations happening close to the optimal value. This is most likely do to the fact that the adjoint does not converge at the interface, so the optimization algorithm also had issues in this area. We can indeed see this in the norm of the control in Figure 4.23.

Finally, in Figure 4.24 we can see the results for the COMIC scheme. The results here are very similar to what we have seen with the THINC scheme. This is largely expected since the main difference between the
two seems to be the size of the plateaus in the adjoint. A notable difference is that in the case of the COMIC scheme the control does not zigzag around the minimum value. However, the optimization algorithm still does not seem to converge to exact solution around the interface.

![Figure 4.24: COMIC Scheme: Optimal solution (left) and cost functional values (right).](image)

### 4.4 Conclusions

This concludes our analysis of the THINC family of schemes. We have looked at the classic THINC scheme [Xiao, 2005], when used in a linearized regime by the linearized and adjoint equations. As originally defined, the THINC scheme is not suited for accurately representing the linearized and adjoint equations. We have determined multiple reasons for this behavior: the switch to a different scheme in pure regions, a badly defined interface direction in pure regions, etc. The main improvement that we present here to the THINC schemes comes in the form of extending it using a first-order approximation in the limit of pure states (4.16). This new extension together with the definition of the slope direction $\gamma$ from (4.19) provide adequate behavior for the linearized and adjoint equations, while not diminishing the qualities of the original nonlinear scheme.

As an alternative to the THINC schemes, we present a new family of schemes based on compact mollifiers that we call COMIC. The compact mollifiers allow us to reach pure states in a finite interval and, thus, remove the need for special treatment in these regions for the numerical flux (the need to redefine $\gamma$ as in (4.19) still remains). This new scheme is slightly more complex than the previous THINC scheme because we cannot analytically determine the jump location $\delta$ in all cases. To alleviate some of computational overhead that comes with using a root finding algorithm for finding the jump location, we propose a breakdown of the possible solutions into 5 regions, 3 of which can be solved analytically. This greatly improves the scheme.
and makes it competitive compared to the THINC scheme in terms of speed, although it has increased complexity.

Overall, we can say that the newly described COMIC scheme performs the best for its intended use in adjoint optimization. In the simple tests we have performed, the nonlinear version of the scheme performs similar to the other members of the THINC family of schemes by significantly reducing diffusion and keeping a constant-width interface. On the other hand, the resulting adjoint is closer to the exact solutions and doesn’t have any artifacts coming from improper use of other schemes (like the original THINC scheme).
Chapter 5

Motion Planning

We have seen in the previous chapters that convergence of linearized and adjoint equations for PDEs with discontinuities is possible. Specifically, we have studied the linear advection equation and the nonlinear Burgers’ equation and showed convergence, to some extent, of the discrete linearized and adjoint equation. Together, these two equations exhibit most of the problems that we can expect to encounter in multidimensional nonlinear conservation laws. The case of rarefaction waves was not studied here, but can be found in [Ulbrich, 2001].

Furthermore, we have studied the THINC family of schemes, which is heavily used in modeling interfaces in two-phase flows. We have seen that, with slight modifications, the classic 1D THINC scheme gives satisfactory results for both the linearized and adjoint equations. We have introduced several extensions and the new COMIC numerical scheme that improve on the THINC scheme in the context of adjoint optimization by reducing the numerical artifacts and providing better convergence.

Given these promising results, in this chapter we will investigate a slightly more complex scenario. The end goal of this chapter is to give us insight into how the optimization of two-phase flows, using the well-known one-fluid model (see [Prosperetti, 2009]), will behave and what results we can expect when modeling sharp interfaces. All the necessary notions related to discontinuities and anti-diffusive schemes that appear in such a problem have already been investigated in the previous chapters, so we are in an optimistic position for further research.

The problem we want to investigate involves matching the movement of the interface between two fluids to a desired motion. In the context of two-phase flows, we would want to find a velocity field that gives us this correct interface movement. However, this requires solving the complete Navier-Stokes equations and the complexity which comes with them. For this reason, we will look at a simplified problem, where the velocity field is given by a velocity potential and we will try to find the boundary conditions on this velocity potential. The choice of only controlling the boundary conditions on the velocity potential greatly reduces the dimensions of the problem, while keeping to the main goal of studying the interface representation. This choice also complicates, to some extent, the formal description of the problem and any proofs of existence.
and uniqueness will generally be hard to obtain.

To our knowledge, there are very few documented studies that attempt to deal with similar issues in multiphase flows. The main inspiration for the problem we are going to investigate is [Bernauer, 2011], where the motion of an interface in the two-phase Stefan problem was optimized using level set methods. In [Springer, 2015], we have a more complex scenario where the Navier-Stokes equations (in RANS form) are used to model the two-phase flow in an attempt to determine the optimal position of a body and minimize the forces acting on it. There are several other papers dealing with multiphase flows in porous media [Jansen, 2011], but they are not related to the current work.

5.1 Problem Description

The movement of the interface is given by the following advection equation:

\[
\begin{aligned}
\alpha_t + \mathbf{u} \cdot \nabla \alpha &= 0, & (t, \mathbf{x}) &\in [0, T] \times \Omega, \\
\alpha(0, \mathbf{x}) &= \alpha_0(\mathbf{x}), & \mathbf{x} &\in \Omega, \\
\alpha(t, \mathbf{x}) &= \alpha_D(t, \mathbf{x}), & (t, \mathbf{x}) &\in [0, T] \times \Gamma,
\end{aligned}
\] (5.1)

where \( \mathbf{u} \) is the velocity field, \( \alpha_0 \) is the initial condition and \( \alpha_D \) denotes the Dirichlet boundary conditions prescribed on the inflowing boundary \( \Gamma = \{ \mathbf{x} \in \partial \Omega \mid \mathbf{u} \cdot \mathbf{n} \leq 0 \} \).

To stay as close as possible to the case of an incompressible Navier-Stokes model, we assume that our velocity is incompressible as well, i.e. \( \nabla \cdot \mathbf{u} = 0 \). Furthermore, we will assume that it is irrotational, i.e. \( \nabla \times \mathbf{u} = 0 \). Together, these requirements allow us to define the velocity field as \( \mathbf{u} = \nabla \varphi \), where \( \varphi \) is the velocity potential that must satisfy the following Laplace equation:

\[
\begin{aligned}
\nabla \cdot \nabla \varphi &= 0, & \mathbf{x} &\in \Omega, \\
\varphi &= g(\mathbf{x}), & \mathbf{x} &\in \Gamma_D, \\
\nabla \varphi \cdot \mathbf{n} &= h(\mathbf{x}), & \mathbf{x} &\in \Gamma_N,
\end{aligned}
\] (5.2)

where \( \Gamma_D \cup \Gamma_N = \partial \Omega \) and \( \Gamma_D \cap \Gamma_N = \emptyset \) are boundaries of the domain, where we impose Dirichlet or Neumann boundary conditions. We are interested in only controlling the Neumann boundary conditions, while the Dirichlet boundaries are given. The case of \( \Gamma_D = \emptyset \) is allowed, in which case all the domain boundaries will be controlled.
Finally, we define the cost functional by:

$$J(\alpha, h) = \frac{\gamma_1}{2} \int_0^T \int_\Omega |\phi_d|^2 [\alpha(1 - \alpha)]^2 \, dx \, dt + \frac{\gamma_2}{2} \int_\Omega |\phi_d(T)|^2 [\alpha(T)(1 - \alpha(T))]^2 \, dx + \frac{\gamma_3}{2} \oint_{\Gamma_N} |h|^2 \, dx.$$  \hspace{1cm} (5.3)

The cost functional from (5.3) was inspired by a similar scenario in [Bernauer, 2011]. The main difference is that in [Bernauer, 2011] the interface is defined using a level set instead of a Heaviside marker function. In our definition of the cost functional, \(\phi_d(t, x)\) is a signed distance function that describes the desired interface movement, \(\alpha\) is the volume fraction, \(h\) is the control variable and \(\gamma_i \geq 0\) are given constant weights. The motivation for this choice of cost functional lies in the fact that the term:

$$|\phi_d|^2 (\alpha(1 - \alpha))^2,$$

will only be 0 when the interface defined by \(\alpha\) is on top of the interface defined by \(\phi_d\). We can easily see this since \(\phi_d\) has its 0 level set at the interface and \(\alpha(1 - \alpha)\) should be zero everywhere except at the interface.

Another choice for the cost functional would have been the more common tracking-type term:

$$|\alpha - \alpha_d|^2,$$

where \(\alpha_d\) is the volume fraction of the desired interface. However, this formulation does not lead to a well-behaving gradient descent. To see why this is, we take the simple case in which our current interface is a sphere centered at a point \(x\) and the desired interface is a sphere centered at \(x_d\) such that \(\|x - x_d\| > R\), where both spheres have a radius \(R\). In this scenario, the difference \(|\alpha - \alpha_d|^2\) will be the same for all \(x\) satisfying the above condition, which implies that we will not obtain a useful descent direction unless the two spheres intersect.

Finally, our optimization problem is given by:

$$\begin{cases} \min_{h \in H_{ad}} J(\alpha(h), h), \\ \text{subject to (5.1) and (5.2)}, \end{cases}$$  \hspace{1cm} (5.4)

where \(\alpha\) and \(\varphi\) are the state variables and the Neumann boundary condition \(h\) on \(\varphi\) is the control variable. The constraint set \(H_{ad}\) for the Neumann boundary conditions on the velocity potential is given by the following two conditions:
• $h$ must be bounded:

$$\|h\| \leq M.$$  \hfill (5.5)

The boundedness of $h$ is also handled to a certain degree by the $\gamma_3$ relaxation term in (5.3), but we must also strictly impose it as a constraint. This is necessary to make sure that the velocity stays bounded, which will allow us to keep a constant time step during the numerical experiments.

• if $\Gamma_D = \emptyset$, $h$ must satisfy the necessary condition for the existence of a solution to (5.2):

$$\int_\Omega \nabla \cdot \nabla \varphi \, dx = \oint_{\partial \Omega} \nabla \varphi \cdot n \, dx = \oint_{\partial \Omega} h(x) \, dx = 0.$$  \hfill (5.6)

As given, the constraint set $H_{ad}$ is a closed and convex set. Unfortunately, the cost functional itself is not convex for all $\gamma_i$ and choices of boundary conditions on $\varphi$. This can easily be seen because the term $(\alpha(1-\alpha))^2$ is not convex. As an example, take $\Gamma_D = \emptyset$ (i.e. all boundaries are controlled) and $\gamma_1 = \gamma_3 = 0$. In this case we can easily come up with the following two scenarios that set the remaining interface term to 0 at the final time $t = T$:

1. The interface described by $\alpha$ coincides with the desired interface $\phi_d$.

2. The interface in $\Omega$ is given by $\alpha \equiv 0$ or $\alpha \equiv 1$, i.e. the interface was moved outside the domain. Both a full or an empty phase leads to $\alpha(1-\alpha) \equiv 0$ on $\Omega$.

Numerically, the first case is less problematic, since the product $|\phi_d|^2(\alpha(1-\alpha))^2$ will never be exactly zero when the two interfaces coincide (either because of diffusion or because of a mollified interface model), so the second scenario is the single (undesirable) minimum. To ensure that this does not happen we have several choices:

• We can make sure that $\Gamma_D \neq \emptyset$ so that the resulting velocity field cannot force the interface to exit the domain, unless specifically desired.

• We can add a mass conservation term to the cost functional that penalizes differences between the mass at time $t = 0$ and time $t = T$, so that one-phase solutions are not viable.

• We can forgo the use of $\alpha$ as a Heaviside function and define the interface using level set methods. In this case, the signed distance function will naturally penalize interfaces that are far away from the desired location, even those which are not in the domain, etc.
5.2 Continuous Adjoint Equations

In the previous section we have defined our optimization problem and discussed several issues that may arise from the proposed formulation. We can now proceed to define the adjoint equations in a continuous setting. The formal way to derive the adjoint equations and optimality condition is using Lagrange relaxation. For problem (5.4), the Lagrangian is given by:

\[
\mathcal{L} = J - \int_0^T \int_\Omega \alpha^* \left( \alpha_t + \nabla \varphi \cdot \nabla \alpha \right) \, dx \, dt
- \int_\Omega \alpha^*_0 \left( \alpha(0, x) - \alpha_0(x) \right) \, dx
- \int_0^T \int_\Gamma \alpha^*_D \left( \alpha - \alpha_D \right) \, dx \, dt
- \int_0^T \int_\Omega \varphi^* \Delta \varphi \, dx
- \int_\Gamma_D \varphi^*_D \left( \varphi - g \right) \, dx
- \int_\Gamma_N \varphi^*_N \left( \nabla \varphi \cdot n - h \right) \, dx,
\]

where \( \alpha^*, \alpha^*_0, \alpha^*_D, \varphi^*, \varphi^*_D \) and \( \varphi^*_N \) are the associated Lagrange multipliers / adjoint variables.

We will now proceed to compute the derivatives of the Lagrangian with respect to its arguments. We have seen in Chapter 2 that the derivatives with respect to the adjoint variables will retrieve our constraints given by (5.1) and (5.2), so we will only look at the derivatives with respect to \( \alpha, \varphi \) and \( h \) to obtain the adjoint equations and the optimality condition, respectively.

The complete derivation of the following equations is given in Appendix F. Without further ado, we have the following advection equation for the adjoint volume fraction:

\[
\begin{aligned}
\begin{cases}
-\alpha^*_t - \nabla \cdot (\alpha^* \nabla \varphi) = \gamma_1 S(\alpha), & (t, x) \in [0, T] \times \Omega, \\
\alpha^*(T) = \gamma_2 S(\alpha(T)), & x \in \Omega, \\
\alpha^*(\nabla \varphi \cdot n) = 0, & (t, x) \in [0, T] \times \partial \Omega \setminus \Gamma,
\end{cases}
\end{aligned}
\]

where the source term comes from the derivative of the cost functional and is given by:

\[
S(\alpha) = |\phi_d|^2 \alpha (1 - \alpha)(1 - 2\alpha).
\]

Note that the boundary conditions are only defined on the inflowing boundaries of the adjoint equation. Since the adjoint is solved backward in time, the outflowing boundaries of the state equation become inflowing boundaries of the adjoint equation (see Figure F.1).
We also have the following Poisson equation for the adjoint velocity potential:

\[
\begin{cases}
\Delta \varphi^* = \int_0^T \nabla \cdot (\alpha^* \nabla \alpha) \, dt, & x \in \Omega, \\
\varphi^* = 0 & x \in \Gamma_D, \\
\nabla \varphi^* \cdot n = \int_0^T \alpha^* \nabla \alpha \cdot n \, dt & x \in \Gamma_N.
\end{cases}
\] (5.9)

Note that the boundary conditions on the adjoint velocity potential automatically satisfy the condition for the existence and uniqueness of a solution when \(\Gamma_D = \emptyset\) and only Neumann boundary conditions are used. Furthermore, unless the interface intersects with the domain boundary, the velocity potential will likely also have homogeneous Neumann boundary conditions.

Finally, we have the optimality condition:

\[
h = \left. \frac{1}{\gamma_3} \varphi^* \right|_{\Gamma_N}.
\] (5.10)

We also have that the gradient of the cost functional is given by:

\[
\nabla h \mathcal{J} = \left. \gamma_3 h - \varphi^* \right|_{\Gamma_N}.
\] (5.11)

### 5.2.1 Formal Description

Until this point we have not rigorously discussed the problem we have posed. Given the discontinuous solutions to the advection equation and the many derivatives, we must be more exact in our definitions to make sure that the problem is well-posed. To this end we will follow many of the ideas presented in [Tröltzsch, 2010], where we point the reader for more details on the vector spaces and results we will invoke.

There are many parts in our optimization that need to be well-defined. We have to define solutions to the advection equation (5.1), its adjoint (5.8), the Laplace equation for the velocity potential (5.2) and its adjoint that solves a Poisson equation (5.9). We will also need to further analyze the cost function to make sure it is well-defined, given the solutions to all the constraints. We will now proceed to look at the weak formulation of each equation and find proper spaces for weak solutions.

For the velocity potential, equation (5.2) requires \(\varphi \in C^2(\Omega)\) for strong solutions, but this result is obviously too restrictive. However, by multiplying the Laplace equation by a test function \(v \in C^\infty(\Omega)\) satisfying \(v = 0\) on \(\Gamma_D\), we have that:

\[
\int_\Omega \Delta \varphi v \, dx = 0,
\]
which can be transformed into the following weak formulation by integration by parts:

\[ \int_\Omega \nabla \varphi \cdot \nabla v \, dx = \int_{\Gamma_N} hv \, dx. \]

Given this weak formulation, we can say that solutions to the Laplace equation are in \( \varphi \in V = H^1(\Omega) \), for all \( h \in L^2(\Gamma_N) \), where \( \nabla \) is to be understood in the sense of weak derivatives. For completeness, we also mention that \( g \in H^{1/2}(\Gamma_D) \), where the space \( H^{1/2} \) is defined as the image of the trace operator and is a subset of \( L^2 \). The existence and uniqueness of such solutions can be proved by using the Lax-Milgram theorem with the vector spaces defined above (see [Tröltzsch, 2010]).

The adjoint velocity potential equation (5.9) is very similar. For all \( v \in C^\infty(\Omega) \) with \( v \equiv 0 \) on \( \Gamma_D \), the weak formulation is:

\[ \int_\Omega \Delta \varphi^* v \, dx = \int_0^T \int_\Omega \nabla \cdot (\alpha^* \nabla \alpha) v \, dx \, dt \]

\[ \Rightarrow \int_{\Gamma_N} (\nabla \varphi^* \cdot n) v \, dx - \int_\Omega \nabla \varphi^* \cdot \nabla v \, dx = \int_{\Gamma_N} \left( \int_0^T \alpha^* \nabla \alpha \cdot n \, dt \right) v \, dx - \int_0^T \int_\Omega \alpha^* \nabla \alpha \cdot \nabla v \, dx \, dt \]

\[ \Rightarrow \int_\Omega \nabla \varphi^* \cdot \nabla v \, dx = \int_0^T \int_\Omega \alpha^* \nabla \alpha \cdot \nabla v \, dx \, dt. \]

Symmetry is very desirable in optimization problems, so we would want that \( \varphi^* \in V = H^1(\Omega) \), just like the velocity potential. This is only possible if the right-hand side is well defined in \( L^2(\Omega) \). As such, we need to require that \( \alpha \in H^1(\Omega) \) and at least \( \alpha^* \in L^2(\Omega) \) for almost every \( t \in [0, T] \). For now, we will assume this is true, so that the adjoint velocity potential does indeed have weak solutions in \( H^1(\Omega) \). Given the chosen space for \( \alpha \) and \( \alpha^* \), the boundary conditions on the adjoint velocity potential are also well-posed, so we have a unique solution by the Lax-Milgram theorem, like before. We also note that given \( \varphi^* \in H^1(\Omega) \), the optimality condition (5.10) is not well-defined unless understood through the use of the trace theorem.

Next we will look at the advection equation (5.1). For all \( v \in C^\infty(\Omega) \), we have that:

\[ \int_0^T \int_\Omega (\alpha_t + u \cdot \nabla \alpha) v \, dx \, dt = 0 \]

\[ \Leftrightarrow \int_0^T \int_\Omega v \alpha_t \, dx \, dt = - \int_0^T \int_\Omega u \cdot \nabla \alpha v \, dx \, dt. \]

The advection equation itself does not require a lot of regularity in the solution, so usually a measure space would suffice for the above weak formulation to be well posed after integrating by parts. However, we have already seen that \( \alpha \) is required to be in \( H^1(\Omega) \) for the adjoint velocity potential equation to be well-posed. As such, the right-hand side of our weak formulation is always well-defined, given this additional regularity.
The only problem we are left with is defining the proper space for $\alpha_t$ for a fixed $t$. Following the reasoning from [Tröltzsch, 2010], we notice that the right-hand side can be considered as a linear functional $F(v)$ in $L^2(\Omega)^*$ for almost every $t$, if we take the test functions in $v \in L^2(\Omega)$. Choosing test functions from $L^2(\Omega)$ instead can be easily motivated considering that $C^\infty(\Omega)$ is dense in $L^2(\Omega)$. Given that $L^2$ is a Hilbert space, with itself as a dual, $\alpha_t$ can also be considered as an element of $L^2$. So, we have that:

$$\alpha \in U = \{ u \mid u \in L^2([0,T];H^1(\Omega)), u_t \in L^2([0,T];L^2(\Omega)) \},$$

which could also be considered as the space $H^1([0,T] \times \Omega)$. We also know that $u = \nabla \varphi \in (L^2(\Omega))^d$, so from [DiPerna, 1989] we can say that the advection equation (5.1) has a unique solution, for $\alpha_0 \in L^2(\Omega)$ and $\alpha_D \in L^2([0,T];H^{1/2}(\Gamma))$.

The adjoint advection equation (5.8) can be analyzed in a similar manner. The treatment of the additional source term is automatic given the regularity of $\alpha$. Therefore, we also have that $\alpha^* \in U$. Note that the signed distance function $\phi_d$ is at least continuous $\phi_d \in L^2([0,T];C(\Omega))$, which we can pose on $L^2(\Omega)$ by a density argument.

We have now defined the solutions to our various equations in a more rigorous manner. The remaining problem is that of the cost function. We have said that the volume fraction is a Heaviside function, but this implies (by most definitions of a Heaviside function) that $\alpha(1 - \alpha) \equiv 0$, for all $(t,x) \in [0,T] \times \Omega$. Furthermore, the Heaviside function does not belong to $H^1(\Omega)$ since its first derivative is the Dirac Delta function and can only be understood in the sense of distributions.

Given the discussion in the previous chapter regarding mollifiers of a fixed thickness, it is natural to introduce these ideas directly into the continuous formulation. We will say that our volume fraction $\alpha$ is in fact a mollified Heaviside function (with any of the mollifiers in Chapter 4). We know that the mollifiers we have defined are at worst $C^1([a,b])$, in the case of the sine-based mollifier (4.20), and at best $C^\infty([a,b])$ in the case of the hyperbolic tangent-based mollifier (4.2). For Lipschitz domains $\Omega$, we know that $C^1(\Omega)$ is dense in $H^1(\Omega)$, so it can be embedded in it, thus validating our choice for a functional space for $\alpha$ and $\alpha^*$.

To summarize, our optimization problem is now defined as:

$$\begin{align*}
\min_{h \in H_{\text{ad}}} & \mathcal{J}(\alpha(h), h), \\
\text{for } h & \in H_{\text{ad}} \subset H, \alpha \in U, \varphi \in V, \\
\text{subject to } (5.1) \text{ and } (5.2),
\end{align*}$$

where $H_{\text{ad}}$ is defined by the two constraints (5.5) and (5.6) and is a closed and convex subset of $H = L^2(\Omega)$.
and $U, V$ have been defined above. We also have that the cost function is defined by:

$$ J : U \times H \rightarrow \mathbb{R}. $$

For completeness, we define the Lagrangian as the continuous linear functional:

$$ L(\alpha, \varphi, h, \alpha^*, \alpha_D^*, \varphi_D^*, \varphi_N^*) : U \times V \times H(\Omega) \times H^{1/2}(\Gamma) \times V \times H^{1/2}(\Gamma_D) \times H \rightarrow \mathbb{R}, $$
given by the following well-defined formulation (as opposed to (5.7)):

$$ L = J - \int_0^T \int_\Omega \alpha^* \left( \alpha_t + \nabla \varphi \cdot \nabla \alpha \right) \, dx \, dt - \int_\Omega \alpha_0^* (\alpha(0, x) - a_0(x)) \, dx - \int_\Gamma \alpha_D^* (\alpha - \alpha_D) \, dx \, dt - \int_{\partial \Omega} \varphi^* (\nabla \varphi \cdot n) \, dx + \int_\Omega \nabla \varphi^* \cdot \nabla \varphi \, dx - \int_{\Gamma_D} \varphi_D^* (\varphi - g) \, dx - \int_{\Gamma_N} \varphi_N^* \left( \nabla \varphi \cdot n - h \right) \, dx. $$

### 5.3 Discrete Adjoint Equations

We will now proceed with defining the discrete adjoint formulation that will be used to implement and solve the optimization problem (5.4). We will be using the notation from Chapter 3 and extend it to 3D domains in the usual manner.

In the description of the Lagrangian (5.7) we have used the standard $L^2$ inner product in the various terms. Discretely, we must define a set of equivalent inner products to use in the discrete formulation. First, we define an inner product on $\Omega_h$, the discrete domain, by:

$$ (u, v)_{\Omega_h} = u^T V v = \sum_{ijk} V_{ijk} u_{ijk} v_{ijk}, $$

where $V_{ijk}$ is a cell volume, which implies that $V$ is a positive-definite weight matrix corresponding to the volumes. Similarly, we define an inner product on the boundary $\Gamma_h^b$:

$$ (u, v)_{\Gamma_h^b} = u^T A v = \sum_{jk} A_{jk} u_{jk} v_{jk}, $$

where $A_{jk}$ is a face area in a cell that intersects the boundary, which implies that $A$ is a positive-definite weight matrix corresponding to the face areas. The associated norms are induced by the inner products in
the usual way:
\[ \| u \|_{\Omega^h} = \sqrt{(u, u)_{\Omega^h}} \quad \text{and} \quad \| u \|_{\Gamma^h} = \sqrt{(u, u)_{\Gamma^h}}. \]

Using this notation, we define our discrete cost functional as:
\[ J^h(\alpha, h) = \frac{\gamma_1}{2} \sum_{m=0}^{M-1} \Delta t \| \phi^m_d \alpha^m (1 - \alpha^m) \|^2_{\Omega^h} + \frac{\gamma_2}{2} \| \phi^M_d \alpha^M (1 - \alpha^M) \|^2_{\Omega^h} + \frac{\gamma_3}{2} \| h \|^2_{\Gamma^h}, \tag{5.12} \]
where the products \( \alpha(1 - \alpha) \) are defined pointwise.

## 5.3.1 Unsplit Scheme

We define a discretization of (5.1) as:
\[ \alpha^{m+1} = \alpha^m + \Delta t V^{-1} A(\alpha^m, \varphi, h), \tag{5.13} \]
where the space discretization operator \( A \) can be further broken into terms pertaining to each dimension:
\[ A(\alpha^m, \varphi, h) = -[X(\alpha^m, \varphi, h) + Y(\alpha^m, \varphi, h) + Z(\alpha^m, \varphi, h)]. \]

In a very general setting, we can define:
\[ X_{ijk}(\alpha^m, \varphi, h) = \left( A_{i+\frac{1}{2}, j, k} f^m_{i+\frac{1}{2}, j, k} (\alpha^m, \varphi, h) - A_{i-\frac{1}{2}, j, k} f^m_{i-\frac{1}{2}, j, k} (\alpha^m, \varphi, h) \right) - \]
\[ \alpha^m_{ijk} \left( A_{i+\frac{1}{2}, j, k} u^m_{i+\frac{1}{2}, j, k} - A_{i-\frac{1}{2}, j, k} u^m_{i-\frac{1}{2}, j, k} \right) \tag{5.14} \]
in the \( x \) direction, with similar definitions for the other dimensions. Note that we have opted for a scheme that does not assume an incompressible velocity field. This is because the velocity field will be computed from the velocity potential and the accuracy with which it will satisfy the divergence-free condition will depend on the accuracy of the Poisson solver and the finite difference approximation that is used. To avoid any issues arising from possibly insufficient accuracy, we use the formulation above with the understanding that the last term is expected to be \( \ll 1 \).
The discrete Laplace equation for the velocity potential is then:

\[
\Delta y_j \Delta z_k \left( \frac{\varphi_{i+1,j,k} - \varphi_{ijk}}{\Delta x_{i+1/2}} - \frac{\varphi_{ijk} - \varphi_{i-1,j,k}}{\Delta x_{i-1/2}} \right) + \Delta x_i \Delta z_k \left( \frac{\varphi_{i,j+1,k} - \varphi_{ijk}}{\Delta y_{j+1/2}} - \frac{\varphi_{ijk} - \varphi_{i,j-1,k}}{\Delta y_{j-1/2}} \right) + \Delta x_i \Delta y_j \left( \frac{\varphi_{i,j,k+1} - \varphi_{ijk}}{\Delta z_{k+1/2}} - \frac{\varphi_{ijk} - \varphi_{i,j,k-1}}{\Delta z_{k-1/2}} \right) = 0,
\]

which we will simply write as:

\[
L \varphi = b(h),
\]

where \( b \) is the vector of boundary conditions, defined as a function of the control variable \( h \). Note that, as in the continuous case, if \( \Gamma_D = \emptyset \), the Laplace operator \( L \) is singular (with a nullspace of dimension 1 containing constant vectors), so the right-hand side must satisfy an equivalent of condition (5.6):

\[
\sum_{ijk} b_{ijk}(h) = 0.
\]

The discrete optimization problem can then be expressed as:

\[
\begin{aligned}
\min_{h \in H^d} & \quad J^h(\alpha, h), \\
\text{subject to } & \quad (5.13) \text{ and } (5.15).
\end{aligned}
\]

The discrete Lagrangian can be expressed as:

\[
L^h = J^h - \sum_{m=0}^{M-1} \left( \alpha^{*,m+1}, \alpha^{m+1} - \alpha^m - \Delta t V^{-1} A(\alpha^m, \varphi, h) \right)_{\Omega^h}
- (\alpha^{*,0}, \alpha^0 - \alpha_0)_{\Omega^h} - (\varphi^*, V^{-1}(L \varphi - b(h)))_{\Omega^h}.
\]

Now that the state equations are discretized, we can proceed to define the adjoint equations and the optimality condition. The adjoint advection equation is given by:

\[
\alpha^{*,m} = \alpha^{*,m+1} + \Delta t V^{-1} \left[ \frac{\partial A}{\partial \alpha^m}(\alpha^m, \varphi, h) \right]^T \alpha^{*,m+1} + \Delta t S(\alpha^m),
\]

with the final state:

\[
\alpha^{*,M} = S(\alpha^M),
\]
where the source term is obtained from (5.12):

\[ S_{ijk}(\alpha^m) = |\phi^m_{d,ijk} \alpha^m_{ijk}(1 - \alpha^m_{ijk})(1 - 2\alpha^m_{ijk}). \]

The adjoint velocity potential equation is given by:

\[ L\varphi^* = \sum_{m=0}^{M-1} \Delta t \left[ \frac{\partial A}{\partial \varphi} (\alpha^m, \varphi, h) \right]^T \alpha^{*,m+1}. \]  

(5.18)

We can also obtain the optimality condition by differentiating with respect to \( h \):

\[ h = -A^{-1} \gamma_3 \left( \left[ \frac{db}{dh}(h) \right]^T \varphi^* + \sum_{m=0}^{M-1} \Delta t \left[ \frac{\partial A}{\partial h} (\alpha^m, \varphi, h) \right]^T \alpha^{*,m+1} \right). \]

The gradient can now be straightforwardly obtained from the optimality condition using the inner products we have defined:

\[ \nabla J^h = \gamma_3 h + A^{-1} \left( \left[ \frac{db}{dh}(h) \right]^T \varphi^* + \sum_{m=0}^{m+1} \Delta t \left[ \frac{\partial A}{\partial h} (\alpha^m, \varphi, h) \right]^T \alpha^{*,m+1} \right). \]  

(5.19)

### 5.3.2 Dimensional Splitting Scheme

We have seen in the previous section how to derive the adjoint equations and the necessary first-order conditions for optimality for an unsplit numerical scheme. However, given that the THINC family of schemes is essentially 1D, we must also investigate a formulation using dimensional splitting. For exposition purposes, we will only use a first-order Lie splitting scheme, but extensions to a second-order Strang splitting scheme are fairly straightforward. A scheme for a dimensionally split advection equation (5.1) is defined as:

\[
\begin{align*}
\alpha^{(1),m} &= \alpha^m + \Delta t V^{-1} A(\alpha^m, \varphi, h), \\
\alpha^{(2),m} &= \alpha^{(1),m} + \Delta t V^{-1} Y(\alpha^{(1),m}, \varphi, h), \\
\alpha^{m+1} &= \alpha^{(2),m} + \Delta t V^{-1} Z(\alpha^{(2),m}, \varphi, h),
\end{align*}
\]

where the dimensional operators are defined exactly as in (5.14) and simply take different arguments. Equation (5.15) remains unchanged, since there is no point in splitting the Laplacian. We will skip formulating the complete discrete Lagrangian for this formulation and go directly to the adjoint equations. The adjoint
The advection equation is given by:

\[
\begin{align*}
\alpha^{*,(2),m+1} &= \alpha^{*,m+1} + \Delta t V^{-1} \left[ \frac{\partial Z}{\partial \alpha^{(2),m}} (\alpha^{(2),m}, \varphi, h) \right]^T \alpha^{*,m+1}, \\
\alpha^{*,(1),m+1} &= \alpha^{*,(2),m+1} + \Delta t V^{-1} \left[ \frac{\partial Y}{\partial \alpha^{(1),m}} (\alpha^{(1),m}, \varphi, h) \right]^T \alpha^{*,(2),m+1}, \\
\alpha^{*,m} &= \alpha^{*,(1),m+1} + \Delta t V^{-1} \left[ \frac{\partial X}{\partial \alpha^{m}} (\alpha^{m}, \varphi, h) \right]^T \alpha^{*,(1),m+1} + \Delta t S(\alpha^m).
\end{align*}
\]

We can see that not only have the individual operators been transposed, but also the order in which the splitting is done has been completely mirrored. For the adjoint velocity potential, we get that:

\[
L \varphi^* = \sum_{m=0}^{M-1} \Delta t \left\{ \left[ \frac{\partial X}{\partial \varphi} (\alpha^m, \varphi, h) \right]^T \alpha^{*,(1),m+1} + \left[ \frac{\partial Y}{\partial \varphi} (\alpha^{(1),m}, \varphi, h) \right]^T \alpha^{*,(2),m+1} + \left[ \frac{\partial Z}{\partial \varphi} (\alpha^{(2),m}, \varphi, h) \right]^T \alpha^{*,m+1} \right\}.
\]

Finally, we have the following gradient:

\[
\nabla J^h = \gamma_3 h + A^{-1} \left( \left[ \frac{dB}{dh} \right]^T \varphi^* + \sum_{m=0}^{M-1} \Delta t \left\{ \left[ \frac{\partial X}{\partial h} \right]^T \alpha^{*,(1),m+1} + \left[ \frac{\partial Y}{\partial h} \right]^T \alpha^{*,(2),m+1} + \left[ \frac{\partial Z}{\partial h} \right]^T \alpha^{*,m+1} \right\} \right).
\]

This completes the description of the optimization problem, both discretely and continuously. We will only be solving the discrete problem using, as before, the discretize-then-differentiate method. The continuous formulation is meant to define the problem in rigorous terms and give a theoretical background to the underlying issues. Most importantly, an analysis of the continuous formulation has lead us to redefine the volume fraction \( \alpha \) in terms of a mollified Heaviside function, whose 0.5 level set would indicate the actual interface. This was necessary to ensure the existence and uniqueness of solutions for the state and adjoint equations, without which we could not hope to solve the optimization problem itself.

One major downside of the optimization problem (5.4) is that the cost function is not convex, so we cannot formally prove uniqueness of a solution in this setting. We will see that in practice uniqueness can be forced, in a certain sense, by limiting and influencing the possible descent directions with a cautious choice of Dirichlet boundary conditions in (5.2) and generally strictly positive coefficients \( \gamma_i \) in the cost functional (5.3).
5.4 Steepest Descent Algorithm

A potential algorithm for solving the full discrete optimality system is given by Algorithm 1. This is a very simple example of how the system could be solved. More complicated algorithms (e.g. a preconditioned nonlinear conjugate gradient method, a BFGS variant, etc.) can be implemented since we have an explicit formula for the cost functional and the gradient.

Algorithm 1: Optimization algorithm for (5.4).

Data: \( h^{(0)} \)

1. \( k \leftarrow 0; \)
2. while \( k < K \) do
   // Solve \( \alpha_{ijk}^m \) for all \( m \) and \( (i,j,k) \)
   3. while \( m < M \) do
      4. Solve (5.15) using \( h^{(k)} \) to get \( \varphi \);
      5. Advance \( \alpha^m \) to \( \alpha^{m+1} \) using (5.13);
      6. \( m \leftarrow m + 1; \)
   end
   8. Compute \( J^h \) using (5.12);
   // Check stopping conditions
   9. if \( |J^h| \leq \epsilon_1 \) then
      10. Stop;
   end
   12. if \( |J^h - J_{old}^h| \leq \epsilon_2 \) then
      13. Stop;
   end
   // Solve \( \alpha_{ijk}^{*,m} \) for all \( m \) and \( (i,j,k) \)
   15. while \( m > 0 \) do
      16. Advance \( \alpha_{ij}^{*,m} \) to \( \alpha_{ij}^{*,m-1} \) using (5.17);
      17. \( m \leftarrow m - 1; \)
   end
   // Update controls
   19. Compute the right-hand side of (5.18);
   20. Solve (5.18) for \( \varphi^* \);
   21. \( h^{(k+1)} \leftarrow h^{(k)} - \rho \nabla J^h; \)
   22. Apply constraints on \( h^{(k+1)} \);
   23. \( k \leftarrow k + 1; \)
end
5.5 Numerical Results

5.5.1 One-dimensional Tests

Test 1: Two Moving Droplets

For our first numerical test, we will directly apply the one dimensional THINC scheme to a given optimization problem. However, in 1D, the only incompressible and irrotational velocity field is a constant velocity field, so we need to slightly reformulate our optimization problem.

We define a new cost functional as:

\[ J(\alpha, c) = \frac{\gamma_1}{2} \int_{\Omega} |\phi_d(T)|^2 (\alpha(T)(1 - \alpha(T)))^2 \, dx + \frac{\gamma_2}{2} c^2, \]

where \( c \) is the velocity field, as in the previous chapters. Note that in this case we completely forgo the use of the velocity potential and the optimization of boundary conditions, since there is no need. We will not go through the complete derivation of the 1D optimality system since it is very similar to the results from previous chapters. The main difference is the gradient, which is given by:

\[ \nabla_c J = \gamma_2 c - \int_0^T \int_{\Omega} \alpha^* \alpha_x \, dx \, dt. \]

The discrete gradient is very similar:

\[ \nabla_c J^h = \gamma_2 c - \Delta t \sum_{m=0}^{M-1} \sum_i \alpha_i^{*,m+1} \left[ \frac{\partial f_{i+1/2}}{\partial c} - \frac{\partial f_{i-1/2}}{\partial c} \right]. \]

This completes the definition of our optimization problem. We have a discrete formulation (3.7), where we can use the THINC flux approximation (4.16) to obtain \( \alpha_i^m \), for all \( m \) and \( i \). Using the computed volume fraction, we can solve the adjoint equations as well and finally compute the above cost function and gradient.

Since this is a simple 1D problem where the controls are not on the boundary, we have a larger choice of optimization algorithms. For this particular test, we have used the TNC (Truncated Newton) algorithm that comes with the numpy library.

For the parameters of the problem, the desired interface will be given by the following signed distance
Figure 5.1: Initial condition of the advection equation (dashed) and the signed distance function to the desired interface (blue) at time $t = 0$.

function:

$$\phi_d(t, x) = \begin{cases} 
    x - (0.8 + \hat{c}t) & x < -0.7 + \hat{c}t, \\
    -0.6 + \hat{c}t - x & x \in [-0.7 + \hat{c}t, -0.45 + \hat{c}t], \\
    x - (0.3 + \hat{c}t) & x \in [-0.45 + \hat{c}t, -0.2 + \hat{c}t], \\
    -0.1 + \hat{c}t - x & x > -0.2 + \hat{c}t, 
\end{cases}$$

where $\hat{c}$ is the velocity at which the desired interface moves. We also take $\Omega = [-1.5, 1.5]$ with a final time $T = 1$, a desired velocity $\hat{c} = 1$ and a CFL condition of 0.9. The initial condition for the advection equation is given by mollifying the above signed distance function:

$$\alpha(0, x) = \frac{1}{2} \left( 1 + \tanh \left( \frac{\beta}{\Delta x} \phi_d(0, x) \right) \right),$$

which can be seen in Figure 5.1. The parameters of the THINC scheme are $\eta = 12.5$ and $\epsilon = 0.01$ (which gives $\beta = 0.18$). Finally, we take an initial estimate for the iteration optimization algorithm of $c^{(0)} = 0.1$ and weights $\gamma_1 = 1.0, \gamma_2 = 0.0$, thus completing the description of the problem.

The solution for $c = 0.1$ and $c = 1.0$ of the state and adjoint equations at time $t = T$ can be seen in Figure 5.2. As expected, for $c = 1.0$, the adjoint variable, which makes up the bulk of the gradient, is uniformly zero, indicating an optimal solution. We cannot say much about the adjoint solution when $c \neq 1.0$, but it is still important to realize that it represents the sensitivity of the cost functional at a particular point.
Figure 5.2: 1D Test: Solution at time $t = T$ for the initial guess $c = 0.1$ (left) and the optimal solution $c = 1.0$ (right).

$(t, x)$ with respect to the velocity $c$.

The behavior of the optimization procedure itself can be seen in Figure 5.3. The truncated Newton algorithm that we have used convergences very rapidly and most of the function evaluations we see in Figure 5.3 are in fact performed in the line search part of the algorithm. For this simple 1D test, we can conclude that our optimization problem is indeed well-posed and gives the expected result.

Figure 5.3: 1D Test: Function evaluations during the optimization algorithm for the cost functional (left) and the velocity field (right).

5.5.2 Two-dimensional Tests

Having seen that the proposed optimization problem is well-posed and we can obtain a correct optimal solution, we will move on to some more complicated examples. In this section we will look at a couple of 2D
tests on a domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$. We will once again put all the focus on the solution at the final time $T = 0.4$ with the following weights:

$$\gamma_1 = 0.0, \gamma_2 = 10^4 \text{ and } \gamma_3 = 10^{-4}$$

in the cost functional (5.3). A large weight is attached to the solution at time $T$ so that the adjoint variable has a larger magnitude, which translates to a larger gradient. Given the slow steepest descent algorithm (1) we have proposed, generating a larger gradient through the adjoint variable in this way has lead to faster convergence. Of course, a more robust approach should implement a line search algorithm to complement the steepest descent method.

On the numerical side, we will be using an unsplit upwind scheme for the flux approximations with a CFL condition of 0.9. The expected velocity field will continue to be uniformly 1. For this reason, we set the constraint (5.5) to:

$$\|h\| \leq 2,$$

and compute a constant time step using $u = (2, 2)$ as a velocity field.

**Test 2: One Boundary Controlled / One Droplet**

The desired interface for the first test contains a single droplet moving uniformly in the $x$ direction. The signed distance function for such an interface is given by:

$$\phi_d(t, x) = r - \|c(t) - x\|,$$

where $r = 0.15$ is the droplet radius and

$$c(t) = (-0.35 + \hat{u}t, \hat{v}t)$$

is the droplet center that moves with a velocity $\hat{u} = (\hat{u}, \hat{v}) = (1, 0)$. The initial condition for the advection equation is given by:

$$\alpha(0, x) = \frac{1}{2} \left( 1 + \text{tanh} \left( \frac{\beta}{\Delta x} \phi_d(0, x) \right) \right),$$
The parameters for the mollifier are a slope thickness of $\eta = 1.0$ and a cutoff value of $\epsilon = 0.01$, which gives a slope steepness of $\beta = 2.3$. The boundary conditions on the velocity potential are set up as follows:

$$
\begin{align*}
\Gamma_N &= \{-0.5\} \times [-0.5, 0.5], \\
\Gamma_D &= \partial \Omega \setminus \Gamma_N,
\end{align*}
$$

so that the left boundary of the domain is controlled with Neumann boundary conditions and all the remaining boundaries are uncontrolled with Dirichlet boundary conditions. The expected exact solution for the velocity potential is:

$$\varphi(x) = \hat{u}x + \hat{v}y$$

The Dirichlet boundary conditions can be computed from the above solution. The initial guess for the remaining Neumann boundary conditions, which are to be optimized, is given by a perturbation around the known optimal solution $h = -1$:

$$h^{(0)}(y) = -1 + \cos(2\pi y).$$

There are many variables to be analyzed in the full problem (5.4). We will start by looking at the performance of the optimization algorithm. We have in Figure 5.4 the decrease in the cost function and in Figure 5.5 the decrease in the norm of the gradient and the norm of the control. Note that we have used the discrete $\ell_2$ norm, given by:

$$v = \sqrt{\sum_i |v_i|^2}.$$
We can see in Figure 5.4 that both of the important terms of the cost functional are decreasing during the optimization. As expected with a steepest descent algorithm, progress is slow, but ultimately a minimum value is attained. The minimum value is not zero for multiple reasons: the interface is mollified, so it will not be completely counteracted by 0 level set of the signed distance function, and the upwind scheme is severely diffusive, which magnifies the effect. In Figure 5.5 we can see the norm of the gradient, which does in fact tend to 0. Our gradient descent algorithm 1 has a fixed descent step, so we can see, towards the end, a small bump in the gradient norm that corresponds to a large step that made the function zig-zag before it got corrected again. This type of problems is easily solved with a line search approach.

The optimal position of the volume fraction at time $t = T$, as obtained by the optimization algorithm, can be seen in Figure 5.6. We can see that the expected linear velocity potential was achieved in Figure 5.7.
Figure 5.7: Test 2: Velocity potential for the initial (left) and the optimal (right) control $h$.

Figure 5.8: Test 2: Adjoint volume fraction for the initial (left) and the optimal (right) control $h$. The white circle represents the desired interface location.

The adjoint variables themselves are also given in Figure 5.8 and Figure 5.9. We can easily see that the final state of the adjoint volume fraction corresponds to the expected result: it is zero around the zero level set of the desired interface and has a large error outside where the upwind scheme has diffused the solution. In the case of the adjoint velocity potential a similar analysis can be done: we can see that the larger values can be found on the $x$ axis, where the adjoint volume fraction has non-zero values.

To analyze the gradient, to some extent, we can look at these adjoint variables. One important hint that the gradient is correct is given by the adjoint velocity potential from Figure 5.9. We can see there that large negative values are present in the center of the left boundary of the domain. This is indeed the expected result since that is where the largest error in the initial guess of the control $h$ resides, i.e. the top of the cosine bump. Towards the end, we can see that the value of the adjoint velocity potential becomes
Figure 5.9: Test 2: Adjoint velocity potential for the initial (left) and the optimal (right) control $h$.

much more uniform on the left boundary, albeit not zero. We cannot expect the adjoint velocity potential to become exactly zero on this boundary because the problem is not solved very accurately (again due to diffusion).

In conclusion, this second test we have presented also gave good convergence results and validated our previous analysis and formulation.

Test 3: All Boundaries Controlled / Two Droplets

A second test we will perform is one where all the boundaries of the domain are controlled, i.e. $\Gamma_N = \partial\Omega$.

In this case, we keep the desired interface movement from the previous case, but introduce the following signed distance function:

$$
\phi_{\text{droplets}}(x) = \max(r_1 - \|c_1 - x\|, r_2 - \|c_2 - x\|),
$$

representing two droplets of radii $r_1 = r_2 = 0.15$ centered at:

$$
\begin{align*}
  c_1 &= (-0.25, -0.25), \\
  c_2 &= (-0.25, 0.25).
\end{align*}
$$

Using this new function, we define our interface as before:

$$
\alpha(0, x) = \frac{1}{2} \left( 1 + \tanh \left( \frac{\beta}{\Delta x} \phi_{\text{droplets}}(x) \right) \right),
$$
where the slope thickness is given by $\beta = 2.3$. As an initial guess for the controlled boundaries, we use a random vector, as generated by the uniform distribution $U(-0.6,0.6)$.

This particular test case is more interesting because it will highlight an inherent problem in our formulation: the fact that the cost functional is not convex. We will see that we actually have multiple solutions and the algorithm will converge to a different one given different random inputs.

Figure 5.10: Test 3: Potential solutions when all boundaries are controlled. The white circle represents the desired interface location.

Figure 5.11: Test 3: Potential solution when all boundaries are controlled. The white circle represents the desired interface location.

The various solutions can be see in Figure 5.10 and Figure 5.11. In particular, since we have two droplets and the desired interface is a single droplet, we find multiple solutions: the top droplet goes out of the domain, the bottom droplet goes out of the domain or both of them exit the domain. The last solution is obviously not desired, but the other two are obviously good results. Another solution that we have not
managed to obtain (likely due to the fact that the velocity field is irrotational) is the merging of the two droplets into a single one at the final time.

Finally, in Figure 5.12, we have the behavior in norm of the gradient and the controls on all boundaries for the first result in Figure 5.10. We can see that, unlike in the previous test case, the behavior of both is significantly more erratic. Similarly, the cost functional itself oscillates during the optimization, as can be seen in Figure 5.13.

![Figure 5.12: Test 3: Norm of the gradient (left) and the control variable (right) during the optimization process.](image)

The most interesting result from Figure 5.13 is that the distributed part of the cost functional actually increases during the optimization. This can be motivated by the fact that, initially, the two droplets were closer to the desired interface, but later they got further away and the error grew, until one of them went outside of the domain. Note, however, that we have chosen $\gamma_1 = 0$ for these tests, so the growth of that part of the cost functional is purely informational.

### 5.6 Conclusions

This concludes our testing of the motion planning problem and formulation. The analysis we have performed in this chapter and the tests we have presented give us very important hints into the optimization of interface and multiphase flows. One of the most important points to make is that it is increasingly difficult to formulate a coherent cost functional in such a scenario. This is due in part to the sharp interface formulation we have started with, that mandated the use of a Heaviside function. To have a well-defined optimization problem, we have defined the volume fraction as a smoothed Heaviside function from the beginning. This is consistent with our intended use of the THINC family of schemes that also mandates a smooth interface. Furthermore,
from a physical point of view, we can also say that the interface between the two fluids is never infinitely sharp. In the case where a sharp interface formulation is desired, the cost functional could still be defined by convoluting the volume fraction $\alpha$ with a mollified Dirac delta bump, which should lead to similar results.

We have shown that the optimization of a fluid-fluid interface is indeed viable in multiple scenarios. However, as conjectured in the continuous formulation, our proposed cost functional is not convex. We have seen this in a simple test case where all the domain boundaries were controlled and different random starting points were used for the optimization algorithm. The problem is not insurmountable and can be solved by imposing further constraints on possible solutions, such as volume conservation in the considered domain. Overall, the results presented here are encouraging when considering the optimization of two-phase flows modeled by a marker function.
Chapter 6

Conclusions

The work presented in this thesis deals with the most obvious issues that can arise in the optimization of multiphase flows using the discretize-then-differentiate approach to adjoint methods. In general, we believe that a well-posed problem involving sharp interfaces (at least in the limit) can be formulated in this scenario.

We have studied the ability of discrete adjoint schemes to correctly converge to weak solutions with various types of discontinuities. The first case we have looked at is related to contact discontinuities in the simplified setting of linear advection equations. Discretely, even in this simple case, convergence of the adjoint is not a given because we are differentiating numerical schemes that can have any number of issues (e.g. highly nonlinear, non-differentiable, etc.). We have seen that, for two commonly used numerical schemes, the adjoint correctly advects the discontinuities backwards in time. However, issues with the linearized fluxes keep the results from being as accurate as what we would obtain by separately discretizing the adjoint equations, in the differentiate-then-discretize fashion. Specifically, in the case of a flux limited scheme, the adjoint is no longer monotonicity preserving and exhibits the oscillations common to dispersive second-order schemes.

The second case we have looked at involves the appearance of shocks in nonlinear hyperbolic conservation laws. The case of shocks is significantly more complicated than that of contact discontinuities because first-order variations of solutions to nonlinear hyperbolic conservation laws require us to take the shock position into account. The combined first-order variation of the actual solution and the shock position is called a generalized tangent vector. Using the theory of generalized tangent vectors, we have presented the derivation of the adjoint equations, which contained both equations for the adjoint variable and the evolution of an adjoint shock position. The main takeaway from this derivation is that the adjoint is necessarily continuous along all characteristics leading into the shock. Furthermore, the adjoint has an extra boundary condition at the shock position defining the value along the aforementioned characteristics. Numerically, this can be approached in two ways: the extra boundary condition can be imposed explicitly or special numerical schemes can be designed to approximate the boundary condition implicitly. We have studied the second case using a set of commonly used schemes. We have seen that convergence to the correct weak adjoint solution is
possible if the shock is smeared over an increasing number of cells during the forward solve. Unfortunately, most commonly used schemes do not have this property, so, unsurprisingly, neither the classic upwind scheme nor a flux limited scheme exhibit mesh convergence. However, the approximated solutions oscillate around the correct exact solution, so in some practical situations (e.g. weak shocks), these schemes can still be acceptable.

Another issue that was studied for both contact discontinuities and shocks is related to the complete vs. incomplete differentiation of the numerical fluxes. Since most numerical fluxes have non-differentiable parts, it may be beneficial to simply consider some parts constant during differentiate, especially if done manually. We have seen that, in the case of the fairly nonlinear flux limited schemes, differentiating the flux limiter itself with respect to the problem variables yielded a very small benefit. In the case of the advection equation, complete differentiation even unearthed the oscillations coming from the Lax-Wendroff flux used in the combination.

Having determined some general intuition about the effect of discontinuities on the discrete adjoint equations, we have moved to another family of numerical schemes more commonly used in multiphase flows. The THINC family of schemes was chosen for our investigation because of its simplicity compared to other commonly used anti-diffusive schemes and because all important components of the scheme are differentiable. We have seen that the classic definition of the THINC scheme leads to unexpected features in the adjoint, which are due to the fact that the scheme is not well-defined in pure states. The main issue was the need to switch to a different numerical scheme in pure states, which leads to large discontinuities in the flux derivatives. We have solved this problem by proving that the THINC scheme can be approximated, to leading order, in the limit of pure states. An natural extension to the results regarding the THINC scheme is to attempt to find a mollifier with well-defined values in the limit of pure states, without the need for further modifications. To this end, we present a new family of numerical schemes called COMIC, based on compact mollifiers, in the sense that pure states are achieved in a finite interval. The sine-based COMIC scheme we have presented is significantly more complex than the THINC scheme, due to the use of root finding algorithms, but it gives better results in some of the numerical tests we have performed, notably for sharp interfaces.

The main result of our analysis of the anti-diffusive schemes is that the linearized and adjoint equations converge everywhere except at the interface. In the linearized scenario, smoothed Dirac delta functions appear at the interface and in the adjoint solution the entire solution at the interface becomes a plateau at the average value 0.5. The extreme case of this can be seen with the COMIC scheme. However, we have performed a simple optimization test with a tracking-type cost functional and we have seen that both the
THINC and the COMIC scheme converge to the desired solution and capture even a mollified interface to
a satisfying degree.

The last chapter of this thesis investigated the definition of a more complex optimization problem using
a volume fraction description of the interface. The problem we have chosen was to match the movement of
our approximated interface to a desired interface movement, defined using a level set function. The control
variables were the boundary conditions on a velocity potential defining an incompressible and irrotational
velocity field. This very simple example has most of the characteristics we would expect in a the optimization
of a two-phase Navier-Stokes flow, without the difficulties arising from solving the complex one-fluid model.

We have seen how the derivation of the adjoint equations is done in a multidimensional setting and the issues
that we have to consider to provide a well-posed problems.

The most difficult part in defining such an optimization problem is finding a suitable cost functional. In
our case, we have seen that formulating the cost functional using a volume fraction defined by a Heaviside
function is not feasible at a continuous level. This lead us to consider a mollified interface definition even
at the continuous level. Numerically, this is also a necessary choice, because interfaces identified by Dirac
delta functions are hard to deal with discretely. Using a smooth definition of the volume fraction, allowed
us to rigorously define our optimization problem by providing the necessary vector spaces in which our
solution and constraints must reside. Unfortunately, the proposed cost functional is not convex, so proofs of
uniqueness of a optimal control is not possible. We have seen in numerical examples that the control of the
interface described in this way is indeed possible. An interesting result was that, in the case when all the
domain boundaries are controlled, we could easily obtain different local minima of our cost functional.

The results we have seen this far are very promising: we have managed to obtain converging discrete
adjoint solutions in the presence of different types of discontinuities and use these results to control the
movement of a fictitious interface in a very simplified model. Future work is plentiful for problems of
this type. An important direction is finding an improved cost functional that has unique solutions or at
least removes trivial undesirable solutions. Another obvious extension is replacing the velocity defined by
a velocity potential by the Navier-Stokes equations and the complete one-fluid model. Besides applications
related to optimizing the movement of the interface, this will allow us to reproduce classic results regarding
various instabilities and move forward to more complicated and nonlinear problems.
Appendix A

Generalized Tangent Vectors: Analytical First Order Variation

In what follows, we will give a complete derivation of the results from Example 3.2. In that example, we were looking at a variation of the solution of Burgers’ equation corresponding to a variation in the initial condition. The perturbed solution was given by:

\[ u(\epsilon, t, x) = (1 + \epsilon) x + (1 + \epsilon) t \chi_{[0, \sqrt{1+(1+\epsilon)t}]}(x), \]

To find the variation in \( u \), we want to look at the limit of:

\[ \lim_{\epsilon \to 0} \frac{1}{\epsilon} (u(\epsilon, t, x) - u(t, x)). \]

This limit only well-defined on a measure space and has an absolutely continuous part \( \mu \) and a point mass of magnitude \( J \mu \), as given in the example.

To obtain the required result, we will look at the limit in the weak sense. For a smooth test function \( \phi \in C^\infty(\mathbb{R}) \), we have that:

\[
\begin{align*}
\lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{\mathbb{R}} \phi(x) (u(\epsilon, t, x) - u(t, x)) \, dx \\
= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{\mathbb{R}} \phi(x) \left[ \frac{(1 + \epsilon)x}{1 + (1 + \epsilon)t} \chi_{[0, \sqrt{1+(1+\epsilon)t}]}(x) - \frac{x}{1 + t} \chi_{[0, \sqrt{1+t}]}(x) \right] \, dx \\
= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{\mathbb{R}} \phi(x) \left[ \left( \frac{(1 + \epsilon)x}{1 + (1 + \epsilon)t} - \frac{x}{1 + t} \right) \chi_{[0, \sqrt{1+t}]}(x) + \frac{(1 + \epsilon)x}{1 + (1 + \epsilon)t} \chi_{[\sqrt{1+t}, \sqrt{1+(1+\epsilon)t}]}(x) \right] \, dx,
\end{align*}
\]

where the first term can be easily proven to converge to the measure \( \mu(t, x) \) we were given in Example 3.2. The derivation reads:

\[
\begin{align*}
\lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{\mathbb{R}} \phi(x) \left( \frac{(1 + \epsilon)x}{1 + (1 + \epsilon)t} - \frac{x}{1 + t} \right) \chi_{[0, \sqrt{1+t}]} \, dx \\
= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{\mathbb{R}} \phi(x) \left( \frac{(1 + \epsilon)x}{1 + (1 + \epsilon)t} - \frac{x}{1 + t} \right) \chi_{[0, \sqrt{1+t}]} \, dx \\
= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{\mathbb{R}} \phi(x) \left( \frac{(1 + \epsilon)x}{1 + (1 + \epsilon)t} - \frac{x}{1 + t} \right) \chi_{[0, \sqrt{1+t}]} \, dx
\end{align*}
\]
so the limit is given by:

$$\lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{\mathbb{R}} \phi(x) \left( \frac{(1+\epsilon)x}{1 + (1+\epsilon)t} - \frac{x}{1 + t} \right) \chi_{[0, \sqrt{1+t}]} \, dx = \int_{\mathbb{R}} \phi(x) \frac{x}{(1+t)^2} \chi_{[0, \sqrt{1+t}]} \, dx$$

which implies that the solution is:

$$d\mu = \frac{x}{(1+t)^2} \chi_{[0, \sqrt{1+t}]} \, dx.$$

The second term, however, corresponds to point mass at $x = \sqrt{1+t}$ whose magnitude is given by:

$$\|u\| \xi = \frac{t}{2(1+t)},$$

where $\|u\| = u(x(t)_+) - u(x(t)_-)$ is the jump of $u$ across the shock at position $x(t)$. The result follows directly from the exact solution, where:

$$\|u\| = u(\sqrt{1+t}) = \frac{\sqrt{1+t}}{1+t}$$

and $\xi$ is given by:

$$\xi(t) = \lim_{\epsilon \to 0} \frac{x^*(t) - x(t)}{\epsilon}$$

$$= \lim_{\epsilon \to 0} \frac{\sqrt{1 + (1+\epsilon)t} - \sqrt{1+t}}{\epsilon}$$

$$= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \frac{(1 + (1+\epsilon)t) - (1+t)}{\sqrt{1 + (1+\epsilon)t} + \sqrt{1+t}}$$

$$= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \frac{\epsilon t}{\sqrt{1 + (1+\epsilon)t} + \sqrt{1+t}} = \frac{t}{2\sqrt{1+t}}$$

To obtain the desired result, we will use the Reynolds transport theorem on a varying upper bound of the integral (corresponding to the variation of the shock position). The second term in the limit from (A.1) can be written as:

$$\lim_{\epsilon \to 0} \frac{1 + \epsilon}{1 + (1+\epsilon)t} \frac{1}{\epsilon} \left( \int_{0}^{\sqrt{1+(1+\epsilon)t}} x\phi(x) \, dx - \int_{0}^{\sqrt{1+t}} x\phi(x) \, dx \right)$$

$$= \frac{1}{1+t} \xi(t) \sqrt{1+t} \phi(\sqrt{1+t}) = \frac{\sqrt{1+t}}{1+t} \xi \phi(\sqrt{1+t}) = \|u\| \xi \phi(\sqrt{1+t})$$

Finally, we can write the limit using a Dirac delta function:

$$\|u\| \xi \phi(\sqrt{1+t}) = \int_{\mathbb{R}} \phi(x) \|u\| \xi \delta_{\sqrt{1+t}}(x) \, dx,$$

so the limit of the second part of (A.1) is the point mass we have given before at $\sqrt{1+t}$. 

102
Appendix B

Burgers’ Equation: Derivation of Adjoint Equations

This derivation follows the results from [Giles, 2002], hopefully adding some more detail and understanding to the problem. As mentioned, we are looking at the Lagrangian of a scalar conservation law in the presence of a single shock at position \( x_s \):

\[
\mathcal{L}((u, x_s), g, (p, y_s)) = \mathcal{J}(u, g) - \int_{(0,T] \times \Omega} \int_{\Gamma} p \left( \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) \right) \, dx \, dt - \int_{\Omega} p(0, x)(u(0, x) - g) \, dx
\]

\[
- \int_{\Gamma} y_s \left( \dot{x}_s \right) [u] - \| f(u) \| \, dt,
\]

We have already seen the linearized cost functional in (3.20), the linearized equations given in (3.16) and the linearized Rankine-Hugoniot conditions from (3.19). Adding the results together, we obtain the following variation of the Lagrangian:

\[
\frac{\partial \mathcal{L}}{\partial u}(u; v, \xi_s) = \int_{a}^{x_s} G'(u(T, x))v \, dx + \int_{x_s}^{b} G'(u(T, x))v \, dx - \xi_s [G] - \\
\int_{(0,T] \times \Omega} \int_{\Gamma} p \left( \frac{\partial v}{\partial t} + \frac{\partial}{\partial x} (a(u)v) \right) \, dx \, dt - \int_{a}^{b} p(0, x)v(0, x) \, dx - \\
\int_{\Gamma} y_s \left( \dot{x}_s \right) [u] + \dot{x}_s [v + \xi_s u_s] - [a(u)v + \xi_s f(u)] \, ds,
\]

where the variations \( v \) are supposed to have homogeneous boundary conditions \( v(t, a) = v(t, b) = 0 \) and the variations \( \xi_s \) satisfy \( \xi_s(0) = 0 \). First, we will look at the third integral above, containing the linearized state equations, and perform a series of integration by parts to obtain:

\[
- \int_{(0,T] \times \Omega} \int_{\Gamma} p \left( \frac{\partial v}{\partial t} + \frac{\partial}{\partial x} (a(u)v) \right) \, dx \, dt = \int_{(0,T] \times \Omega} \int_{\Gamma} \left( \frac{\partial p}{\partial t} + a(u) \frac{\partial p}{\partial x} \right) v \, dx \, dt - \\
\left[ \int_{a}^{x_s} pv \, dx \right]_{0}^{T} + \left[ \int_{x_s}^{b} pv \, dx \right]_{0}^{T} - \\
\left[ \int_{a}^{T} a(u)pv \, dt \right]_{0}^{x_s} + \left[ \int_{a}^{T} a(u)pv \, dt \right]_{x_s}^{b} - \\
\left[ \int_{0}^{T} \dot{x}_s \left( pv - \frac{pv}{x_s} \right) \right. \]
where we have also used the Reynolds Transport Theorem to account for the moving shock position (which is where the last term comes from). Using the boundary conditions on $v$, the result can be rewritten as:

$$
\int \int_{(0,T] \times \Omega} \left( \frac{\partial p}{\partial t} + a(u) \frac{\partial p}{\partial x} \right) v \, dx \, dt - \left[ \int_a^{\frac{x_s}{T}} pv \, dx \right]_0^T + \int_{x_s}^b \int_0^T \int_\Gamma (\dot{x}_s \| pv \| + \| a(u) pv \|) \, dt.
$$

This leaves the following linearized Lagrangian:

$$
\frac{\partial \mathcal{L}}{\partial u} (u; v, \xi_s) = \int_a^{\frac{x_s}{T}} (G'(u(T)) - p(T)) v(T) \, dx + \int_{x_s}^b (G'(u(T)) - p(T)) v(T) \, dx - \xi_s(T) \| G(T) \| + \int_\Gamma (\dot{x}_s \| v \| + \| a(u) v \| (p + y_s)) \, dt - \int_\Gamma y_s \left( \dot{x}_s \| u \| + \xi_s \| u \| + \xi_s f(u)(x) \right).
$$

Next, we will look at the linearized Rankine-Hugoniot conditions and perform the necessary integrations by parts:

$$
\int_\Gamma y_s \left( \dot{x}_s \| u \| + \xi_s \| u \| + \xi_s f(u)(x) \right) \, dt = \int_\Gamma \left( -(y_s \| u \|)_t + y_s \dot{x}_s \| u \| - y_s \| f(u)(x) \| \right) \, dt + \left[ y_s \| u \| \xi_s \right]_0^T
$$

We also have that:

$$
\frac{d}{dt} \| u \| = \left[ \frac{\partial u}{\partial t} \right] + \dot{x}_s \left[ \frac{\partial u}{\partial x} \right] = - \left[ \frac{\partial f(u)}{\partial x} \right] + \dot{x}_s \left[ \frac{\partial u}{\partial x} \right],
$$

which simplifies the equation to:

$$
\int_\Gamma \left( -(y_s \| u \|)_t + y_s \| u \| \right) \, dt + \left[ y_s \| u \| \xi_s \right]_0^T = - \int_\Gamma \left( \frac{dy_s}{dt} \right) \xi_s \, dt + \left[ y_s \| u \| \xi_s \right]_0^T.
$$

We can use the initial condition on $\xi_s$ to simplify some more terms and get the final directional derivative of the Lagrangian with respect to the state variables $u$:

$$
\frac{\partial \mathcal{L}}{\partial u} (u; v, \xi_s) = \int_a^{\frac{x_s}{T}} (G'(u(T)) - p(T)) v(T) \, dx + \int_{x_s}^b (G'(u(T)) - p(T)) v(T) \, dx + \int_\Gamma (\dot{x}_s \| v \| + \| a(u) v \| (p + y_s)) \, dt + \int_\Gamma \left( \frac{dy_s}{dt} \right) \xi_s - \xi_s(T) \left( \| G(T) \| + y_s(T) \| u(T) \| \right).
$$

Setting the variation to 0 gives the adjoint equations.
Appendix C

THINC: Flux Derivatives

We will define here a complete set of derivatives of the THINC fluxes, as given by (4.7) and (4.8), and of the jump location \( \delta \), given by (4.6). We will perform the differentiation under the assumptions that neither \( \gamma \) nor \( \beta \) (which is the case for the THINC/SW scheme, for example) are functions of \( \alpha \).

For simplicity, we remove the index notation from all formulae in the subsequent derivations and define:

\[
\omega_m = \frac{\beta}{2}(1 - \omega) \quad \text{and} \quad \omega_p = \frac{\beta}{2}(1 + \omega)
\]

so that the jump location formula becomes:

\[
\delta = \frac{1}{2\beta} \log \left( \frac{\sinh \omega_m}{\sinh \omega_p} \right),
\]

By the chain rule, the derivative is:

\[
\frac{\partial \delta}{\partial \alpha} = \frac{1}{2\beta} \left( \frac{\sinh \omega_p}{\sinh \omega_m} \right) \left[ \left( -\frac{2\gamma \beta \cosh \omega_m}{2} \right) - \left( \frac{2\gamma \beta}{2} \coth \omega_p \frac{\sinh \omega_m}{\sinh \omega_p} \right) \right],
\]

which can be further simplified to:

\[
\frac{\partial \delta}{\partial \alpha} = -\frac{\gamma}{2}(\coth \omega_m + \coth \omega_p). \tag{C.1}
\]

Similarly to the jump location itself, its derivative goes to plus or minus infinity as \( \alpha \) gets closer to a pure state, as can be seen in Figure C.1.

We will now look at the flux for \( c > 0 \) given by (4.7) and define:

\[
\begin{align*}
\xi_p^- &= \beta(\delta - 0.5), \\
\xi_m^- &= \beta(\delta - 0.5 + \lambda),
\end{align*}
\]
so that the flux becomes:

\[ f^- = \frac{1}{2} \left[ \lambda + \frac{\gamma}{\beta} \log \left( \frac{\cosh \xi^-_p}{\cosh \xi^-_m} \right) \right]. \]

By the chain rule, the derivative is:

\[ \frac{\partial f^-}{\partial \alpha} = \frac{\gamma}{2\beta} \left( \frac{\cosh \xi^-_m}{\cosh \xi^-_p} \right) \left[ \beta \frac{\partial \delta}{\partial \alpha} \frac{\sinh \xi^-_p}{\cosh \xi^-_m} - \beta \frac{\partial \delta}{\partial \alpha} \frac{\tanh \xi^-_m}{\cosh \xi^-_m} \right], \]

which can be further simplified to:

\[ \frac{\partial f^-}{\partial \alpha} = \frac{\gamma}{2} \frac{\partial \delta}{\partial \alpha} (\tanh \xi^-_p - \tanh \xi^-_m) \tag{C.2} \]

Similarly, we have the derivative of the flux for \( c \leq 0 \) given by (4.8):

\[ \frac{\partial f^+}{\partial \alpha} = \frac{\gamma}{2} \frac{\partial \delta}{\partial \alpha} (\tanh \xi^+_p - \tanh \xi^+_m), \tag{C.3} \]

where:

\[
\left\{
\begin{align*}
\xi^+_p &= \beta(\delta + 0.5 + \lambda), \\
\xi^+_m &= \beta(\delta + 0.5).
\end{align*}
\right.
\]
Appendix D

THINC: First-order Approximations

Volume Fraction Approximations

In this section we will provide first-order approximations for the THINC flux at pure states. Using a Taylor expansion to first order, the THINC flux (for $c > 0$) is:

\[
\begin{align*}
  f^- (\alpha) &= f^- (0) + \alpha \frac{\partial f^-}{\partial \alpha} (0) + O(\alpha^2), \\
  f^- (\alpha) &= f^- (1) + (\alpha - 1) \frac{\partial f^-}{\partial \alpha} (1) + O(\alpha^2),
\end{align*}
\]

using the same simplified notation from Appendix C. We will continue using the flux for $c > 0$ with the understanding that the opposite case is completely analogous.

The problematic term in the flux definition is the jump location and its derivative from (C.1), which go to infinity in the limits of pure states. Namely (see Figure 4.1 for $\gamma = -1$):

\[
\lim_{\alpha \to 0^+} \delta = \begin{cases} 
  -\infty, & \gamma = -1, \\
  \infty, & \gamma = 1,
\end{cases}
\quad\text{and}\quad...
\]

and the derivatives (see Figure C.1 for $\gamma = 1$):

\[
\lim_{\alpha \to 0^+} \frac{\partial \delta}{\partial \alpha} = \lim_{\alpha \to 1^-} \frac{\partial \delta}{\partial \alpha} = \begin{cases} 
  \infty, & \gamma = -1, \\
  -\infty, & \gamma = 1.
\end{cases}
\]

Using the available information, we will compute the limits as $\alpha \to 0$ and $\alpha \to 1$ of the terms in the first-order expansion (D.1). The limit of the flux at 0 is given by:

\[
\lim_{\alpha \to 0^+} f^+ = \lim_{\alpha \to 0^+} \frac{1}{2} \left( \lambda + \frac{\gamma}{\beta} \log \left( \frac{\cosh \xi_p}{\cosh \xi_m} \right) \right),
\]

where the term of interest is the argument of the logarithm, which contains $\delta$ going to $\pm \infty$. For $\gamma = -1$, we
have that:

\[
\lim_{\alpha \to 0^+} \frac{\cosh \beta (\delta - 0.5)}{\cosh \beta (\delta - 0.5 + \lambda)} = \lim_{\alpha \to 0^+} \frac{2 \exp \beta (\delta - 0.5 + \lambda)}{2 \exp(2 \beta (\delta - 0.5))} \frac{\exp(2 \beta (\delta - 0.5 + \lambda)) + 1}{\exp(2 \beta (\delta - 0.5)) + 1}
\]

\[= \exp(\beta \lambda),\]

since the second term goes to 1 for \(\delta \to -\infty\). Plugging this result into the limit of the flux, we can remove the logarithm. Finally, we have the following results as \(\alpha \to 0^+\):

\[
\lim_{\alpha \to 0^+} f^+ = 0 \quad \text{and} \quad \lim_{\alpha \to 0^+} f^- = 0
\]  

(D.2)

and as \(\alpha \to 1^-\):

\[
\lim_{\alpha \to 1^-} f^+ = \lambda \quad \text{and} \quad \lim_{\alpha \to 1^-} f^- = \lambda.
\]

(D.3)

The results from (D.2) and (D.3) are what we expected all along, having seen that the flux obtained for the THINC scheme does indeed tend to the upwind flux. This justifies the use of the UPWIND flux for small or large \(\alpha\). Next we will look at the flux derivative (see derivation in Appendix C):

\[
\lim_{\alpha \to 0^+} \frac{\partial f^+}{\partial \alpha} = \lim_{\alpha \to 0^+} \frac{\gamma}{2} \frac{\partial \delta}{\partial \alpha} \left[ \tanh \xi_p - \tanh \xi_m \right],
\]

(D.4)

\[= \lim_{\alpha \to 0^+} -\frac{1}{4} \left[ \coth \omega_m + \coth \omega_p \right] \left[ \tanh \xi_p - \tanh \xi_m \right].\]

We will now fix \(\gamma = 1\), so that we can know which of the two coth in the first term will got to \(\infty\). Notice that the hyperbolic tangents will both tend to the same limit, so their difference will tend to 0, which means that which ever of the two coth does not explode can be removed. In this case, we have that

\[\coth \omega_p = \coth(\beta \alpha) \xrightarrow{\alpha \to 0^+} \infty;\]

so we are left with:

\[
\lim_{\alpha \to 0^+} \frac{\partial f^+}{\partial \alpha} = \lim_{\alpha \to 0^+} -\frac{1}{4} \coth(\beta \alpha) \left[ \tanh \xi_p - \tanh \xi_m \right].
\]

We will now look at each term individually in terms of exponentials. We have that:

\[\coth(\beta \alpha) = \frac{\exp(2 \beta \alpha) + 1}{\exp(2 \beta \alpha) - 1}.\]
Next we will look at the hyperbolic tangents:
\[
\tanh \xi_p - \tanh \xi_m = \tanh \beta (\delta - 0.5) + \tanh -\beta (\delta - 0.5 + \lambda) \\
= \frac{\exp(2\beta (\delta - 0.5)) - 1}{\exp(2\beta (\delta - 0.5)) + 1} + \frac{\exp(-2\beta (\delta - 0.5 + \lambda)) - 1}{\exp(-2\beta (\delta - 0.5 + \lambda)) + 1} \\
= \frac{2[\exp(-2\beta \lambda) - 1]}{[\exp(2\beta (\delta - 0.5)) + 1][\exp(-2\beta (\delta - 0.5 + \lambda)) + 1]}.
\]

For \( \gamma = 1 \), we know that \( \delta \to \infty \) as \( \alpha \to 0^+ \), so the second term in the denominator will go to 1 in the limit. Putting the two terms together and taking the limit of any finite terms, we are left with:
\[
\lim_{\alpha \to 0^+} \frac{\partial f^+}{\partial \alpha} = \lim_{\alpha \to 0^+} - \frac{\exp(-2\beta \lambda) - 1}{[\exp(2\beta \alpha) - 1][\exp(2\beta (\delta - 0.5)) + 1]}.
\]
Before going further, we notice that:
\[
\exp(2\beta (\delta - 0.5)) = \exp(-\beta) \frac{\sinh(\beta(1 - \alpha))}{\sinh(\beta \alpha)}
\]
and that:
\[
\exp(2\beta \alpha) - 1 = 2 \exp(\beta \alpha) \sinh(\beta \alpha).
\]
This simplifies the limit to:
\[
\lim_{\alpha \to 0^+} \frac{\partial f^+}{\partial \alpha} = \exp(\beta(1 - \lambda)) \frac{\sinh(\beta \lambda)}{\sinh(\beta)}.
\]
Reintroducing the \( \gamma \) dependence, we get:
\[
\lim_{\alpha \to 0^+} \frac{\partial f^+}{\partial \alpha} = \exp(\gamma \beta (1 - \lambda)) \frac{\sinh(\beta \lambda)}{\sinh(\beta)},
\]
\[
\lim_{\alpha \to 0^+} \frac{\partial f^-}{\partial \alpha} = \exp(-\gamma \beta (1 + \lambda)) \frac{\sinh(\beta \lambda)}{\sinh(\beta)},
\]
and similarly we can obtain:
\[
\lim_{\alpha \to 1^-} \frac{\partial f^+}{\partial \alpha} = \exp(-\gamma \beta (1 - \lambda)) \frac{\sinh(\beta \lambda)}{\sinh(\beta)},
\]
\[
\lim_{\alpha \to 1^-} \frac{\partial f^-}{\partial \alpha} = \exp(\gamma \beta (1 + \lambda)) \frac{\sinh(\beta \lambda)}{\sinh(\beta)}.
\]
An example of this approximation can be seen in Figure D.1. From the plots we can see that \( \epsilon \) should take values smaller than \( \approx 0.05 \) if we desire the jump to be minimal.
Figure D.1: First order approximation (dashed) for of the THINC flux as $\alpha \to 0$ (left) and as $\alpha \to 1$ (right) for $c > 0$ and $\gamma = 1$.

**Slope Steepness Approximations**

Another interesting problem can arise when $\beta \to 0$. Similarly to the case where $\alpha$ goes to one of the pure states, the jump location is no longer well defined in the case of infinite interface thickness. This can be seen in Figure D.2.

Figure D.2: Jump location $\delta$ as a function of $\beta$ for various values of the volume fraction $\alpha$.

Of course, this scenario is not of interest in the case of the classic THINC scheme, but can arise very easily in the case of the THINC/SW scheme where the slope steepness is defined on a per-cell and per-dimension
basis by:

\[
\beta_{ijkl,x}^m = \frac{|n_{ijkl,x}^m|}{\eta} \tanh^{-1}(1 - 2\epsilon),
\]

where \( n_{ijkl,x}^m \) is the interface normal in the current cell. Using this definition, the case of \( \beta \to 0 \) can happen if the interface is completely aligned with the mesh. We will now proceed to look at the jump location \( \delta \) as a function of \( \beta \):

\[
\delta(\beta) = \frac{1}{2\beta} \log \left( \frac{\sinh(\beta \omega_m)}{\sinh(\beta \omega_p)} \right),
\]

using the definitions of \( \omega_m, \omega_p \) from Appendix C. As we have seen, \( \delta \to \pm \infty \) as \( \beta \to 0^+ \):

\[
\lim_{\beta \to 0^+} \delta(\beta) = \begin{cases} \infty, & \omega_m > \omega_p, \\ 0, & \omega_m = \omega_p, \\ -\infty, & \omega_m < \omega_p. \end{cases}
\]

The derivative of the jump location with respect to \( \beta \) is given by:

\[
\frac{\partial \delta}{\partial \beta} = -\frac{1}{2\beta^2} \log \left( \frac{\sinh(\beta \omega_m)}{\sinh(\beta \omega_p)} \right) + \frac{1}{2 \beta} \left[ \omega_m \coth(\omega_m \beta) - \omega_p \coth(\omega_p \beta) \right],
\]

which similarly goes to infinity. As before, we cannot do a first-order expansion for \( \delta \) since it is not differentiable at 0, but we can still perform one for the flux, which is well behaved. A first-order Taylor expansion of the flux is given by (for \( c > 0 \)):

\[
f^{-}(\beta) = f^{-}(0) + \beta \frac{\partial f^{-}}{\partial \beta}(0).
\]

Before going forward, we will compute a couple of smaller limits to aid us. First, we have:

\[
\lim_{\beta \to 0^+} \beta \delta(\beta) = \lim_{\beta \to 0^+} \frac{1}{2} \log \left( \frac{\sinh(\omega_m \beta)}{\sinh(\omega_p \beta)} \right) = \frac{1}{2} \log \left( \frac{\omega_m}{\omega_p} \right).
\]

Using this, we also have:

\[
\lim_{\beta \to 0^+} \tanh(\beta(\delta - 0.5)) = \lim_{\beta \to 0^+} \tanh(\beta(\delta - 0.5 + \lambda)) = \tanh \left[ \frac{1}{2} \log \left( \frac{\omega_m}{\omega_p} \right) \right]. \tag{D.5}
\]

Finally, we look at the limit of the flux function as \( \beta \to 0^+ \) using the previous results and the l’Hôpital
rule:
\[
\lim_{\beta \to 0^+} \frac{1}{2} \left[ \lambda + \frac{\gamma}{\beta} \log \left( \frac{\cosh(\beta(\delta - 0.5))}{\cosh(\beta(\delta - 0.5 + \lambda))} \right) \right],
\]
\[
= \frac{1}{2} \lambda + \lim_{\beta \to 0^+} \frac{\gamma}{2} \left[ \tanh(\beta(\delta - 0.5)) \left\{ (\delta - 0.5 + \beta \delta'(\beta)) - \tanh(\beta(\delta - 0.5 + \lambda)) \right\} \right] - \tanh(\beta(\delta - 0.5 + \lambda)) \left\{ (\delta - 0.5 + \lambda) + \beta \delta'(\beta) \right\}
\]
\[
= \frac{1}{2} \lambda - \lim_{\beta \to 0^+} \frac{\gamma}{2} \lambda \tanh(\beta(\delta - 0.5 + \lambda)) + \lim_{\beta \to 0^+} \frac{\gamma}{2} \{ \delta - 0.5 + \beta \delta' \} (\tanh(\beta(\delta - 0.5)) - \tanh(\beta(\delta - 0.5 + \lambda))).
\]

The only term of interest in the sum above is the last limit which is essentially a product of $\infty \times 0$ at $0^+$. We have already dealt with a similar case when computing the limit of the flux derivative with respect to $\alpha$ at $0$ in (D.4). The treatment can be done almost identically and we notice that the limit is $0$. Therefore, we are left with:
\[
\lim_{\beta \to 0^+} f_-(\beta) = \frac{1}{2} \lambda \left( 1 - \gamma \tanh \left[ \frac{1}{2} \log \left( \frac{\omega_m}{\omega_p} \right) \right] \right).
\]

The limit in the case of $c \leq 0$ is exactly the same. The limits of the flux derivatives can also be derived in a similar manner. They are given by:
\[
\left\{ \begin{array}{l}
\lim_{\beta \to 0^+} \frac{\partial f_+}{\partial \beta} = -\frac{1}{4} \gamma \lambda (\lambda - 1) \sech^2 \left[ \frac{1}{2} \log \left( \frac{\omega_m}{\omega_p} \right) \right],
\lim_{\beta \to 0^+} \frac{\partial f_-}{\partial \beta} = -\frac{1}{4} \gamma \lambda (\lambda + 1) \sech^2 \left[ \frac{1}{2} \log \left( \frac{\omega_m}{\omega_p} \right) \right].
\end{array} \right.
\]

Figure D.3: First-order approximation of the flux (dashed) for $c \leq 0$ (left) and $c > 0$ (right).
Appendix E

COMIC: Fluxes and Flux Derivatives

Following the work done in Chapter 4 on separating the primitive of the COMIC mollifier into different cases to obtain simpler problems to solve, we give here the fluxes and flux derivatives in each of the cases outlined there.

Case 0: $\delta + \eta < -0.5$ or $\delta - \eta > 0.5$

This is the case where $\alpha \approx 0$ or $\alpha \approx 1$. In this case, we simply set $\delta = \pm \gamma(\delta + \eta)$ depending on the state. Here, $\delta$ has no real dependence on $\alpha$ and we get:

$$\frac{\partial \delta}{\partial \alpha} = 0.$$

The flux and its derivatives are given by:

- if $c > 0$ and $\delta + \eta < -0.5$:
  $$\begin{cases} f^- = b \lambda, \\ \frac{\partial f^-}{\partial \alpha} = 0. \end{cases}$$

- if $c > 0$ and $\delta - \eta > 0.5$:
  $$\begin{cases} f^- = a \lambda, \\ \frac{\partial f^-}{\partial \alpha} = 1. \end{cases}$$

- if $c \leq 0$ and $\delta + \eta < -0.5$:
  $$\begin{cases} f^+ = b \lambda, \\ \frac{\partial f^+}{\partial \alpha} = -1. \end{cases}$$

- if $c \leq 0$ and $\delta - \eta > 0.5$:
  $$\begin{cases} f^+ = a \lambda, \\ \frac{\partial f^+}{\partial \alpha} = 0. \end{cases}$$
Note that the flux and flux derivatives given here are the limits of the results from case 3 or case 4.

**Case 1:** $\delta - \eta > -0.5$ and $\delta + \eta < 0.5$

In this case, we have that:

$$\delta = \frac{1}{a - b} \left( \alpha - \frac{1}{2} \right),$$

so

$$\frac{\partial \delta}{\partial \alpha} = \frac{1}{a - b}.$$

The flux and flux derivatives are given by:

- if $c > 0$ and $\delta - \eta > 0.5 - \lambda$.

$$\begin{aligned}
&\{ f^- = (b - a) \left( \frac{1}{2} - \delta - \eta \right) + (1 + 2a)\eta + a\lambda, \\
&\frac{\partial f^-}{\partial \alpha} = (a - b) \frac{\partial \delta}{\partial \alpha}. \\
\end{aligned}$$

- if $c > 0$ and $\delta - \eta < 0.5 - \lambda$ (same as Case 3.2, but $\delta - \eta$ is not smaller than $-0.5$ too).

$$\begin{aligned}
&\{ f^- = \left( b - \frac{1}{2} \right) \left( \frac{1}{2} - \delta - \eta \right) + \frac{1}{2} \left( \lambda + \gamma \frac{2\eta}{\pi} \cos \left( \frac{\pi}{2\eta} (\delta - 0.5 + \lambda) \right) \right), \\
&\frac{\partial f^-}{\partial \alpha} = \frac{\partial \delta}{\partial \alpha} \left[ \left( \frac{1}{2} - b \right) - \gamma \frac{2}{\pi} \sin \left( \frac{\pi}{2\eta} (\delta - 0.5 + \lambda) \right) \right]. \\
\end{aligned}$$

- $c \leq 0$ and $\delta + \eta < -0.5 - \lambda$.

$$\begin{aligned}
&\{ f^+ = (b - a) \left( \frac{1}{2} + \delta - \eta \right) - (1 - 2a)\eta + a\lambda, \\
&\frac{\partial f^+}{\partial \alpha} = (a - b) \frac{\partial \delta}{\partial \alpha}. \\
\end{aligned}$$

- $c \leq 0$ and $\delta + \eta > -0.5 - \lambda$ (same as Case 4.3, but $\delta + \eta$ is not larger than 0.5 too).

$$\begin{aligned}
&\{ f^+ = \left( \frac{1}{2} - a \right) \left( \delta - \eta + \frac{1}{2} \right) + \frac{1}{2} \left( \lambda + \gamma \frac{2\eta}{\pi} \cos \left( \frac{\pi}{2\eta} (\delta + 0.5 + \lambda) \right) \right), \\
&\frac{\partial f^+}{\partial \alpha} = \frac{\partial \delta}{\partial \alpha} \left[ \left( \frac{1}{2} - a \right) - \gamma \frac{2}{\pi} \sin \left( \frac{\pi}{2\eta} (\delta + 0.5 + \lambda) \right) \right]. \\
\end{aligned}$$
**Case 2: \( \delta - \eta < -0.5 \) and \( \delta + \eta > 0.5 \)**

In this case, \( \delta \) is given by:

\[
\delta = -\frac{2\eta}{\pi} \arcsin \left( \frac{\pi \gamma}{4\eta} \sin \left( \frac{\pi}{4\eta} \right) \right)^{-1} (2\alpha - 1),
\]

so:

\[
\frac{\partial \delta}{\partial \alpha} = -\frac{\gamma}{\sqrt{1 - z^2}} \left( \sin \left( \frac{\pi}{4\eta} \right) \right)^{-1},
\]

where \( z \) is the argument of the arcsin.

For the flux and flux derivatives, we have:

- if \( c > 0 \)

\[
\begin{align*}
    f^- &= \frac{1}{2} \left( \lambda - \frac{2\eta}{\pi} \left[ \cos \left( \frac{\pi}{2\eta} (\delta - 0.5) \right) - \cos \left( \frac{\pi}{2\eta} (\delta - 0.5 + \lambda) \right) \right] \right), \\
    \frac{\partial f^-}{\partial \alpha} &= \frac{\gamma}{2} \frac{\partial \delta}{\partial \alpha} \left[ \sin \left( \frac{\pi}{2\eta} (\delta - 0.5) \right) - \sin \left( \frac{\pi}{2\eta} (\delta - 0.5 + \lambda) \right) \right].
\end{align*}
\]

- if \( c \leq 0 \)

\[
\begin{align*}
    f^+ &= \frac{1}{2} \left( \lambda + \frac{2\eta}{\pi} \left[ \cos \left( \frac{\pi}{2\eta} (\delta + 0.5 + \lambda) \right) - \cos \left( \frac{\pi}{2\eta} (\delta + 0.5) \right) \right] \right), \\
    \frac{\partial f^+}{\partial \alpha} &= -\frac{\gamma}{2} \frac{\partial \delta}{\partial \alpha} \left[ \sin \left( \frac{\pi}{2\eta} (\delta + 0.5 + \lambda) \right) - \sin \left( \frac{\pi}{2\eta} (\delta + 0.5) \right) \right].
\end{align*}
\]

**Case 3: \( \delta - \eta < -0.5 \) and \( \delta + \eta < 0.5 \)**

In this case, we have to use the implicit function theorem and chain rule on:

\[
\Psi(\delta) = \alpha \implies \frac{\partial \Psi}{\partial \delta} \frac{\partial \delta}{\partial \alpha} = 1 \implies \frac{\partial \delta}{\partial \alpha} = \left( \frac{\partial \Psi}{\partial \delta} \right)^{-1}.
\]

We have that:

\[
\frac{\partial \Psi}{\partial \delta} = -\frac{\gamma}{2} \left( 1 + \sin \left( \frac{\pi}{2\eta} (\delta + 0.5) \right) \right),
\]

so:

\[
\frac{\partial \delta}{\partial \alpha} = \frac{-2\gamma}{1 + 2 \sin \left( \frac{\pi}{2\eta} (\delta + 0.5) \right)}.
\]

The flux and flux derivatives are given by:

- if \( c > 0 \) and \( \delta + \eta \leq 0.5 - \lambda \)

\[
\begin{align*}
    f^- &= b\lambda, \\
    \frac{\partial f^-}{\partial \alpha} &= 0.
\end{align*}
\]
• if $c > 0$ and $\delta + \eta > 0.5 - \lambda$.

\[
\begin{align*}
  f^- &= \left(b - \frac{1}{2}\right) \left(\frac{1}{2} - \delta - \eta\right) + \frac{1}{2} \left(\lambda + \gamma \frac{2\eta}{\pi} \cos\left(\frac{\pi}{2\eta}(\delta - 0.5 + \lambda)\right)\right), \\
  \frac{\partial f^-}{\partial \alpha} &= \frac{\partial \delta}{\partial \alpha} \left[\left(\frac{1}{2} - b\right) - \gamma \frac{2}{\sin\left(\frac{\pi}{2\eta}(\delta - 0.5 + \lambda)\right)}\right].
\end{align*}
\]

• if $c \leq 0$ and $\delta + \eta < -0.5 - \lambda$.

\[
\begin{align*}
  f^+ &= \left(b - \frac{1}{2}\right) \left(\frac{1}{2} + \delta + \eta\right) + \frac{1}{2} \left(2b\lambda - \gamma \frac{2\eta}{\pi} \cos\left(\frac{\pi}{2\eta}(\delta + 0.5)\right)\right), \\
  \frac{\partial f^+}{\partial \alpha} &= \frac{\partial \delta}{\partial \alpha} \left[\left(\frac{1}{2} - b\right) + \gamma \frac{2}{\sin\left(\frac{\pi}{2\eta}(\delta + 0.5)\right)}\right].
\end{align*}
\]

• if $c \leq 0$ and $\delta + \eta > -0.5 - \lambda$.

\[
\begin{align*}
  f^+ &= \frac{1}{2} \left(\lambda + \gamma \frac{2\eta}{\pi}\right) \left[\cos\left(\frac{\pi}{2\eta}(\delta - 0.5 + \lambda)\right) - \cos\left(\frac{\pi}{2\eta}(\delta + 0.5)\right)\right], \\
  \frac{\partial f^+}{\partial \alpha} &= \frac{\gamma \partial \delta}{\partial \alpha} \left(\sin\left(\frac{\pi}{2\eta}(\delta + 0.5 + \lambda)\right) - \sin\left(\frac{\pi}{2\eta}(\delta + 0.5)\right)\right).
\end{align*}
\]

**Case 4:** $\delta - \eta > -0.5$ and $\delta + \eta > 0.5$

Similarly to the previous case, we can obtain:

\[
\frac{\partial \delta}{\partial \alpha} = \frac{-2\gamma}{1 - \sin\left(\frac{\pi}{2\eta}(\delta - 0.5)\right)}.
\]

The flux and flux derivatives are given by:

• if $c > 0$ and $\delta - \eta \leq 0.5 - \lambda$.

\[
\begin{align*}
  f^- &= \frac{1}{2} \left(\lambda - \gamma \frac{2\eta}{\pi}\right) \left[\cos\left(\frac{\pi}{2\eta}(\delta - 0.5 + \lambda)\right) - \cos\left(\frac{\pi}{2\eta}(\delta - 0.5)\right)\right], \\
  \frac{\partial f^-}{\partial \alpha} &= \frac{\gamma \partial \delta}{\partial \alpha} \left(\sin\left(\frac{\pi}{2\eta}(\delta - 0.5)\right) - \sin\left(\frac{\pi}{2\eta}(\delta - 0.5 + \lambda)\right)\right).
\end{align*}
\]

• if $c > 0$ and $\delta - \eta > 0.5 - \lambda$.

\[
\begin{align*}
  f^- &= \left(a - \frac{1}{2}\right) \left(\delta - \eta - \frac{1}{2}\right) + \frac{1}{2} \left(2a\lambda - \gamma \frac{2\eta}{\pi} \cos\left(\frac{\pi}{2\eta}(\delta - 0.5)\right)\right), \\
  \frac{\partial f^-}{\partial \alpha} &= \frac{\partial \delta}{\partial \alpha} \left[\left(a - \frac{1}{2}\right) + \gamma \frac{2}{\sin\left(\frac{\pi}{2\eta}(\delta - 0.5)\right)}\right].
\end{align*}
\]

116
• if $c \leq 0$ and $\delta - \eta < -0.5 - \lambda$.

\[
\begin{align*}
 f^+ &= \left( \frac{1}{2} - a \right) \left( \delta - \eta + \frac{1}{2} \right) + \frac{1}{2} \left( \lambda + \gamma \frac{2\eta}{\pi} \cos \left( \frac{\pi}{2\eta} (\delta + 0.5 + \lambda) \right) \right), \\
 \frac{\partial f^+}{\partial \alpha} &= \left( \frac{1}{2} - a \right) - \frac{\gamma}{2} \sin \left( \frac{\pi}{2\eta} (\delta + 0.5 + \lambda) \right).
\end{align*}
\]

• if $c \leq 0$ and $\delta - \eta > -0.5 - \lambda$.

\[
\begin{align*}
 f^+ &= a \lambda, \\
 \frac{\partial f^+}{\partial \alpha} &= 0.
\end{align*}
\]
Appendix F

Motion Planning: Derivation of the Continuous Adjoint

Here we will derive the continuous adjoint equations stated in Chapter 5. The constraints are given by an advection equation (5.1) and a Laplace equation (5.2) with a cost functional given by (5.3). The resulting Lagrangian then reads (5.7):

\[ L = J - \int_0^T \int_\Omega \alpha^* \left( \alpha_t + \nabla \phi \cdot \nabla \alpha \right) \, dx \, dt - \int_\Omega \alpha_0^* \left( \alpha(0, x) - \alpha_0(x) \right) \, dx - \int_0^T \int_\Gamma \alpha_D^* \left( \alpha - \alpha_D \right) \, dx \, dt - \int_0^T \int_\Omega \varphi^* \Delta \varphi \, dx - \int_\Gamma \varphi^*_D \left( \varphi - g \right) \, dx - \int_\Gamma \varphi^*_N \left( \nabla \varphi \cdot n - h \right) \, dx, \]

Note that the Lagrangian above is largely symbolic, since we have not rigorously defined the functional spaces of any of the variables. A discussion on this topic is given in Chapter 5 and here we are merely interested in deriving the adjoint equations, for which we will simply assume that the functions are sufficiently smooth.

Adjoint Volume Fraction

First, we will differentiate the cost functional (5.3) with respect to the volume fraction \( \alpha \). We have, for all admissible variations \( \dot{\alpha} \):

\[ \frac{\partial J}{\partial \alpha} (\alpha; \dot{\alpha}) = \gamma_1 \int_0^T \int_\Omega |\phi_d|^2 \alpha (1 - \alpha)(1 - 2\alpha) \dot{\alpha} \, dx \, dt + \gamma_2 \int_\Omega |\phi_d(T)|^2 \alpha(T)(1 - \alpha(T))(1 - 2\alpha(T)) \dot{\alpha}(T) \, dx. \]

The derivative of the Lagrangian is then given by:

\[ \frac{\partial L}{\partial \alpha} (\alpha; \dot{\alpha}) = \frac{\partial J}{\partial \alpha} (\alpha; \dot{\alpha}) - \int_0^T \int_\Omega \alpha^* \left( \dot{\alpha}_t + \nabla \phi \cdot \nabla \dot{\alpha} \right) \, dx \, dt - \int_\Omega \alpha_0^* \dot{\alpha} \, dx - \int_0^T \int_\Gamma \alpha_D^* \dot{\alpha} \, dx \, dt \]
For the first integral, we need to do two integration by parts, in time and space. The time derivative becomes:

\[
\int_{T_0}^{T} \int_{\Omega} \alpha^* \alpha \cdot \partial_t \bar{\alpha} \, dx \, dt = \int_{\Omega} \left( \alpha^*(T) \bar{\alpha}(T) - \alpha^*(0) \bar{\alpha}(0) \right) \, dx - \int_{T_0}^{T} \int_{\Omega} \alpha^*_0 \bar{\alpha} \, dx \, dt.
\]

For the space derivative, we get:

\[
\int_{T_0}^{T} \int_{\Omega} \alpha^* \nabla \varphi \cdot \nabla \bar{\alpha} \, dx \, dt = \int_{\Omega} \int_{\partial \Omega} \left( \alpha^* (\nabla \varphi \cdot n) \right) \bar{\alpha} \, dx \, dt - \int_{T_0}^{T} \int_{\Omega} \left( \nabla \cdot (\alpha^* \nabla \varphi) \right) \bar{\alpha} \, dx \, dt.
\]

Gathering the results back into the Lagrangian derivative, we get that:

\[
\frac{\partial \mathcal{L}}{\partial \alpha}(\alpha; \bar{\alpha}) = \int_{T_0}^{T} \int_{\Omega} \left( \gamma_1 |\phi_d|^2 \alpha (1 - \alpha) (1 - 2\alpha) + \alpha^*_0 + \nabla \cdot (\alpha^* \nabla \varphi) \right) \bar{\alpha} \, dx \, dt + \int_{\Omega} \int_{\partial \Omega} \left( \gamma_2 |\phi_d(t)|^2 \alpha (1 - \alpha)(1 - 2\alpha) - \alpha^*(t) \right) \bar{\alpha} \, dx - \int_{T_0}^{T} \int_{\Omega} \left( \alpha^*_0 - \alpha^*(0) \right) \bar{\alpha} (0) \, dx - \int_{T_0}^{T} \int_{\Gamma} \left( \alpha^* (\nabla \varphi \cdot n) \right) \bar{\alpha} \, dx - \int_{T_0}^{T} \int_{\partial \Omega} \alpha^*_D \bar{\alpha} \, dx = 0.
\]

Systematically picking \( \bar{\alpha} \) to remove certain terms from the derivatives, we get the following necessary conditions in the form of the adjoint volume fraction equation:

\[
\begin{align*}
-\alpha^*_t & - \nabla \cdot (\alpha^* \nabla \varphi) = \gamma_1 S(\alpha), \quad (t, x) \in [0, T] \times \Omega, \\
\alpha^*(T) & = \gamma_2 S(\alpha(T)), \quad x \in \Omega, \\
\alpha^*(0) & = \alpha^*_0, \quad x \in \Omega \\
\alpha^* (\nabla \varphi \cdot n) & = -\alpha^*_D, \quad (t, x) \in [0, T] \times \Gamma, \\
\alpha^*(\nabla \varphi \cdot n) & = 0, \quad (t, x) \in [0, T] \times \partial \Omega \setminus \Gamma,
\end{align*}
\]

where the source term comes from the derivative of the cost functional and is given by:

\[
S(\alpha) = |\phi_d|^2 \alpha (1 - \alpha)(1 - 2\alpha).
\]

We are still left with two undefined adjoint variables, namely \( \alpha^*_0 \) and \( \alpha^*_D \). The solution at time \( t = 0 \) for \( \alpha^* \) is already known from solving the PDE, so we can set:

\[
\alpha^*_0(x) = \alpha^*(0, x), \quad x \in \Omega.
\]

In the case of the boundary conditions, the problem is slightly more subtle. We can see in Figure F.1...
a simple representation of the characteristics of the state equation solved forward in time and the adjoint equation solved backward in time, for the same constant velocity field. Indeed, the inflow boundaries of the state equations become the outflow boundaries of the adjoint, when solved backward in time, and, similarly, the outflow boundaries of the state equations become inflow boundaries for the adjoint. This leads us to believe that the adjoint only requires boundary conditions on \( \partial \Omega \setminus \Gamma \). Then we can simply set:

\[
\alpha^*_D(t, x) = -\alpha^*(t, x)(\nabla \varphi \cdot n), \quad (t, x) \in [0, T] \times \Gamma,
\]

since the outflowing values of \( \alpha^* \) and the velocity potential are known.

![Figure F.1: Characteristics of the state equation (left) and the adjoint equation (right) for a constant velocity.](image)

**Adjoint Velocity Potential**

Next, we will differentiate the Lagrangian with respect to the velocity potential \( \varphi \) in an admissible direction \( \tilde{\varphi} \), which gives:

\[
\frac{\partial L}{\partial \varphi}(\varphi; \tilde{\varphi}) = -\int_0^T \int_\Omega \alpha^*[\nabla \tilde{\varphi} \cdot \nabla \alpha] \, dx \, dt - \int_0^T \int_{\partial \Omega} (\nabla \tilde{\varphi} \cdot n) \alpha^*_D(\alpha - \alpha_D) \, dx \, dt \\
- \int_\Omega \varphi^* \Delta \tilde{\varphi} \, dx - \int_{\Gamma_D} \varphi^*_D \tilde{\varphi} \, dx - \int_{\Gamma_N} \varphi^*_N \nabla \tilde{\varphi} \cdot n \, dx.
\]

Note that the inflowing boundary \( \Gamma \) of the state advection equation also depends on the velocity potential \( \varphi \), so we must use the Reynolds transport theorem to differentiate the boundary term. We will now look at the first integral. We have:

\[
\int_0^T \int_\Omega \alpha^*[\nabla \alpha \cdot \nabla \tilde{\varphi}] \, dx \, dt = \int_{\partial \Omega} \left( \int_0^T (\alpha^* \nabla \alpha) \cdot n \, dt \right) \tilde{\varphi} \, dx - \int_\Omega \left( \int_0^T \nabla \cdot (\alpha^* \nabla \alpha) \, dt \right) \tilde{\varphi} \, dx.
\]
For the second integral, we need to apply integration by parts twice, to get:

\[ \int_\Omega \varphi^* \Delta \tilde{\varphi} = \int_{\partial\Omega} \varphi^* (\nabla \tilde{\varphi} \cdot \mathbf{n}) \, d\mathbf{x} - \int_\Omega \left( \int_0^T (\alpha^* \nabla \alpha - \Delta \varphi^*) \, d\mathbf{x} \right) \frac{\partial}{\partial t} (\int_0^T (\alpha_D^* (\alpha_D - \alpha_D) \, dt) \right) \nabla \tilde{\varphi} \cdot \mathbf{n} \, d\mathbf{x} + \int_\Omega (\Delta \varphi^*) \tilde{\varphi} \, d\mathbf{x}. \]

We can now get back to our derivative and expand all the terms:

\[ \frac{\partial L}{\partial \varphi} (\varphi^*; \tilde{\varphi}) = \int_\Omega \left( \int_0^T (\nabla (\alpha^* \nabla \alpha) \, d\mathbf{x} - \int_\Omega (\int_0^T (\alpha_D^* (\alpha_D - \alpha_D) \, dt) \right) \nabla \tilde{\varphi} \cdot \mathbf{n} \, d\mathbf{x} + \int_\Gamma_D \left( \int_0^T (\alpha_D^* \nabla \alpha \cdot \mathbf{n} \, d\mathbf{x} + \int_\Gamma_D (\alpha_D^* \nabla \alpha \cdot \mathbf{n} \, d\mathbf{x} = 0, \]

We get the following adjoint velocity potential equation:

\[
\begin{cases}
\Delta \varphi^* = \int_0^T \nabla \cdot (\alpha^* \nabla \alpha) \, d\mathbf{x}, & \mathbf{x} \in \Omega, \\
\varphi^* = 0 & \mathbf{x} \in \Gamma_D, \\
\nabla \varphi^* \cdot \mathbf{n} = \varphi_D^* + \int_0^T \alpha^* \nabla \alpha \cdot \mathbf{n} \, d\mathbf{x} & \mathbf{x} \in \Gamma_D, \\
\varphi^* = -\varphi_N^* & \mathbf{x} \in \Gamma_N, \\
\nabla \varphi^* \cdot \mathbf{n} = \int_0^T \alpha^* \nabla \alpha \cdot \mathbf{n} \, d\mathbf{x} & \mathbf{x} \in \Gamma_N.
\end{cases}
\]

We also have the condition:

\[ \alpha_D^* (\alpha - \alpha_D) = 0, \quad \text{on } \partial \Gamma, \]

which is automatically satisfied for all \( \alpha_D^* \) if the boundary condition on \( \alpha \) is enforced.

We can see that the Poisson equation for the adjoint velocity potential has overspecified boundary conditions. The choice in this case is made as follows:

- **On** \( \Gamma_D \), we chose the homogeneous Dirichlet boundary conditions:

\[ \varphi^*(\mathbf{x}) = 0, \]

To mimic the boundary conditions in the forward equation.
• On $\Gamma_N$, we chose the Neumann boundary conditions:

$$\nabla \phi^* \cdot n = \int_0^T \alpha^* \nabla \alpha \cdot n \, dt,$$

for two reasons: $\phi^*_N$ is unknown at this point and, in the case of $\Gamma_D = \emptyset$, the condition for the uniqueness of a solution:

$$\int_\Omega \nabla \cdot \nabla \phi \, dx = \oint_{\partial \Omega} \nabla \phi^* \cdot n \, dx = \oint_{\partial \Omega} \int_0^T (\alpha^* \nabla \alpha) \cdot n \, dt \, dx,$$

is automatically satisfied by this choice.

The adjoint Poisson equation for the velocity potential has a unique solution, given the boundary conditions defined above, so we can set the remaining adjoint variable $\phi^*_D$ from the remaining boundary condition:

$$\phi^*_D = \nabla \phi^* \cdot n - \int_0^T \alpha^* \nabla \alpha \cdot n \, dt.$$

Note that the Lagrange multipliers $(\alpha_0^*, \alpha_D^*)$ used to impose the constraints on the volume fraction and $\phi^*_D$ for the velocity potential are in fact redundant, since they can be uniquely determined from the other adjoint variables. This comes as no surprise since these variables impose fixed conditions that do not depend on any of the problem parameters.

**Optimality Condition and Gradient**

Finally, we look at the the derivatives of the Lagrangian with respect to $h$. For all admissible directions $\tilde{h}$, we get:

$$\frac{\partial \mathcal{L}}{\partial h}(h; \tilde{h}) = \gamma_3 \oint_{\Gamma_N} \tilde{h} \, dx + \oint_{\Gamma_N} \phi^*_N \tilde{h} \, dx = 0.$$

So we are left with the following *optimality condition*:

$$h = -\frac{1}{\gamma_3} \phi^*_N = \frac{1}{\gamma_3} \phi^*_N \bigg|_{\Gamma_N},$$

where we have used one of the boundary conditions on $\phi^*$ to remove $\phi^*_N$ from the equation. We also have that the gradient of the cost functional is given by:

$$\nabla J = \gamma_3 h - \phi^*_N \bigg|_{\Gamma_N},$$

by applying the Riesz Representation theorem on the space $L^2(\Gamma_N)$ with the standard inner product.
Bibliography


