Title
Models for Human Navigation and Optimal Path Planning Using Level Set Methods and Hamilton-Jacobi Equations

Author
Parkinson, Christian Andrew

Publication Date
2020

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Models for Human Navigation and Optimal Path Planning Using Level Set Methods and Hamilton-Jacobi Equations

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Christian Andrew Parkinson

2020

ABSTRACT OF THE DISSERTATION

Models for Human Navigation and Optimal Path Planning Using Level Set Methods and Hamilton-Jacobi Equations

by

Christian Andrew Parkinson
Doctor of Philosophy in Mathematics
University of California, Los Angeles, 2020
Professor Andrea Bertozzi, Co-Chair
Professor Stanley J. Osher, Co-Chair

We present several models for different physical scenarios which are centered around human movement or optimal path planning, and use partial differential equations and concepts from control theory. The first model is a game-theoretic model for environmental crime which tracks criminals' movement using the level set method, and improves upon previous continuous models by removing overly restrictive assumptions of symmetry. Next, we design a method for determining optimal hiking paths in mountainous regions using an anisotropic level set equation. After this, we present a model for optimal human navigation with uncertainty which is rooted in dynamic programming and stochastic optimal control theory. Lastly, we consider optimal path planning for simple, self-driving cars in the Hamilton-Jacobi formulation. We improve upon previous models which simplify the car to a point mass, and present a reasonably general upwind, sweeping scheme to solve the relevant Hamilton-Jacobi equation.

The dissertation of Christian Andrew Parkinson is approved.

Paul Jeffrey Brantingham

Luminita Aura Vese

Andrea Bertozzi, Committee Co-Chair

Stanley J. Osher, Committee Co-Chair

University of California, Los Angeles

2020

to Stephanie

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Thank you to my parents Charles & Lynda for instilling in me an appreciation for mathematics and reason. *Come now, let us reason together,* Isaiah 1:18.

Thank you to my whole family—Charles, Lynda, Cory, Elisabeth, Caili, Michael, Connor, Anna, and an ever-growing list of nephews and nieces—for all of the support, love, friendship, and fun times over the years. And thank you to my new family—Guy, Nancy, Ali, Adam—for all of the support, love, friendship, and fun times I am sure are to come.

Thank you to the great friends I made at UCLA—especially Austin & Kristin—for all the board games, rounds of drinks, bar trivia nights, etc. and for countless stimulating conversations, mathematical and otherwise.

Thank you to all my collaborators and anyone who has piqued my interest in mathematics over the past several years—especially Stephen Pankavich, who advised my master's thesis, and David Arnold, who worked closely with me on many projects during my time at UCLA.

Thank you to the National Geospatial-Intelligence Agency for financial support. Much of the work in this dissertation—previously published in [PAB19, AFJ19] as well as the preprints listed below—was supported by NGA NURI Award No. #HM0210-14-1-0003: *Sparsity models for spatiotemporal analysis and modeling of human activity and social networks in a geographic context.*

A special thank you to my advisors, Andrea Bertozzi and Stan Osher for sharing their incredible range and depth of knowledge, for helping to develop my interests and skills in applied mathematics and mathematical modeling, and for the patience, advice, and guidance they continually offered.

Most importantly, thank you to my wife Stephanie who has been an incarnation of grace, love and encouragement for me, who has taught me to have a hungry mind, and who has put up with me through a stressful period of job-searching and thesis-writing. I love you.

VITA

2013	B.S. Computational and Applied Mathematics, Colorado School of Mines, Golden CO.
2014	Participated in Graduate Student Mathematical Modeling Camp, Rensselaer Polytechnic Institute, Troy, NY.
2015	M.S. Computational and Applied Mathematics, Colorado School of Mines, Golden CO.
2016	Participated in Computational Physics Summer Workshop, Los Alamos National Lab, Los Alamos, NM.
2016-2017	Graduate Teaching Assistant, University of California, Los Angeles, CA.
2017-2019	Mentor, Research Experience for Undergraduates, University of California, Los Angeles, CA.
2017-2020	Graduate Research Assistant/Fellow, University of California, Los Angeles, CA.
2018	Ph.D. Candidate, Mathematics, University of California, Los Angeles, CA. Advisors: Dr. Andrea L. Bertozzi and Dr. Stanley J. Osher.
2020	Participated in Online Net-COVID Workshop: Understanding and Exploring Network Epidemiology in the Time of Coronavirus. Hosted by COMBINE program at University of Maryland and the University of Vermont Complex Systems Center.

PUBLICATIONS

A Rotating-Grid Upwind Fast Sweeping Scheme for a Class of Hamilton-Jacobi Equations, C. Parkinson, submitted to the Journal of Scientific Computing.

A Hamilton-Jacobi Formulation for Time-Optimal Paths of Rectangular Nonholonomic Vehicles, C. Parkinson, A. L. Bertozzi, and S. Osher, submitted to the 59$^{\text{th}}$ IEEE Conference on Decision and Control.

A Model for Optimal Human Navigation with Stochastic Effects, C. Parkinson, D. Arnold, A. L. Bertozzi, and S. Osher, accepted for publication in the SIAM Journal on Applied Mathematics.

Matrix Completion with Selective Sampling, C. Parkinson, K. Huynh, and D. Needell. 13$^{\text{th}}$ International Conference on Sampling Theory and Applications (SampTA), Bordeaux, France, 2019, pp. 1-4.

Modeling Environmental Crime in Protected Areas Using the Level Set Method, D. Arnold, A. L. Bertozzi, D. Fernandez, R. Jia, S. Osher, C. Parkinson, D. Tonne, and Y. Yaniv. SIAM Journal on Applied Mathematics, 79 (2019), no. 3, 802-821.

Optimal Human Navigation in Steep Terrain: A Hamilton-Jacobi-Bellman Approach, C. Parkinson, D. Arnold, A. L. Bertozzi, Y. T. Chow, and S. Osher. Communications in Mathematical Sciences, 17 (2019), 227-242.

Mathematical Analysis of an in-host Model of Viral Dynamics with Spatial Heterogeneity, S. Pankavich and C. Parkinson, Discrete and Continuous Dynamical Systems B., 21 (2016), no. 5, 1237-1257.

CHAPTER 1

Introduction & Overview

The problem of optimal path planning is fundamental to many areas of mathematics and engineering. In its most basic form, the question of determining shortest paths was encoded into ancient Greek geometry via Euclid's axioms and the triangle inequality. Modern advances in partial differential equations, the calculus of variations, and optimal control theory provide robust and widely applicable methods for computing different notions of optimal paths. In this document, we present several mathematical models that address different physical scenarios but are all centered around optimal path planning or human navigation. Each of the models makes use of Hamilton-Jacobi equations, which can be seen to describe extremal geometry problems arising from optimal control.

1.1 Organization of the Dissertation

This dissertation will be organized as follows. Chapter 2 will give a basic description of the mathematics we use throughout the remainder of the document. We will begin by describing the celebrated level set method of Osher & Sethian [OS88]. From there, we will address more general Hamilton-Jacobi equations, including a discussion of continuous viscosity solutions [CL83]. Next, we briefly discuss optimal control theory and dynamic programming, and demonstrate how level set methods and Hamilton-Jacobi equations can be seen as special cases of problems in optimal control. Finally, we present basic grid-based methods used to compute approximate solutions to time-dependent Hamilton-Jacobi equations.

In chapter 3, we present a model for illegal deforestation in areas that are protected by a law enforcement agency. The model includes a differential game wherein a group of

hypothetical attackers enter the protected region from the boundary and perpetrate crimes based on the benefit they expect to receive, the cost they expect to incur, and the likelihood that they will be captured by law enforcement. We use the level set method to model the movement of the entire group at once. As level sets evolve inward from the boundary of the set, points along the level sets follow paths corresponding to optimal travel. In our model, "optimal travel" accounts for both travel time and capture likelihood. We compare our model to previous continuous models for deforestation, which are overly simplistic in that they require the protected region to be highly symmetric and assume all quantities can be one dimensionalized. We test our model in two real domains: Yosemite National Park in California, and Kangaroo Island in South Australia.

In chapter 4, we design a model for optimal path planning in mountainous regions with anisotropic motion depending on local slope of the terrain. This model also uses the level set method to compute isocontours of optimal travels, though here the level set equation involved more explicitly describes the solution to an optimal control problem. We test our model by determining hiking paths in the vicinity of El Capitan and Half Dome, two mountains in Yosemite National Park, and address several implementation issues that are somewhat specific to this situation. We note that in mountainous regions, where geographical features such as cliff faces and mountain passes often determine the best hiking paths, the paths from different starting locations to a common ending location can be clustered into relatively few routes.

In chapter 5, we modify the optimal walking path model to include uncertainty in the walking velocity equation. Uncertainty in walking velocity supplies diffusion in the Hamilton-Jacobi equation, meaning that level set methods are no longer applicable. Accordingly, we consider a control theoretic model wherein we solve a Hamilton-Jacobi-Bellman equation and determine optimal walking directions by way of a minimization problem. We again test our model in the area surrounding El Capitan. When uncertainty is present, we consider two different notions of optimal path, and observe that as the uncertainty disappears, the optimal path converges to the deterministic optimal path.

Lastly, in chapter 6, we consider optimal path planning in a different scenario. Here we

develop a method for resolving optimal paths of simple self-driving cars in the presence of obstacles, using dynamic programming and a steady-state Hamilton-Jacobi equation. Such vehicles are subject to nonholonomic constraints, meaning the motion is not omnidirectional, as in the case of a hiker. Here the nonholonomic constraint specifies that motion occurs tangential to the rear wheels, resulting in paths of bounded local curvature. We do not simpify the car to a point mass, as is common practice with these models, meaning that no special consideration is needed near obstacles, and there is no need for hierarchical algorithms for path planning and collision avoidance. We present an upwind sweeping scheme to solve the Hamilton-Jacobi equation for the travel time function, and test our algorithm in a number of different ways.

CHAPTER 2

Level Set Methods, Hamiton-Jacobi Equations & Optimal Control Theory

In this chapter, we discuss the basics of the mathematical tools that will be used throughout the remainder of this document. We begin with a discussion of the level set method of Osher and Sethian [OS88]. A key component of the level set method is a Hamilton-Jacobi equation which describes the the evolution of a front with prescribed velocity. Accordingly, we discuss some of the classical theory of Hamiton-Jacobi equations. Next, we formulate an optimal control problem and demonstrate the link between optimal control theory and Hamilton-Jacobi equations. Finally, we discuss numerical methods for solving Hamilton-Jacobi equations.

2.1 The Level Set Method

The level set method was devised by Osher and Sethian [OS88] to track fronts that are propagating with speed depending on properties inherent to the front itself (normal direction, local curvature, etc.). It has since found numerous applications across several fields of applied mathematics and engineering [GFO18], including fluid dynamics [TBE01, SSO94a], combustion [LSC19, TR06], crystal growth [GFC03, JT96], image segmentation [JZN12, QWH07], dynamic visibility [KT08, TCO04], trajectory planning [PAB19, LUY12], and environmental crime modeling [AFJ19, CV19] to name a few.

2.1.1 Time-Dependent Formulation of the Level Set Method

Suppose that $\Gamma_0 \subset \mathbb{R}^n$ is a closed $(n-1)$-dimensional hypersurface. To evolve Γ_0 using level set motion, we first find a Lipschitz continuous function $\phi_0 : \mathbb{R}^n \to \mathbb{R}$ such that $\Gamma_0 = \{x \in \mathbb{R}^n : \phi_0(x) = 0\}$, and $\phi_0(x) < 0$ for x inside Γ_0, while $\phi_0(x) > 0$ for x outside Γ_0. That is, Γ_0 is precisely the zero level set of the auxiliary function ϕ_0. Next, we evolve $\phi : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}$ according to the partial differential equation

$$\begin{aligned} &\phi_t(x,t) + F(t, x, \nabla\phi, n, \kappa, \ldots) = 0, \\ &\phi(x,0) = \phi_0(x). \end{aligned} \tag{2.1}$$

As ϕ evolves, the zero level contour $\Gamma_t = \{x \in \mathbb{R}^n : \phi(x,t) = 0\}$ also evolves. In equation (2.1), n represents the unit normal to the current level set, κ represents the local curvature, and other geometric properties could be included. One advantage of the level set method is that these geometric properties can all be described in terms of the level set function ϕ. For example, since Γ_t is the zero level contour of $\phi(\cdot, t)$, the vector $\nabla\phi(x,t)$ points in the (outward) normal direction to Γ_t for any $x \in \Gamma_t$, and thus $n(x,t) = \nabla\phi(x,t)/|\nabla\phi(x,t)|$. Likewise, the curvature $\kappa = \nabla \cdot n$ of Γ_t can be expressed in terms of the first and second derivatives of ϕ. For example,

$$\kappa = \frac{\phi_x^2\phi_{yy} - 2\phi_x\phi_y\phi_{xy} + \phi_y^2\phi_{xx}}{(\phi_x^2 + \phi_y^2)^{3/2}} \tag{2.2}$$

in two spatial dimensions, or

$$\begin{aligned} \kappa = &(\phi_x^2\phi_{yy} - 2\phi_x\phi_y\phi_{xy} + \phi_y^2\phi_{xx} + \phi_x^2\phi_{zz} - 2\phi_x\phi_z\phi_{xz} + \phi_z^2\phi_{xx} \\ &+ \phi_y^2\phi_{zz} - 2\phi_y\phi_z\phi_{yz} + \phi_z^2\phi_{yy})/(\phi_x^2 + \phi_y^2 + \phi_z^2)^{3/2} \end{aligned} \tag{2.3}$$

in three dimensions [OF03].

In its most basic form, the level set method models growth of Γ_0 in the outward normal direction. To accomplish this level set motion, the level set function should evolve with advection in the direction of $n = \nabla\phi/|\nabla\phi|$. This results in the time-dependent Eikonal

equation

$$\begin{aligned} \phi_t + |\nabla \phi| &= 0, \\ \phi(x, 0) &= \phi_0(x). \end{aligned} \tag{2.4}$$

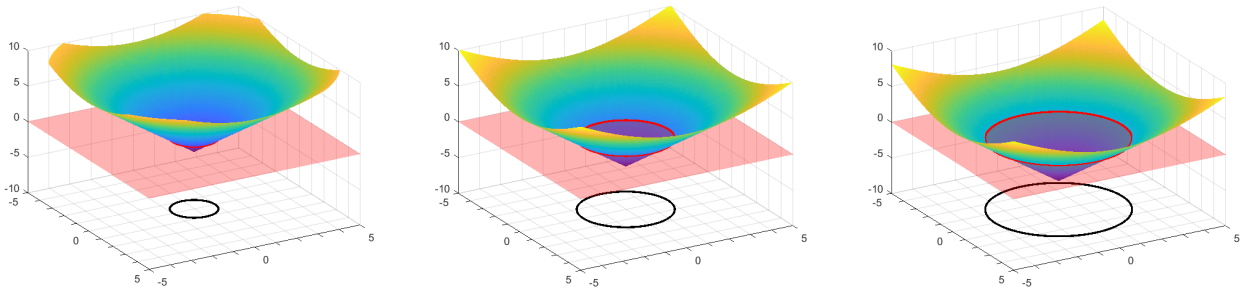
If we choose ϕ_0 to be the signed distance function to Γ_0

$$\phi_0(x) = \begin{cases} -\text{dist}(x, \Gamma_0), & x \text{ inside } \Gamma_0, \\ \text{dist}(x, \Gamma_0), & x \text{ outside } \Gamma_0, \end{cases} \qquad x \in \mathbb{R}^n, \tag{2.5}$$

then $|\nabla \phi_0| = 1$ near the zero level set, and so advancing (2.4) in time simply prescribes local translation of ϕ_0 by $-t$. This is illustrated in fig. 2.1. In that case, $\phi_0(x, y) = \sqrt{x^2 + y^2} - 1$ so that Γ_0 is the circle of radius 1. As time advances, "pulling" the initial profile down the z-axis causes the level set—which is the intersection of the surface with the xy-plane—to expand.

2.1.2 Time-Dependent Eikonal Equation: Distance Functions via Level Sets

Something to note about fig. 2.1 is that since the initial curve Γ_0 is a circle of radius 1 and the curve evolves outward with normal velocity 1, the curve Γ_t will simply be a circle of radius $1 + t$. Considering this, we see that for any point $x^{(t)} \in \Gamma_t$, we have $\text{dist}(x^{(t)}, \Gamma_0) = t$, which provides an alternate way of looking at Γ_t. For $t > 0$, the curve Γ_t is precisely the set of points at distance t from Γ_0. While it is easy to identify the set of points at distance $t > 0$

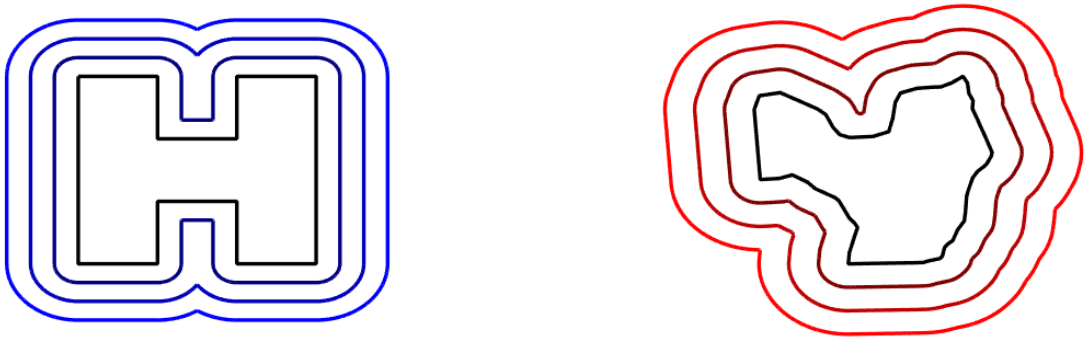


(a) The surface $(x, y, \phi_0(x, y))$ and the level set Γ_0.

(b) The surface $(x, y, \phi(x, y, 1))$ and the level set Γ_1.

(c) The surface $(x, y, \phi(x, y, 2))$ and the level set Γ_2.

Figure 2.1: Illustration of the level set method with $\phi_0(x, y) = \sqrt{x^2 + y^2} - 1$. As one "pulls" the initial profile down the z-axis, the zero level set expands.



(a) Successive isocontours of equal distance from the dumbbell shape.

(b) Successive isocontours of equal distance from the UCLA campus boundary.

Figure 2.2: Level set flow provides a method for finding isocontours of equal distance (colored) from an initial contour (black).

from a given circle, it can be quite difficult to do so with more complicated geometries, but level set flow provides a simple method. This is seen in fig. 2.2, where an artificial dumbbell shape and the campus boundary of UCLA are used as examples of irregular geometries.

This "distance function" interpretation establishes an intimate connection between level set methods and extremal geometry problems [OS01]. If we consider a particle sitting somewhere on Γ_0 and traveling outward with maximum velocity 1 for time $t > 0$, then the interior of Γ_t represents the set of points that the particle could potentially reach after traveling for time t. If instead we are given a velocity field $v(x)$ prescribing the speed with which the particle can move throughout the domain, we can still account for the maximum reachable set using the level set method. Now Γ_0 should evolve outward with normal velocity $v(x)$, meaning that the level set function should exhibit local advection in the normal direction $n = \nabla\phi / |\nabla\phi|$ with velocity $v(x)$:

$$\begin{aligned} \phi_t + v(x)\,|\nabla\phi| &= 0, \\ \phi(x,0) &= \phi_0(x), \end{aligned} \tag{2.6}$$

where ϕ_0 is the signed distance to Γ_0 as before. This is seen in fig. 2.3. Here the normal velocity is given by $v(x,y) = 1 + \frac{4}{5}\sin(x)\sin(y)$ is plotted in fig. 2.3b. There are some regions where the velocity dips to roughly 0.2 and other regions where the velocity is as high as 1.8. The corresponding level set flow is seen in fig. 2.3a. In this case, for a given $t > 0$, Γ_t

represents the boundary of the reachable set after traveling outward from the unit circle for time t in the velocity field $v(x, y)$.

2.1.3 Steady State Formulation for Monotonically Advancing Fronts

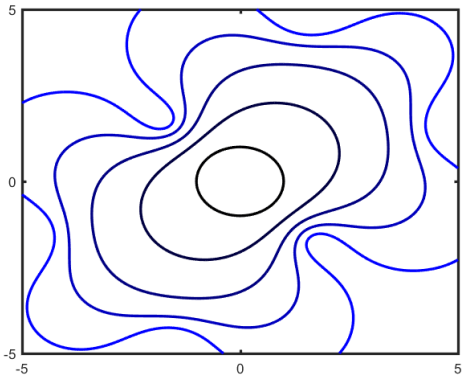
As pointed out by [Set99], the "travel-time" interpretation of the problem leads to an alternate formulation of the level set equation when the normal velocity $v(x)$ is strictly positive. In this case, as level sets Γ_t evolve outward from Γ_0, for any x in the domain, there will be a unique $T(x)$ such that $x \in \Gamma_{T(x)}$. Due to the evolution of Γ_t, the travel-time function $T(x)$ will satisfy

$$T(x + n\Delta t) = T(x) + \frac{\Delta t}{v(x)} + o(\Delta t), \tag{2.7}$$

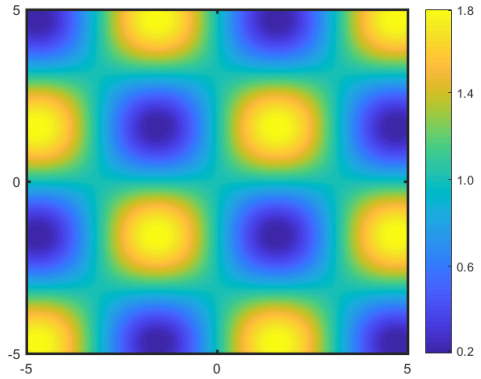
where n is the normal to the level set $\Gamma_{T(x)}$. Realizing that $n = \nabla T / |\nabla T|$, we can rearrange and send $\Delta t \to 0$ to arrive at

$$\begin{aligned} v(x) \, |\nabla T(x)| &= 1, \\ T(x) &= 0, \quad x \in \Gamma_0. \end{aligned} \tag{2.8}$$

Thus, in the case that $v(x) > 0$, the travel-time function satisfies a time-independent Eikonal equation. As described, $T(x)$ is only defined for x outside of Γ_0, but the solution to (2.8) can be defined globally and will take negative values inside Γ_0. Thus for any x, $|T(x)|$ is the travel



(a) Level sets Γ_t for $t = 0, 1, 2, 3, 4$. These represent isocontours of equal travel time from the initial contour in the given velocity field.



(b) The normal velocity $v(x, y) = 1 + \frac{4}{5}\sin(x)\sin(y)$ with which the level sets evolve.

Figure 2.3: Level sets (left) traveling outward from the unit circle with presecribed normal velocity (right).

time to Γ_0, and $\text{sign}(T(x))$ indicates whether x is inside or outside of Γ_0. We can handle "inward" level set evolution with the time-dependent formulation (2.6) by either running time in reverse, or by swapping ϕ_0 for $-\phi_0$ to change the orientation of the problem. This shows that when normal velocity is positive, (2.6) and (2.8) describe equivalent dynamics with the understanding that $\phi(x,t) = 0$ if and only if $T(x) = t$.

Each of these formulations is useful in different cases. The time-dependent formulation (2.6) allows for non-positive normal velocity, can easily account for more sophisticated geometric effects, and can be solved numerically at arbitrarily high-order with little added effort [OF03, OS91]. The time-independent formulation (2.8) bears a lower computational burden due to removal of the time dimension, and can be approximated efficiently using fast marching methods [SV01, SV03]. We primarily deal with the time-dependent formulation (2.6) and its variations for the remainder of this document.

2.1.4 General Level Set Equations

More generally, level set equations are a subclass of Hamilton-Jacobi equations. The general time-dependent Hamilton-Jacobi equation reads

$$\begin{aligned} &\phi_t + H(t, x, \nabla\phi) = 0, \\ &\phi(x, 0) = \phi_0(x), \end{aligned} \tag{2.9}$$

for some Hamiltonian function $H(t, x, p)$. Here we use p as a proxy for $\nabla\phi$. We call (2.9) a level set equation if H is positively homogeneous of degree 1 in the variable p:

$$H(t, x, \alpha p) = |\alpha|\, H(t, x, p), \tag{2.10}$$

for all (t, x, p) in the domain and all $\alpha \in \mathbb{R}$. Note that this condition is trivially satisfied by (2.4) where $H(x, p) = v(x)\,|p|$. This homogeneity is important because *a priori* we are only concerned with the moving front Γ_t and the level set function ϕ may not have any physical meaning. Since there are several functions with the same same zero level sets, the choice of

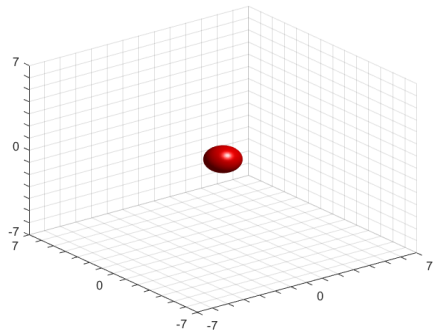
which particular function is used to describe Γ_t should be somewhat arbitrary, and this is captured by the homogeneity in the following way. Suppose $\phi(x,t)$ satisfies (2.9) and that H is homogeneous of degree 1 in p, and define $\Gamma_t = \{x \in \mathbb{R}^n : \phi(x,t) = 0\}$. Further suppose that $f : \mathbb{R} \to \mathbb{R}$ is a smooth, strictly increasing map so that $f' > 0$. Then $\phi(x,t)$ and $\psi(x,t) \coloneqq f(\phi(x,t))$ have the same zero level sets Γ_t. Accordingly, we would like $\psi(x,t)$ to solve the same level set equation as $\phi(x,t)$—with a modified initial condition—and indeed

$$\psi_t + H(t,x,\nabla\psi) = f'(\phi)\phi_t + H(t,x,f'(\phi)\nabla\phi) = f'(\phi)(\phi_t + H(t,x,\nabla\phi)) = 0.$$

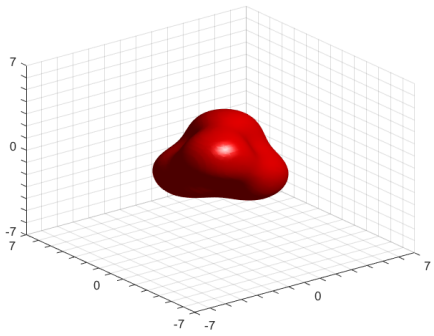
This ensures that for level set equations, the essential piece of data is the initial contour Γ_0, and not the function used to model it. In certain cases, a particular level set function ϕ can have a meaningful interpretation in its own right, and in these cases there is a *best* level set function in some sense, which we address later.

In the ensuing sections, we briefly discuss some of the classical theory of Hamilton-Jacobi equations and see how Hamilton-Jacobi equations and level set methods can be used to solved problems in optimal control theory, but we close this section by pointing out two strengths of the level set method as it pertains to tracking fronts. First, while all the examples shown thus far use $\mathbb{R}^2$ as the domain, the level set method generalizes trivially to any dimension (none of the above discussion requires that $n = 2$). Figure 2.4 shows a three-dimensional analog to the level set motion in fig. 2.3. In this case, we let $\phi_0(x,y,z) = \sqrt{x^2+y^2+z^2} - 1$ so that Γ_0 is the unit sphere in $\mathbb{R}^3$. We then evolve Γ_0 with normal velocity $v(x,y,z) = 1 + \frac{4}{5}\sin(x)\sin(y)\sin(z)$ and display the level sets Γ_t at time $t = 0, 2, 4$.

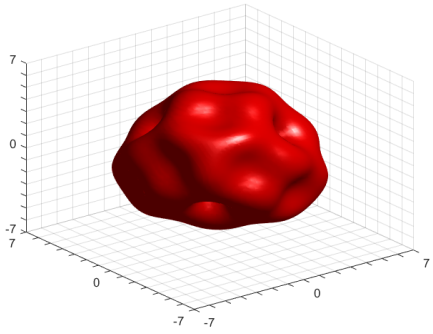
Second, a key component of any front tracking method is how it handles changes in topology. For example, the initial front Γ_0 could be disconnected—one can envision two bubbles expanding independently—while the disconnected pieces could combine to make a single front Γ_t at some finite time $t > 0$. If we are simply tracking points along the front, it may be difficult to ascertain when such a change in topology has occured, but the level set method can handle this with no special consideration since the change in topology occurs only in the projection to the lower dimension, not in the surface $(x, \phi(x,t))$. This can be



(a) The level set Γ_0.

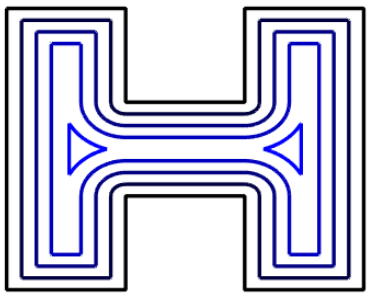


(b) The level set Γ_2.



(c) The level set Γ_4.

Figure 2.4: Level set flow in $\mathbb{R}^3$. Γ_0 is the unit sphere and is evolved with normal velocity $v(x, y, z) = 1 + \frac{4}{5}\sin(x)\sin(y)\sin(z)$.



(a) Level sets propagating inward from the dumbbell eventually break apart.

(b) Level sets propagating outward from three circles eventually join together.

Figure 2.5: The level set method has no problems tracking changes in the topology of the curve.

seen in fig. 2.5. In fig. 2.5a, the initial contour is the same dumbbell shape as before, but the level sets propagate inward, eventually splitting from one curve into two. In fig. 2.5b, the initial contour is the collection of three disjoint circles. As the level sets propagate outward, they eventually join together.

2.2 Hamilton-Jacobi Equations

In this section, we formally address some of the classical theory of Hamilton-Jacobi equations. Hamilton-Jacobi equations have been studied in connection with classical mechanics since the 1800s [Ham33, Ham34], but the mathematical theory surrounding them has been established relatively recently. As stated above, the general time-dependent Hamilton-Jacobi equation is given by

$$\phi_t + H(t, x, \nabla\phi) = 0, \quad x \in \mathbb{R}^n, t > 0, \tag{2.11}$$

where $H : (0, \infty) \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is called the *Hamiltonian*. In this case, an initial condition $\phi(x, 0) = \phi_0(x)$ is often specified. We may restrict to a domain $\Omega \subset \mathbb{R}^n$ and specify a boundary condition that ϕ must satisfy on $\partial\Omega$. This is especially true for the steady state Hamilton-Jacobi equation

$$\tilde{H}(x, \nabla\phi) = 0, \quad x \in \Omega, \tag{2.12}$$

where $\tilde{H} : \Omega \times \mathbb{R}^n \to \mathbb{R}$. Here Ω is typically a bounded, open set and one is given prescribed boundary data. Often in application—as is the case with (2.6, 2.8)—the Hamiltonian does not depend explicitly on the variables x and t, so we may suppress this dependence later in this document when no confusion could arise.

Note that (2.11) can actually be envisioned as a special case of (2.12) by considering new coordinates $\tilde{x} = (t, x)$ and $\tilde{\nabla} = (\partial_t, \nabla)$, so we will focus some of our discussion on (2.12). But equations of the form (2.11) arise naturally as level set equations as above, and in optimal control theory, so they are worth discussing in their own right.

2.2.1 Lack of Regularity and Non-Uniqueness

Since (2.11) is in general nonlinear, we do not expect globally smooth solutions to exist, even when the initial data ϕ_0 is smooth. Consider the case of the one-dimensional time-dependent Hamilton-Jacobi equation,

$$\phi_t + H(\phi_x) = 0, \tag{2.13}$$

with H smooth. Formally, differentiating by x and setting $u = \phi_x$ we see that

$$u_t + H(u)_x = 0. \tag{2.14}$$

That is, the x-derivative of ϕ solves a scalar conservation law. The solutions of conservation laws are well-known to develop discontinuities along shocks, necessitating the theory of discontinuous entropy solutions. Thus, since ϕ is the integral of a solution of a conservation law, we can expect ϕ to remain continuous, but develop discontinuities in the first derivative. While this explicit link between Hamilton-Jacobi equations and conservation laws is no longer available in higher dimensions, the general conclusion still holds, and we must consider some notion of a weak solution.

In the 1980s, Crandall and Lions [CL83, CL84] introduced the notion of continuous *viscosity solutions* for Hamilton-Jacobi equations. We build toward the definition using, as a motivating example, the steady-state Eikonal equation in one dimension:

$$\begin{aligned} &|\phi'(x)| - 1 = 0, \quad x \in (-1, 1), \\ &\phi(-1) = \phi(1) = 0. \end{aligned} \tag{2.15}$$

By the mean value theorem, any smooth function satisfying the boundary conditions would need to have $\phi'(x) = 0$ for some $x \in (-1, 1)$, and thus cannot satisfy (2.15). On the other hand, it is trivial to build infinitely many Lipschitz continuous functions that satisfy the equation almost everywhere. Figure 2.6 shows three such solutions that are constucted by flipping and translating copies of the function $f(x) = |x|$. Thus it is natural to ask which of these almost everywhere solutions is the "correct" solution.

2.2.2 Vanishing Viscosity for the 1D Eikonal Equation

To settle on a solution of (2.15), we perturb the equation by $\varepsilon > 0$ and instead consider

$$\begin{aligned} &\varepsilon\phi_\varepsilon''(x) + \phi_\varepsilon'(x)^2 - 1 = 0, \quad x \in (-1, 1), \\ &\phi_\varepsilon(-1) = \phi_\varepsilon(1) = 0. \end{aligned} \tag{2.16}$$

We refer to $\varepsilon\phi_\varepsilon''(x)$ as a *viscosity* term, in analogy to fluid dynamics where diffusion arises due to viscosity of the liquid. Note that as $\varepsilon \to 0$, the equation returns to an equivalent formulation of (2.15). However, by adding viscosity, we have changed a fully nonlinear equation into a quasilinear, elliptic equation. Using the classical theory of elliptic regularity, one can prove existence and uniqueness of a smooth solution of (2.16) [GT01]. In this case, we can explicitly resolve the solution using basic methods from ordinary differential equations. Rearranging the equation gives

$$\frac{\varepsilon\phi_\varepsilon''(x)}{1-\phi_\varepsilon'(x)^2} = 1, \tag{2.17}$$

whence integrating yields

$$\frac{\varepsilon}{2}\log\left(\frac{1+\phi_\varepsilon'(x)}{1-\phi_\varepsilon'(x)}\right) = x + C, \tag{2.18}$$

for a constant C which will change from line to line. Isolating ϕ_ε', we find

$$\phi_\varepsilon'(x) = \frac{1-Ce^{2x/\varepsilon}}{1+Ce^{2x/\varepsilon}} = 1 - \frac{2Ce^{2x/\varepsilon}}{1+Ce^{2x/\varepsilon}}. \tag{2.19}$$

Integrating again and enforcing $\phi_\varepsilon(-1) = 0$ yields

$$\phi_\varepsilon(x) = \int_{-1}^{x}\left(1 - \frac{2Ce^{2y/\varepsilon}}{1+Ce^{2y/\varepsilon}}\right)dy = x + 1 - \varepsilon\log\left(\frac{1+Ce^{2x/\varepsilon}}{1+Ce^{-2/\varepsilon}}\right). \tag{2.20}$$

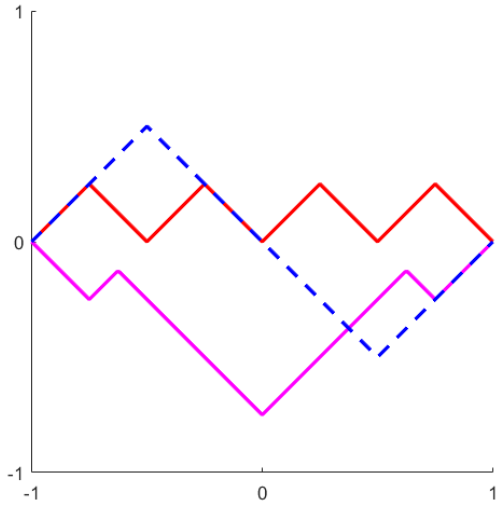


Figure 2.6: Three a.e. solutions to $|\phi'(x)| - 1 = 0$ that satisfy $\phi(-1) = \phi(1) = 0$.

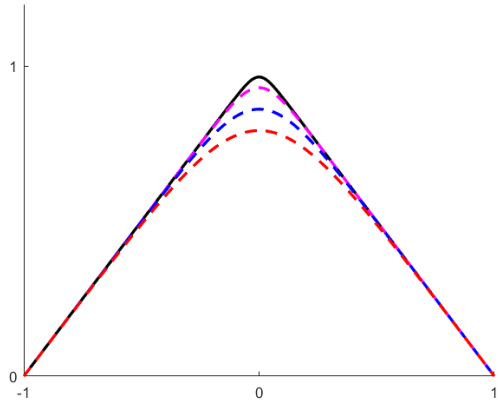


Figure 2.7: $\phi_\varepsilon(x)$ for $\varepsilon = 0.3$ (dotted red), 0.2 (dotted blue), 0.1 (dotted magenta), 0.05 (black).

Finally, enforcing $\phi_\varepsilon(1) = 0$ shows that $C = 1$ and gives the solution

$$\phi_\varepsilon(x) = x + 1 - \varepsilon \log\left(\frac{1 + e^{2x/\varepsilon}}{1 + e^{-2/\varepsilon}}\right), \quad x \in (-1, 1). \tag{2.21}$$

Plots of this solution at different levels of $\varepsilon > 0$ are displayed in fig. 2.7. From here we can intuit that $\phi_\varepsilon(x) \to 1 - |x|$ as $\varepsilon \to 0^+$. It is easy to justify this limit: if $x \leq 0$, then $\frac{1+e^{2x/\varepsilon}}{1+e^{-2/\varepsilon}}$ remains bounded as $\varepsilon \to 0^+$ and so $\phi_\varepsilon(x) \to 1 + x$. If $x > 0$, then $\frac{1+e^{2x/\varepsilon}}{1+e^{-2/\varepsilon}} \sim e^{2x/\varepsilon}$ as $\varepsilon \to 0^+$, and so

$$\lim_{\varepsilon \to 0^+} \phi_\varepsilon(x) = x + 1 - \lim_{\varepsilon \to 0^+} \left[\varepsilon \log(e^{2x/\varepsilon})\right] = x + 1 - 2x = 1 - x. \tag{2.22}$$

Putting this together, we have

$$\phi(x) \coloneqq \lim_{\varepsilon \to 0^+} \phi_\varepsilon(x) = \begin{Bmatrix} 1 + x, & x \leq 0, \\ 1 - x, & x > 0, \end{Bmatrix} = 1 - |x|. \tag{2.23}$$

We observe that ϕ as defined by (2.23) is indeed an almost everywhere solution of (2.15) and satisfies the boundary conditions. This method for generating a solution to a Hamilton-Jacobi equation is precisely the *vanishing viscosity* method introduced by [CL83], and ϕ defined as the $\varepsilon \to 0^+$ limit of the solution of (2.16) is the *viscosity solution* of the Hamilton-Jacobi equation (2.15). (Note, this is not the definition of the viscosity solution; merely an alternate characterization that is correct in this case.) One may ask what makes this solution better than those suggested in fig. 2.6. The answer is two-fold. First, this notion of solution

turns out to be the correct one for proving existence and uniqueness results. For elliptic equations these results are often accomplished using tools such as comparison principles or Perron's method, and some of these tools are preserved in the limit as $\varepsilon \to 0^+$ [Bar13]. Second, the viscosity solution turns out to have the correct physical interpretation in many applications such as optimal control [BC08], mean field games [BFY13, GLL11], and mean curvature flow [CGG99].

2.2.3 Viscosity Solutions of Hamilton-Jacobi Equations with Application to the 1D Eikonal Equation

With the previous example in mind, we can define the viscosity solution of (2.11) and (2.12), and discuss some of the basic properties. We largely follow [BC08], omitting proofs in favor of examples and discussion.

Definition (Viscosity Solution). The function $\phi \in C(\Omega)$ is a *viscosity subsolution* of (2.12) if and only if for any test function $v \in C^1(\Omega)$, if $x_0 \in \Omega$ is a local maximum of $\phi - v$, then

$$\tilde{H}(x_0, \nabla v(x_0)) \leq 0. \tag{2.24}$$

Likewise, the function $\phi \in C(\Omega)$ is a *viscosity supersolution* of (2.12) if and only if for any test function $v \in C^1(\Omega)$, if $x_0 \in \Omega$ is a local minimum of $\phi - v$, then

$$\tilde{H}(x_0, \nabla v(x_0)) \geq 0. \tag{2.25}$$

Finally, $\phi \in C(\Omega)$ is a *viscosity solution* of (2.12) if and only if it is both a viscosity subsolution and a viscosity supersolution.

In the case of the time-dependent equation (2.11), the above equations become

$$v_t(x_0, t_0) + H(t_0, x_0, \nabla v(x_0, t_0)) \leq 0, \tag{2.26}$$

whenever (x_0, t_0) is a local maximum of $\phi - v$, and

$$v_t(x_0, t_0) + H(t_0, x_0, \nabla v(x_0, t_0)) \geq 0, \tag{2.27}$$

whenever (x_0, t_0) is a local minimum of $\phi - v$, respectively.

Note that in these definitions, we can add a constant to v without changing anything. Thus we can assume without loss of generality that $v(x_0) = \phi(x_0)$, so that the graph of v touches ϕ from above at x_0 when ϕ is a subsolution, and the graph of v touches ϕ from below at x_0 when ϕ is a supersolution.

We can easily prove that $\phi(x) = 1 - |x|$ is a viscosity solution to (2.15) according to these definitions. In this case, $\tilde{H}(x, \phi'(x)) = |\phi'(x)| - 1$. Suppose that $v \in C^1((-1, 1))$ is such that $\phi - v$ has a local extremum at $x_0 \in (-1, 1)$. If $x_0 \neq 0$, this means $\phi'(x_0) - v'(x_0) = 0$ and so $|v'(x_0)| = |\phi'(x_0)| = 1$. This shows that both the subsolution and supersolution conditions hold trivially. Next, consider $x_0 = 0$, and without loss of generality, let $v(0) = \phi(0) = 1$. Note that 0 cannot be a local minimum of $\phi - v$; this would imply that $1 - |x| - v(x) \geq 0$ in a for x in a neighborhood of zero. But then $-x \geq v(x) - v(0) \geq x$, whence dividing by x and taking a limit shows $-1 \geq v'(0) \geq 1$; a contradiction. Thus ϕ vacuously satisfies the supersolution condition at $x_0 = 0$. Conversely, if $x_0 = 0$ is a local maximum, this same computation holds with the inequalities flipped, showing that $\tilde{H}(0, v'(0)) \leq 0$ and so ϕ satisfies the subsolution condition at $x_0 = 0$.

While this shows that $\phi(x) = 1 - |x|$ is a viscosity solution of $\tilde{H}(x, \phi'(x)) = |\phi'(x)| - 1 = 0$, it is also helpful to note that the other almost everywhere solutions from fig. 2.6 are *not* viscosity solutions. Using $\psi(x) = |x| - 1$ as an example, we see that ψ is still an almost everywhere solution to the equation. However, taking $v(x) = -x^2$, we note that $\psi - v$ has a local minimum at $x_0 = 0$, while $\tilde{H}(0, v'(0)) = -1 < 0$, violating the supersolution condition. A computation similar to above verifies that ψ—and indeed any of the functions from fig. 2.6—is a viscosity subsolution. However, any function having points where the derivative discontinuously jumps from negative to positive will violate the supersolution condition in this example. Thus viscosity solutions to this particular equation are allowed to

exhibit "infinite acceleration" in the negative direction but not in the positive direction. This requirement is somewhat analogous to the entropy condition for shock-exhibiting solutions to conservation laws, and under the relation established for (2.13) and (2.14), viscosity solutions can be seen as integrated entropy solutions [AS06].

2.2.4 Properties of Viscosity Solutions

We briefly address some of the nice properties of viscosity solutions. The following proposition establishes that viscosity solutions are a generalization of classical solutions.

Proposition. Suppose that $\phi \in C^1(\Omega)$ is a classical solution of (2.12). Then ϕ is a viscosity solution of (2.12).

Conversely, suppose that $\phi \in C(\Omega)$ is a viscosity solution of (2.12) and suppose that ϕ is differentiable at $x_0 \in \Omega$. Then

$$\tilde{H}(x_0, \nabla\phi(x_0)) = 0. \tag{2.28}$$

That is, a viscosity solution satisfies the equation wherever it is differentiable.

This proposition is trivial to prove from the definition simply by observing that if ϕ is differentiable at a point $x_0 \in \Omega$ and $\phi - v$ has a local extremum at x_0, then $\nabla\phi(x_0) = \nabla v(x_0)$. As stated above, viscosity solutions have some of the nice properties expected of solutions of elliptic equations. First and foremost, existence and uniqueness can be proved using similar methods as used for elliptic equations.

Henceforth, we deal with the time-dependent formulation (2.11). We assume that the Hamiltonian H is uniformly continuous on $[0,T] \times \mathbb{R}^n \times \overline{B(0,r)}$ for any $T, r > 0$ and satisfies a Lipschitz type inequality:

$$|H(t,x,p) - H(t,y,p)| \leq \omega(|x-y|\,(1+|p|)), \tag{H1}$$

for all $t \in (0, \infty)$ and $x, y, p \in \mathbb{R}^n$, where $\omega : [0, \infty) \to [0, \infty)$ is continuous and increasing with $\omega(0) = 0$. With these assumptions, we have the following two results. The proofs of these and more general statements can be found in [Bar13, BC08].

Theorem (Comparison Principle). Suppose that $\phi, \psi \in C(\mathbb{R}^n \times [0, T])$ are a viscosity subsolution and viscosity supersolution of (2.12), respectively. If $\phi(x, 0) \leq \psi(x, 0)$ for all $x \in \mathbb{R}^n$, then $\phi(x, t) \leq \psi(x, t)$ for all $(x, t) \in \mathbb{R}^n \times (0, \infty)$.

Theorem (Existence via Perron's Method). Suppose that $\phi_0 \in C(\mathbb{R}^n)$ is bounded and uniformly continuous. Let $\mathcal{S}$ denote the set of viscosity subsolutions ψ of (2.11) with $\psi(x, 0) \leq \phi_0(x)$ for $x \in \mathbb{R}^n$. Then $\phi : \mathbb{R}^n \times (0, T]$ defined by

$$\phi(x, t) \coloneqq \sup_{\psi \in \mathcal{S}} \psi(x, t), \quad (x, t) \in \mathbb{R}^n \times (0, T] \tag{2.29}$$

is a viscosity solution of (2.11).

The first theorem trivial provides uniqueness of the viscosity solution. The theorem can be modified to allow for an open, bounded spatial domain $\Omega \subset \mathbb{R}^n$, by requiring that the boundary inequality $\phi \leq \psi$ holds on the parabolic boundary $\Omega \times \{0\} \cup \partial\Omega \times (0, \infty)$.

For the second theorem, it is important to note that the set $\mathcal{S}$ is non-empty. We can explicitly construct one member of $\mathcal{S}$. If $A = \sup_{(x,t)} |H(t, x, 0)|$, then $\psi^*(x, t) = -\|\phi_0\|_\infty - At$ is a viscosity subsolution satisfying the bound at $t = 0$. This method is analogous to Perron's method for solving Laplace's equation, wherein one maximizes the set of subharmonic functions to arrive at a harmonic function. While the theorem is stated in the time-dependent case, it holds slightly more generally. It works in our example (2.15). We remarked earlier that any sawtooth function like those in fig. 2.6 is a viscosity subsolution to (2.15), while the only viscosity solution is $\phi(x) = 1 - |x|$ which is the supremum of all such functions.

We close our discussion of Hamilton-Jacobi equations and viscosity solutions by justifying the term "viscosity solution." As noted above, this terms refers to fluid dynamics, where

the inviscid Euler equations can be seen formally as the "vanishing viscosity" limit of the Navier-Stokes equations [Con06]. Here we can take the vanishing viscosity limit as in (2.16) to arrive at the viscosity solution.

Theorem (Vanishing Viscosity Method). For $\varepsilon > 0$, suppose that $\phi^{(\varepsilon)} : \mathbb{R}^n \times [0, T]$ satisfies the Hamilton-Jacobi equation with diffusion:

$$\begin{aligned} &\phi_t^{(\varepsilon)}(x,t) + H(t, x, \nabla\phi^{(\varepsilon)}) = \varepsilon\Delta\phi^{(\varepsilon)}, \quad (x,t) \in \mathbb{R}^n \times (0,\infty), \\ &\phi^{(\varepsilon)}(x,0) = \phi_0(x). \end{aligned} \tag{2.30}$$

Then as $\varepsilon \to 0^+$, we have $\phi^{(\varepsilon)} \to \phi$ locally uniformly where ϕ is the viscosity solution to (2.11).

This theorem was first proven in a slightly different form by Evans [Eva80] before the notion of viscosity solutions was well established, and later re-proven in this context [CEL84]. This explains the name and is a perfectly valid alternate characterization of viscosity solution in the case of Hamilton-Jacobi equations. However, the notion of viscosity solution extends very naturally to second-order degenerate elliptic equations where the method of vanishing viscosity is no longer broadly applicable, and one must define the viscosity solution in terms of the inequalities (2.24, 2.25) [BC08]. Thus the name is perhaps too liberally applied.

With this, we move on to discuss the basics of optimal control theory and dynamic programming, and the connection between these fields and Hamilton-Jacobi equations.

2.3 Optimal Control Theory

Modern optimal control theory dates to the 1950s, when Pontryagin established his optimality principle [Pon58, PBG62] and Bellman developed the theory of dynamic programming [Bel54, Bel61]. Bryson traces the early roots of optimal control back through variational calculus, stochastic processes and nonlinear programming [Bry96]. We focus much of our discussion on the dynamic programming approach of Bellman, and provide examples linking

optimal control theory to level set equations. We will again operate mostly formally. Similar derivations in slightly different scenarios and with varying levels of rigor can be found in a number of sources [Ber00, Bry18, CD18, Eva10, FR75, Kno81, Nis09, Pha09].

2.3.1 The Basic Optimal Control Problem

The basic optimal control problem involves a state equation that is parameterized by an input variable which can be chosen by some external user. Different parameter inputs will "steer" the state along different trajectories, and the goal of the user is to find the optimal trajectory in terms of some cost or payoff functional.

Assume that the state $\boldsymbol{x} : [0, T] \to \mathbb{R}^n$ satisfies the differential equation

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= f(t, \boldsymbol{x}(t), \boldsymbol{a}(t)), \qquad t \in [0, T], \\ \boldsymbol{x}(0) &= x_0, \end{aligned} \tag{2.31}$$

where x_0 is some initial state. Here $f : [0, T] \times \mathbb{R}^n \times A \to \mathbb{R}^n$ is some function describing the state dynamics, and it is parameterized by $\boldsymbol{a} : [0, T] \to A$. For simplicity, we can suppose $A \subset \mathbb{R}^m$, and we call this the set of admissable control actions. At each time $t \in [0, T]$, the value $\boldsymbol{a}(t)$ defines the control action chosen by the user. Equation (2.31) defines a family of trajectories parameterized by functions $\boldsymbol{a} \in \mathcal{A}$; where $\mathcal{A}$ is the set of measurable functions from $[0, T]$ to A. The cost associated with a given trajectory is

$$C[\boldsymbol{x}(\cdot), \boldsymbol{a}(\cdot)] = g(\boldsymbol{x}(T)) + \int_0^T r(t, \boldsymbol{x}(t), \boldsymbol{a}(t))dt, \tag{2.32}$$

where $r : [0, T] \times \mathbb{R}^n \times A \to \mathbb{R}$ is the marginal running cost along the trajectory and $g : \mathbb{R}^n \to \mathbb{R}$ measures the exit cost (in application, we usually have $r, g \geq 0$ but this assumption is not necessary in formulating the problem).

The optimization problem one wishes to solve is

$$\inf_{\boldsymbol{a} \in \mathcal{A}} C[\boldsymbol{x}(\cdot), \boldsymbol{a}(\cdot)] \quad \text{subject to the state equation (2.31).} \tag{2.33}$$

That is, we wish to find the optimal control plan $\boldsymbol{a}(\cdot)$, which determines the optimal trajectory $\boldsymbol{x}(\cdot)$ in terms of the cost functional C. This is the most basic optimal control problem. In many cases one wishes to maximize the functional rather than minimize, whence we would refer to it is a "payoff" or "profit" functional and replace C with P. The analysis remains identical modulo some sign changes so in our derivations we will assume we are minimizing cost. One can further modify the equation to allow for other physical scenarios such as infinite horizon time, a compact spatial domain or a fixed endpoint.

2.3.2 The Value Function, the Dynamic Programming Principle, and the Hamilton-Jacobi-Bellman Equation

To solve (2.33), we fix $x \in \mathbb{R}^n$ and $t \in [0,T]$, consider the modified "remaining cost" functional

$$C_{x,t}[\boldsymbol{x}(\cdot), \boldsymbol{a}(\cdot)] = g(\boldsymbol{x}(T)) + \int_t^T r(s, \boldsymbol{x}(s), \boldsymbol{a}(s))ds, \tag{2.34}$$

where we restrict the state equation to the interval $[t,T]$ and consider trajectories with $\boldsymbol{x}(t) = x$. Define the value function

$$\phi(x,t) = \inf_{\boldsymbol{a} \in \mathcal{A}} C_{x,t}[\boldsymbol{x}(\cdot), \boldsymbol{a}(\cdot)], \quad (x,t) \in \mathbb{R}^n \times [0,T]. \tag{2.35}$$

The value $\phi(x,t)$ is the optimal remaining cost incurred by a trajectory that is at x at time t. Then our original optimization problem (2.33) boils down to finding $\phi(x_0, 0)$. In order to do so, we derive a partial differential equation solved by ϕ.

First, notice that for any $x \in \mathbb{R}^n$, we have $\phi(x,T) = g(x)$ because there is no remaining trajectory after time T so the cost incurred is the exit cost. Fix $x \in \mathbb{R}^n, t \in [0,T)$ and $\Delta t > 0$ such that $t + \Delta t \leq T$. Consider a trajectory beginning at $\boldsymbol{x}(t) = x$, following an arbitrary control plan $\boldsymbol{a}(\cdot)$ on the interval $[t, t+\Delta t]$, and ending at some point $\boldsymbol{x}(t+\Delta t)$. From here, we follow an almost optimal control plan: let $\varepsilon > 0$ and find a control $\boldsymbol{a}_\varepsilon(\cdot)$ such that

$$g(\boldsymbol{x}_\varepsilon(T)) + \int_{t+\Delta t}^T r(s, \boldsymbol{x}_\varepsilon(s), \boldsymbol{a}_\varepsilon(s))ds \leq \phi(\boldsymbol{x}(t+\Delta t), t+\Delta t) + \varepsilon, \tag{2.36}$$

where $\boldsymbol{x}_\varepsilon(\cdot)$ is the trajectory resulting from $\boldsymbol{a}_\varepsilon(\cdot)$. Then following the control plan $\boldsymbol{a}(\cdot)$ on $[t, t+\Delta t]$ and then $\boldsymbol{a}_\varepsilon(\cdot)$ on $(t+\Delta t, T]$, we see that

$$\begin{aligned}\phi(x,t) &\le g(\boldsymbol{x}_\varepsilon(T)) + \int_{t+\Delta t}^{T} r(s, \boldsymbol{x}_\varepsilon(s), \boldsymbol{a}_\varepsilon(s))ds + \int_t^{t+\Delta t} r(s, \boldsymbol{x}(s), \boldsymbol{a}(s))ds \\ &\le \phi(\boldsymbol{x}(t+\Delta t), t+\Delta t) + \int_t^{t+\Delta t} r(s, \boldsymbol{x}(s), \boldsymbol{a}(s))ds + \varepsilon.\end{aligned} \tag{2.37}$$

Taking the infimum over all such $\boldsymbol{a}(\cdot)$ shows that

$$\phi(x,t) \le \inf_{\boldsymbol{a}\in\mathcal{A}} \left\{ \phi(\boldsymbol{x}(t+\Delta t), t+\Delta t) + \int_t^{t+\Delta t} r(s, \boldsymbol{x}(s), \boldsymbol{a}(s))ds \right\} + \varepsilon. \tag{2.38}$$

On the other hand, we can find another almost optimal control plan $\boldsymbol{a}^*_\varepsilon(\cdot)$, with corresponding trajectory $\boldsymbol{x}^*_\varepsilon(\cdot)$ such that

$$\int_t^T r(s, \boldsymbol{x}^*_\varepsilon(s), \boldsymbol{a}^*_\varepsilon(s))ds + g(\boldsymbol{x}^*_\varepsilon(T)) \le \phi(x,t) + \varepsilon. \tag{2.39}$$

By its very definition, we have

$$\phi(\boldsymbol{x}^*_\varepsilon(t+\Delta t), t+\Delta t) \le \int_{t+\Delta t}^T r(s, \boldsymbol{x}^*_\varepsilon(s), \boldsymbol{a}^*_\varepsilon(s))ds + g(\boldsymbol{x}^*_\varepsilon(T)), \tag{2.40}$$

which, combined with (2.39) gives

$$\inf_{\boldsymbol{a}\in\mathcal{A}} \left\{ \phi(\boldsymbol{x}(t+\Delta t), t+\Delta t) + \int_t^{t+\Delta t} r(s, \boldsymbol{x}(s), \boldsymbol{a}(s))ds \right\} - \varepsilon \le \phi(x,t). \tag{2.41}$$

Since $\varepsilon > 0$ is arbitrary, equations (2.38, 2.41) together yield the celebrated dynamic programming principle of Bellman [Bel54]:

$$\phi(x,t) = \inf_{\boldsymbol{a}\in\mathcal{A}} \left\{ \phi(\boldsymbol{x}(t+\Delta t), t+\Delta t) + \int_t^{t+\Delta t} r(s, \boldsymbol{x}(s), \boldsymbol{a}(s))ds \right\}. \tag{2.42}$$

From here we can rearrange and divide by Δt to find

$$\inf_{\boldsymbol{a}\in\mathcal{A}}\left\{\frac{\phi(\boldsymbol{x}(t+\Delta t),t+\Delta t)-\phi(x,t)}{\Delta t}+\frac{1}{\Delta t}\int_t^{t+\Delta t} r(x,\boldsymbol{x}(x),\boldsymbol{a}(s))ds\right\}=0. \tag{2.43}$$

Assuming that $\phi\in C^1(\mathbb{R}^n\times[0,T])$ and sending $\Delta t\to 0^+$ yields

$$\phi_t(x,t)+\inf_{a\in A}\left\{\langle\dot{\boldsymbol{x}}(t),\nabla\phi(x,t)\rangle+r(t,x,a)\right\}=0. \tag{2.44}$$

Finally, considering the state equation (2.31) and the terminal condition, this shows that ϕ satisfies the terminal value problem

$$\begin{aligned}&\phi_t(x,t)+\inf_{a\in A}\left\{\langle f(t,x,a),\nabla\phi(x,t)\rangle+r(t,x,a)\right\}=0,\quad (x,t)\in\mathbb{R}^n\times[0,T),\\&\phi(x,T)=g(x),\quad x\in\mathbb{R}^n.\end{aligned} \tag{2.45}$$

Equation (2.45) is called the *Hamilton-Jacobi-Bellman equation* for this optimal control problem. Note that it runs backwards in time. In general, one cannot expect that $\phi\in C^1(\mathbb{R}^n\times[0,T])$ even when f,r,g are as nice as desired. However, under very mild conditions on the data, ϕ will be the unique viscosity solution of (2.45) [BC08].

2.3.3 Optimal Trajectory Generation

Given the Hamilton-Jacobi-Bellman equation (2.45), one can generate optimal trajectories as follows. As the equation is integrated backwards in time, set

$$\boldsymbol{a}^*(x,t)\coloneqq\operatorname*{argmin}_{a\in A}\left\{\langle f(t,x,a),\nabla\phi(x,t)\rangle+r(t,x,a)\right\}. \tag{2.46}$$

Then the solution $\boldsymbol{x}^*(\cdot)$ to the equation

$$\begin{aligned}&\dot{\boldsymbol{x}}(t)=f(t,\boldsymbol{x}(t),\boldsymbol{a}^*(\boldsymbol{x}(t),t)),\quad t\in(0,T],\\&\boldsymbol{x}(0)=x_0,\end{aligned} \tag{2.47}$$

is an optimal trajectory starting from the point x_0. In this way, the Hamilton-Jacobi-Bellman equation can be seen as a *sufficient* condition for existence of an optimal control: if (2.46) has a unique minimizer $\boldsymbol{a}^*(x,t)$ for each (x,t), then we can determine an optimal trajectory.

The Pontryagin Optimality Principle provides the corresponding *necessary* condition for a control plan $\boldsymbol{a}^*$ to be optimal. Define the control Hamiltonian $H : [0,T] \times \mathbb{R}^n \times \mathbb{R}^n \times A \to \mathbb{R}$ by

$$H(t,x,p,a) = \langle f(t,x,a), p \rangle + r(t,x,a), \tag{2.48}$$

for t, x, p, a in their respective domains. Here, as in (2.9, 2.10), we use p as a proxy for $\nabla\phi$. The Pontryagin Optimality Principle states that if $\boldsymbol{a}^*(\cdot)$ is an optimal control plan for (2.33) with corresponding trajectory $\boldsymbol{x}^*(\cdot)$, then there exists a dual trajectory $\boldsymbol{p}^*(\cdot)$, such that

$$\begin{aligned} \dot{\boldsymbol{x}}^*(t) &= \quad \nabla_p H(t, \boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{a}^*(t)), \\ \dot{\boldsymbol{p}}^*(t) &= -\nabla_x H(t, \boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{a}^*(t)), \end{aligned} \tag{2.49}$$

for $t \in [0,T]$ except at the end points where $\boldsymbol{x}(0) = x_0$ and $\boldsymbol{p}^*(T) = \nabla g(\boldsymbol{x}^*(T))$. Further, along the trajectory $(\boldsymbol{x}^*(\cdot), \boldsymbol{p}^*(\cdot))$, we have

$$H(t, \boldsymbol{x}^*(t), \boldsymbol{p}^*(t), \boldsymbol{a}^*(t)) = \inf_{a \in A} H(t, \boldsymbol{x}^*(t), \boldsymbol{p}^*(t), a). \tag{2.50}$$

Supposing the solution ϕ of (2.45) is smooth, we will have $\boldsymbol{p}^*(t) = \nabla\phi(\boldsymbol{x}^*(t), t)$ along the trajectory. Thus (2.49) says, in essence, that optimal trajectories follow the characteristics of the Hamilton-Jacobi-Bellman equation. Ordinarily, this is difficult to use in practice because of the initial-terminal condition. However, in the case of a fixed endpoint problem—where we are given $x_f \in \mathbb{R}^n$ and require that $\boldsymbol{x}(T) = x_f$—we have terminal data for both equations in (2.49) and can integrate the system backwards in time. In this case, (2.49) gives an alternative method for resolving optimal trajectories. We revisit this idea in chapter 4.

2.3.4 The Eikonal Equation in Optimal Control

We return to our example of the Eikonal equation (2.6) and demonstrate how the level set method can be seen as the solution to an optimal control problem.

Consider the state equation

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= v(\boldsymbol{x}(t))\boldsymbol{a}(t), \quad t \in (0, T] \\ \boldsymbol{x}(0) &= x_0, \end{aligned} \tag{2.51}$$

where $v : \mathbb{R}^n \to \mathbb{R}$ is some velocity function and the control variable $\boldsymbol{a}(\cdot)$ takes values in $\mathbb{S}^{n-1} = \{a \in \mathbb{R}^n \,:\, |a| = 1\}$. We could imagine this equation describing the motion of a particle traveling about $\mathbb{R}^n$ and v describes the velocity field determining the speed with which the particle can move through space. The motion is *isotropic*, meaning that at any point, the particle can decide which direction $a \in \mathbb{S}^{n-1}$ to move and the velocity depends only on where the particle is, not on the direction of motion.

Suppose that we would like the particle to end at the origin $x_f = 0$, and thus penalize a trajectory by the distance it ends away from the origin:

$$C[\boldsymbol{x}(\cdot), \boldsymbol{a}(\cdot)] = |\boldsymbol{x}(T)|\,. \tag{2.52}$$

The running cost along the path is zero: $r(t, x, a) \equiv 0$.

Following the above derivation, the value function ψ for this control problem solves the Hamilton-Jacobi-Bellman equation

$$\begin{aligned} &\psi_t + \inf_{|a|=1} \left\{ v(x)\langle a, \nabla\psi\rangle \right\} = 0, \\ &\psi(x, T) = |x|\,. \end{aligned} \tag{2.53}$$

The infimum can be explicitly evaluated. By the Cauchy-Schwarz inequality, we see that

$a = -\nabla\psi/|\nabla\psi|$ and so the equation reads

$$\begin{aligned} &\psi_t - v(x)\,|\nabla\psi| = 0, \\ &\psi(x,T) = |x|\,. \end{aligned} \tag{2.54}$$

Finally, putting $\phi(x,t) = \psi(x,T-t)$, we see that ϕ solves

$$\begin{aligned} &\phi_t + v(x)\,|\nabla\phi| = 0, \\ &\phi(x,0) = |x|\,, \end{aligned} \tag{2.55}$$

which is the same as (2.6) with a slightly modified initial condition.

In the case that $v(x) \equiv 1$, the equation reduces to (2.4) and we can resolve the viscosity solution expicitly. Notice that $\phi(x,t) = |x| - t$ satisfies the equation almost everywhere. However, this function takes negative values, whereas the value function here should remain non-negative since $C[\boldsymbol{x}(\cdot), \boldsymbol{a}(\cdot)] \geq 0$ for any choice of $\boldsymbol{a}(\cdot)$. Separate from the value function interpretation, this non-negativity also results from the comparison principle. We note that the zero function is a subsolution to (2.55), so since $|x| \geq 0$, the comparison principle states that the viscosity solution to (2.55) remains non-negative. Here the viscosity solution is given by $\phi(x,t) = \max\{|x| - t, 0\}$, which makes intuitive sense: if the particle begins at a distance less than t from the origin, then after traveling for time t with velocity 1, it will be able to reach the origin and thus incur zero cost. If the distance is initially greater than t, then traveling for time t allows the particle to reduce the distance (and thus the exit cost) by t, resulting in a cost of $|x| - t$.

A similar analysis can be performed when $v(x)$ is an arbitrary positive function. With this interpretation, we could consider defining $\Omega_t = \{x \in \mathbb{R}^n : \phi(x,t) = 0\}$. In the above example, $\Omega_t = \overline{B(0,t)}$, though this would change if we change the velocity function. Then Ω_t is precisely the set of points that can be reached when traveling outward from the origin for time t. This is essentially the same comment that we made when considering (2.6), though in that case Γ_t represented the boundary of the reachable set, whereas here Ω_t is the whole reachable set. This hints at how one may approach optimal path planning problems using

level set methods.

2.4 Numerical Methods for Hamilton-Jacobi Equations

Finally, to conclude this chapter, we address numerical analysis and simulation of Hamilton-Jacobi equations. We focus our discussion on finite difference methods for the time-dependent Hamilton-Jacobi equation.

2.4.1 Monotone Schemes for the Time-Dependent Hamilton-Jacobi Equation

Because the solutions of Hamilton-Jacobi equations do not remain smooth, simple finite differencing schemes usually fail to capture the intricate dynamics. Thus, effort has been devoted to developing schemes that converge, under grid refinement, to the viscosity solution of the equation. Inspired by similar methods for conservation laws [CM80], Crandall and Lions introduced monotone finite difference schemes for Hamilton-Jacobi equations [CL84]. Such schemes were studied in their earliest form by Godunov [God59], and since the theory of viscosity solutions has become well-established, they have been vastly broadened and extended [ALM10, BO91, BS98, BJ05, HEO87, Sou85]. Oberman [Obe06] gives an overview of monotone schemes, and their extension to so-called degenerate elliptic schemes for second order equations.

For simple exposition, we describe the numerics in two-spatial dimensions on a rectangular grid and we suppress the dependence of H on t, x. The generalization to higher dimensions and the dependence of H on t, x are achieved in obvious ways. Consider the equation

$$\phi_t + H(\phi_x, \phi_y) = 0. \tag{2.56}$$

Define the grid $x_i = i\Delta x$, $y_j = j\Delta y$ for grid parameters $\Delta x, \Delta y > 0$, and discretize time similarly: $t_n = n\Delta t$ for $\Delta t > 0$. Further, let ϕ_{ij}^n denote the numerical approximation to $\phi(x_i, y_j, t_n)$. We assume we have prescribed initial data, meaning that ϕ_{ij}^0 is known for all

(i, j). We consider an update scheme of the form

$$\phi_{ij}^{n+1} = F(\phi_{ij}^n, \phi_{i+1,j}^n, \phi_{i-1,j}^n, \phi_{i,j+1}^n, \phi_{i,j-1}^n), \tag{2.57}$$

that advances the equation from time t_n to time t_{n+1}. More generally, we could allow F to depend on the values of ϕ^n for all nodes in some specified neighborhood of (i, j). As discussed above, one key property of viscosity solutions is that they respect a version of the maximum principle. We would like our discretized equation to respect this as well. Thus, if we have two discrete solutions ϕ^n and ψ^n to (2.56), we require that the update law (2.57) gives

$$\phi^n \leq \psi^n \quad \Longrightarrow \quad \phi^{n+1} \leq \psi^{n+1}. \tag{2.58}$$

To accomplish this, it is sufficient to require that F be non-decreasing in each of its arguments.

In our case, we consider forward Euler time integration, which leads to an update rule of the form

$$\phi^{n+1} = \phi^n - \Delta t H(\phi_x^n, \phi_y^n). \tag{2.59}$$

Since the viscosity solution may develop discontinuities in the derivatives, naive differencing methods will not accurately represent the solution. Thus, to approximate the spatial derivatives of ϕ, we trade the Hamiltonian $H(\phi_x, \phi_y)$ for a numerical Hamiltonian $\hat{H}(\phi_x^+, \phi_x^-; \phi_y^+, \phi_y^-)$, where ϕ_x^+, ϕ_x^- denote the forward and backward difference approximations to ϕ_x respectively, and similarly for ϕ_y^+, ϕ_y^-. The numerical Hamiltonian $\hat{H}$ will deftly combine or choose between the forward and backward differences so as to accurately represent the equation. In order for the update rule

$$\phi_{ij}^{n+1} = \phi_{ij}^n - \Delta t \hat{H}(\phi_x^+, \phi_x^-; \phi_y^+, \phi_y^-)_{ij}^n \tag{2.60}$$

to be consistent, we require that $\hat{H}(\phi_x, \phi_x; \phi_y, \phi_y) = H(\phi_x, \phi_y)$. Under this condition, (2.60) defines a monotone scheme so long as Δt is sufficiently small, and $\hat{H}$ is non-decreasing in the backwards differences and non-increasing in the forward differences: $\hat{H}(\downarrow, \uparrow; \downarrow, \uparrow)$ [Shu07].

The size of Δt is restricted by a CFL condition:

$$\Delta t \left(\frac{H_1}{\Delta x} + \frac{H_2}{\Delta y} \right) \leq 1 \tag{2.61}$$

where H_i is a bound on the derivative of H with respect to its i^{th} argument [OF03]. It is proven in [CL84] that the descrete solution ϕ_{ij}^n produced by a consistent, monotone scheme will converge to the viscosity solution of (2.56) as $\Delta x, \Delta y, \Delta t \to 0$.

There are several choices for the numerical Hamiltonian $\hat{H}$ [OS91]. Perhaps the easiest to implement is the Lax-Friedrichs Hamiltonian

$$\hat{H}^{LF}(\phi_x^+, \phi_x^-; \phi_y^+, \phi_y^-) = H\left(\frac{\phi_x^+ + \phi_x^-}{2}, \frac{\phi_y^+ + \phi_y^-}{2} \right) - \frac{H_1}{2}(\phi_x^+ - \phi_x^-) - \frac{H_2}{2}(\phi_y^+ - \phi_y^-). \tag{2.62}$$

This numerical Hamiltonian uses the centered difference approximations for ϕ_x, ϕ_y, but introduces diffusion on the order of $\Delta x, \Delta y$ to ensure that the solution remains smooth. Accordingly, this Hamiltonian can be seen as a numerical analog to the vanishing viscosity method.

Alternatively, the Godunov Hamiltonian gives a fully upwind, minimally diffusive approximation to (2.56) [BO91]. This Hamiltonian is given by

$$\hat{H}^G(\phi_x^+, \phi_x^-; \phi_y^+, \phi_y^-) = \operatorname*{ext}_{p \in I(\phi_x^+, \phi_x^-)} \operatorname*{ext}_{q \in I(\phi_y^+, \phi_y^-)} H(p, q), \tag{2.63}$$

where

$$I(a, b) = [\min\{a, b\}\,,\, \max\{a, b\}], \tag{2.64}$$

and

$$\operatorname*{ext}_{z \in I(a,b)} = \begin{cases} \min_{z \in [a,b]}, & \text{when } a \leq b, \\ \max_{z \in [b,a]} & \text{when } a > b. \end{cases} \tag{2.65}$$

The Godunov Hamiltonian is typically more difficult to implement. However, in certain cases, these extrema can be determined analytically. For example, in the basic Eikonal

equation (2.4) where $H(\phi_x, \phi_y) = \sqrt{\phi_x^2 + \phi_y^2}$, we have

$$\hat{H}^G(\phi_x^+, \phi_x^-; \phi_y^+, \phi_y^-) = \sqrt{\max\{(\phi_x^-)_+^2, (\phi_x^+)_-^2\} + \max\{(\phi_y^-)_+^2, (\phi_y^+)_-^2\}}, \tag{2.66}$$

where $(X)_+ = \max\{X, 0\}$ and $(X)_- = -\min(X, 0)$.

There are other general numerical Hamiltonians. Otherwise, some numerical Hamiltonians have been suggested that apply when H has a specific form. For example, in the original level set paper, Osher and Sethian suggest an upwind numerical Hamiltonian that applies when $H(\phi_x, \phi_y) = h(\phi_x^2, \phi_y^2)$, for some function h [OS88].

In general, monotone schemes can be only first order accurate [God59]. However, to attain higher order accuracy, Osher and Shu suggest (weighted) essentially non-oscillatory (ENO) approximations to the derivatives ϕ_x, ϕ_y, in concert with one of these monotone Hamiltonians [OS91, Shu07, Shu98]. For example, the second-order ENO approximations to ϕ_x are given by

$$\begin{aligned}
(\phi_x^+)_{i,j} &= \begin{cases} -\frac{\phi_{i+2,j} - 4\phi_{i+1,j} + 3\phi_{i,j}}{2\Delta x}, & \text{if } |(\phi_{xx})_{i,j}| \geq |(\phi_{xx})_{i+1,j}|, \\ \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x}, & \text{if } |(\phi_{xx})_{i,j}| < |(\phi_{xx})_{i+1,j}|, \end{cases} \\
(\phi_x^-)_{i,j} &= \begin{cases} \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x}, & \text{if } |(\phi_{xx})_{i,j}| \geq |(\phi_{xx})_{i-1,j}|, \\ \frac{\phi_{i-2,j} - 4\phi_{i-1,j} + 3\phi_{i,j}}{2\Delta x}, & \text{if } |(\phi_{xx})_{i,j}| < |(\phi_{xx})_{i-1,j}|, \end{cases}
\end{aligned} \tag{2.67}$$

where $(\phi_{xx})_{i,j}$ denotes the usual second-order centered approximation to ϕ_{xx} at the node (i, j). The philosophy here is that we can avoid spurious oscillations near kinks by choosing the one-sided approximation to the derivative which does not cross the kink. When using these higher order approximations, the overall scheme is no longer monotone, but the monotonicity is only violated by an error of size $O(\Delta x, \Delta y, \Delta t)$, so convergence to the viscosity solution is maintained and abitrarily high-order accuracy can be achieved in regions where ϕ remains smooth [OS91].

2.4.2 Numerical Diffusion and the Level Set Method

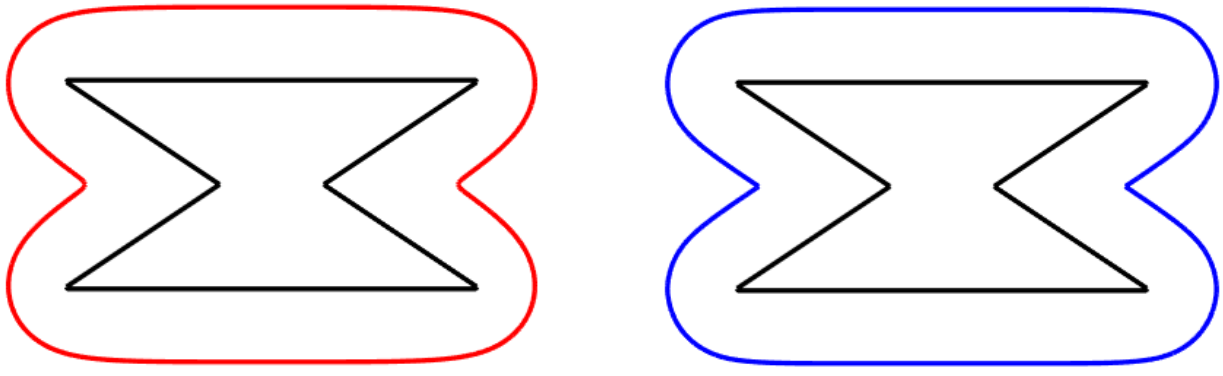
While the Godunov method is usually more difficult to implement than Lax-Friedrichs or other monotone schemes, we argue that it is highly preferable in the case of level set equations. The strategy of the Lax-Friedrichs method is to add diffusion on the order of $\Delta x, \Delta y$ so as to smooth the solution. While this will still provide an appoximation to the viscosity solution, the smoothing is often undesirable in applications where the geometry of the level sets is important.

We demonstrate this undesirable smoothing effect by looking at two examples of the Eikonal equation. Recall, the equation is given by

$$\begin{aligned} \phi_t + v(x)\,|\nabla\phi| &= 0, \\ \phi(x,0) &= \phi_0(x), \end{aligned} \tag{2.6}$$

where, in the level set application, we typically specify that ϕ_0 is the signed distance to the initial curve Γ_0 that will be evolved with level set flow.

In the case that Γ_0 has very sharp edges, the smoothing effect of the diffusion can cause undesirable distortion. This is seen in fig. 2.8. Here the initial contour is shown in black, and the velocity is constant $v \equiv 1$ so the level sets should represent isocontours of equal distance from the initial contour. This means that near the sharp concavity, the level set should



(a) Lax-Friedrichs level set.

(b) Godunov level set.

Figure 2.8: Level sets propagating outward from an initial contour (black) with sharp convexities and concavities. Near the concavities, the Lax-Friedrichs level set (left, red) is overly smoothed, failing to mimic the sharp point. Near the convexities, the Lax-Friedrichs level set did not evolve "fast enough"; it is closer to the initial contour at those points than at others. The Godunov level set (right, blue) more effectively captures the geometry.

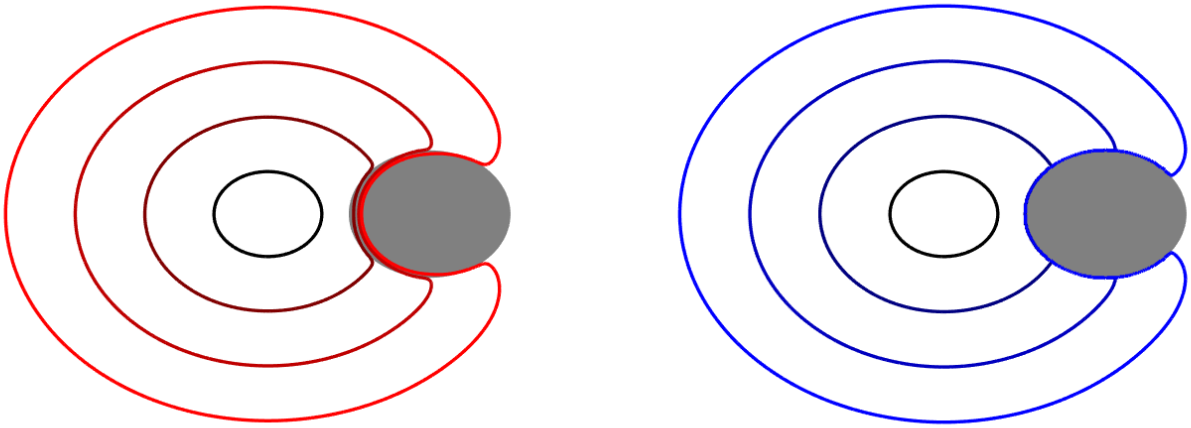
maintain the sharp point, which is seen to happen when using the Godunov Hamiltonian in fig. 2.8b. However, due to the smoothing, the Lax-Friedrichs Hamiltonian is unable to maintain the point as seen in fig. 2.8a. Likewise, the Godunov Hamiltonian more effectively captures the behavior near the sharp convexities, where the Lax-Friedrichs level set seems to flatten out rather than maintaining equidistance from the initial contour. Examples like this are very relevant to in applications like crystal growth [CMK01, GFC03, JT96, TZ06] where we sharp edges form naturally, or image processing [JZN12, LHD11, QWH07, VC02] where the level sets need to closely mirror the shapes in images.

Our second example reveals a failure in the Lax-Friedrichs method when there are regions of zero velocity. In this example, we will let Γ_0 be the unit circle, and suppose that velocity is 1 everywhere except for a circular region where velocity is zero (pictured in grey in fig. 2.9). In fig. 2.9a, we notice that the level set created using the Lax-Friedrichs method do not entirely stop when they reach the region of zero velocity. In that region, the numerical diffusion is noticeable since the local advection is absent. By contrast, the Godunov level sets in fig. 2.9b do stop in regions of zero velocity. This is again a concern in image segmentation where, for example, the level set may be used to map out blood vessels in a CT scan [LHD11]. As we will see, it is also a major concern in path-planning where the level sets represent the boundary of the reachable set at different times and regions of zero velocity can model impassable obstacles. Especially in the case that the obstacles are fairly thin relative to the grid resolution, the numerical diffusion can seriously skew results since it can cause the level sets to diffuse through boundaries.

2.4.3 Other Numerical Methods for Hamilton-Jacobi Equations

We have briefly covered grid-based numerical schemes for the time-dependent Hamilton-Jacobi equation. These will suffice for our computations, but there are several other methods for solving Hamilton-Jacobi equations that bear mentioning.

As stated above, in the case of monotonically advancing fronts, one can opt for a steady-state formulation (2.8) of the level set equation. A similar phenomenon occurs with the



(a) Lax-Friedrichs level sets.

(b) Godunov level sets.

Figure 2.9: Level sets propagating outward from the unit circle (black), with a region of zero velocity (grey). The Lax-Friedrichs level sets (left, red) continue to slowly propagate through the circle due to numerical diffusion. The Godunov level sets (right, blue) stop at the circle.

optimal control problem when the state dynamics (f in (2.31)) and running cost (r in (2.32)) do not depend explicitly on t. In this case, the value function can be taken to be independent of t and can be shown to solve a steady-state Hamilton-Jacobi-Bellman equation. Thus there has been some effort devoted to developing schemes for steady-state Hamilton-Jacobi equations. These can be broadly divided into fast marching schemes and fast sweeping schemes. Fast marching schemes can be seen as a continuous generalization of Dijkstra's algorithm, navigating through the domain and assigning values to grid nodes as the front passes them [Tsi95, Set99, SV01, SV03, AM06, AM09]. Fast sweeping schemes rely on a Gauss-Seidel type iteration to monotonically update values at grid nodes [KOQ04, KOT05, TCO03, Zha05]. Steady state methods are less general, but have the advantage of lower computation burden due to the removal of the time dimension. For level set equations, if efficiency is an issue and a time-dependent formulation is necessary, one can use so-called narrow band method where computations are only performed near the propagating front [AS95, LDM14, Min04, PMO99].

Recently, there has been increased interest in high dimensional Hamilton-Jacobi equations due to applications like optimal transport [San15] and mean field games [BFY13, GLL11]. Such problems suffer from the curse of dimensionality [Bel54, Bel61], meaning that grid-based methods are computationally infeasible. Accordingly, there has been much effort devoted to

developing algorithms that break the curse of dimensionality. Many such methods are based on the Hopf-Lax or Lax-Oleinik formulas [Eva10], and use optimization routines such as the primal-dual hybrid gradient method [CP11] or the split Bregman method [GO09] to resolve the values of the solution at individual points [DO16, LCO18, CDO19]. Alternatively, deep learning methods can be applied to solve the equations [HJW18, NGK19, Rai18].

CHAPTER 3

A Level Set Model of Deforestation in Protected Areas

Environmental crime in protected national parks is a concern to authorities around the world. National parks often serve as hotspots for illegal logging and animal poaching, henceforth referred to as illegal extraction. In recent years, scientists have aided law enforcement agencies in the prevention of such crimes by building models to describe deforestation, track animal movement, and predict adversarial behavior of criminals among other things. Leader-Williams and Milner-Gulland examined a case study of poaching in the Luangwa Valley, Zambia and concluded that increasing detection rates was a more effective deterrent to environmental crime than increasing severity of punishment [LM93]. Thus models that can help improve detection rates of environmental criminals are a useful tool for patrollers of protected areas.

3.1 Introduction & Previous Work

An important model of illegal extraction was given by Albers [Alb10]. They describe illegal extraction as a continuous spatial game between patrol units (henceforth known as "patrol") and criminals (henceforth known as "extractors"). In the Albers model, extractors maximize their expected profit by trading off costs of penetrating further into a protected region (increasing effort required and risk of capture) against increased benefits of extracting further from the boundary of the protected region. Albers' model was formulated as a Stackelberg game; an adversarial game with a defender and many attackers with perfect information about the defender's strategy. Albers gave some qualitative results in a very simplified situation where the protected region was circular, and all quantities were radially symmetric.

Johnson, Fang and Tambe proved an additional result regarding optimal patrol strategies for this simple case [JFT12]. Kamra et al. further extended and improved the model, retaining the continuous state space, removing the assumption that trees are homogeneously distributed, considering collisions between law enforcement and extractors and calculating optimal or near-optimal patrol strategies using machine learning techniques [KGF18]. All of this work considered a circular protected region, wherein the goal of the extractors is to move toward the center of the region.

Subsequently, several models for illegal extraction (both deforestation and animal poaching) in protected regions were developed using discrete, rather than continuous, methods [FJT13, FNP17, KFD15, KFG17, MTK16]. These algorithms incorporate different methods for modeling human behavior and different ways to treat the protected region. Kar et al. used repeated Stackelberg games to attempt to understand the evolution of attackers strategies, but did not consider realistic spatial domains [KFD15]. Fang et al. developed the PAWS algorithm, considering detailed terrain and spatial information by describing the protected region as a series of nodes connected by edges corresponding to natural pathways through the region (for example along rivers or walking trails) [FNP17]. This algorithm has been deployed in Queen Elizabeth National Park (QENP) in Uganda, and in Malaysian forests. Taking another approach, Kar et al. used machine learning techniques to construct a model for predicting poaching attacks, again deploying the model for a field test in QENP, Uganda [KFG17]. In addition to the patrol strategy, McCarthy et al. are able to distinguish between different types of patrol teams, and their algorithm was deployed in Madagascar [MTK16].

In this work, we generalize the model of Albers [Alb10] so that it is applicable to realistic protected regions, can include the effects of terrain and geometry, and can be executed for real protected regions. The most important mathematical technique that enables this extension is the level set method [OS88]. Indeed, as hinted in the previous chapter, the level set method and Hamilton-Jacobi equations can be used to model movement in a variety of contexts. Sethian and Vladimirsky [SV01, SV03] discuss a level set method for optimal travel on manifolds. Martin and Tsai suggest a steady-state HJB formulation for deter-

mining optimal paths on manifolds which are represented computationally by unstructured point clouds [MT]. In application, Dubins[Dub57], Reeds and Shepp [RS90] and later Takei and Tsai [TTS10, TT13], modeled movement of simple cars through terrain with obstacles using methods from control theory and Hamilton-Jacobi-Bellman equations (we discuss this further in chapter 6). Similar methods have been used to determine reachable sets for aircraft autolanders [BMO07], and to model human movement in adversarial reach-avoid games [CLS19, ZDH18]. Thus there is a precedent for using Hamilton-Jacobi type equations to model movement throughout domains. However, as discussed above, attempts to model environmental crime have either required overly restrictive assumptions regarding radially symmetry and geometry of the protected region, or have been discrete in nature. This work serves to fill a gap in the literature. We suggest a continuum model for environmental crime that is still able to account for realistic terrain information, and bridges the divide between environmental crime modeling and Hamilton-Jacobi formulations for modeling human movement.

This chapter is structured as follows. In section 3.2, we describe the model of Albers [Alb10] in some detail. In section 3.3, we show how the level set method can be applied to model movement through regions with terrain and describe our model for illegal extraction from protected regions. Further, we provide an algorithm to calculate the regions through which extractors will pass. In section 3.4, we present results and discuss their implications towards both patrol strategies and our modeling approach. Lastly, in section 3.5, we make conclusions and identify potential directions for further research.

We note that this work has been previously published in the SIAM Journal on Applied Mathematics [AFJ19]. A full copyright acknowledgement is given at the end of the chapter.

3.2 The Illegal Deforestation Model of Albers

Albers [Alb10] presented a model for illegal extraction from protected regions that includes spatial effects. They use a Stackelberg game in a circular region between defenders and extractors. In their model, an extractor gains benefit and incurs cost based on the depth d

into the region where they choose to extract. The benefit is associated with the extraction point, and is modeled as a concave increasing function $B(d)$. Cost is based on the trip distance, and is modeled as a quasiconvex increasing function $C(d)$, but no specific functions are given. This concavity and convexity is an expression of diminishing returns: the next step will provide less benefit and incur more cost than the previous step.

The extractor's profit associated with extracting at depth d is then

$$P(d) = B(d) - C(d). \tag{3.1}$$

To deter the extractors, the patrol specify a strategy $\psi(d)$, representing the marginal detection probability of the extractors. The probability that the extractor is detected is then $\Psi(d) = \int_0^d \psi(r)dr$ over the trip out of the protected region. If the illegal extractor is caught, any resources they've extracted are confiscated and they must leave empty-handed. The extractor's expected profit can therefore be written as

$$P(d) = (1 - \Psi(d))B(d) - C(d). \tag{3.2}$$

We assume extractors have perfect information of the patrol strategy (they know the function $\Psi(d)$) and hence their task is to find the optimal distance d^* to penetrate into the protected region so as to maximize their profit. Simultaneously, the patrollers task is to pick the patrol strategy $\psi(d)$ that minimizes d^*. The region that is not entered by the extractors is referred to as the *pristine area*. Figure 3.1 illustrates the model, showing the pristine area where extractors are never present and the outer region that they pass through while traveling into or out of the protected region.

The key simplification here is that the model is entirely one-dimensional: all quantities depend only on the depth d into the forest. In the original analysis, Albers suggested several different types of patrol strategy including patrols that are homogeneous throughout the region, patrols that are focused on an annulus near the boundary, or Dirac-mass patrol concetrated at a single depth [Alb10]. Johnson, Fang and Tambe analyzed the same model

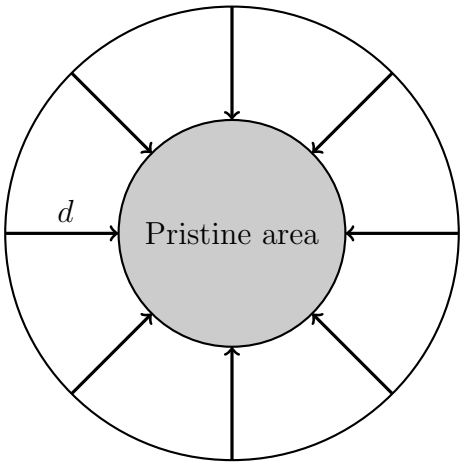
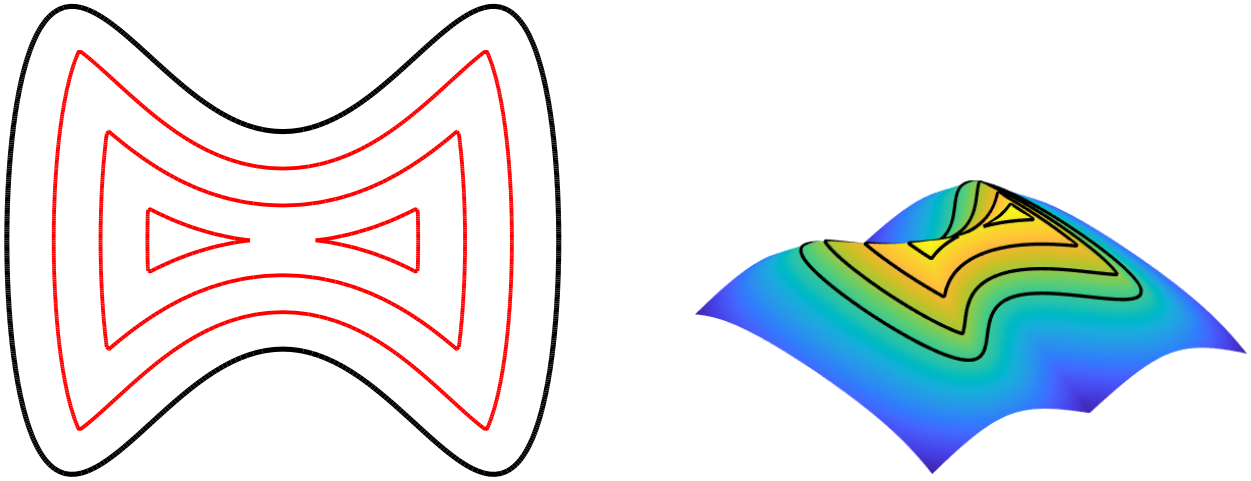


Figure 3.1: Illustration of the Albers [Alb10] model, where extraction occurs at depth d in a circular region. Reproduction of Figure 2 from [JFT12].

[JFT12]. They point out that even in the absence of patrol, if $C(d)$ grows quickly enough relative to $B(d)$, there will be a pristine area. However, they also note that in certain cases, homogeneous and boundary patrols can result in an empty pristine area. Finally, they prove that the optimal patrol strategy will focus on an annulus and will increase toward the outer edge of the annulus. We consider some of these patrols later in this chapter.

3.3 A Level Set Model for Illegal Deforestation

As described in section 2.1, the level set method can be used to model fronts propagating with prescribed normal velocity. Here, as in [Alb10], we envision environmental criminals moving inward from the boundary of a protected region. Since we are modeling movement within the region, we want the level sets to evolve inward from the boundary $\Gamma = \Gamma(0)$, and so we set the initial condition ϕ_0 to be the signed distance to Γ, taking positive values inside Γ and negative outside. Thus, in contrast to fig. 2.1, translation of ϕ_0 down the z-axis will cause the level sets to shrink. This is illustrated in Figure 3.2. In this case, Γ is defined by $(x(\theta), y(\theta)) = (\cos(\theta), \sin(\theta) + \sin(3\theta)/2)$ and shown in black in fig. 3.2a. The level set velocity is constant, so the series of red contours in fig. 3.2a represent sets of points accessible after traveling a certain time from the boundary. Figure 3.2b shows the graph of the level set function and with the same level sets overlaid on the surface.



(a) Contours of equal travel time (red) from the boundary of a region (black).

(b) The level set function ϕ with level sets (black).

Figure 3.2: Using the level set method to find contours of equal travel time inward from the boundary.

3.3.1 Modeling Extractors' Behavior

Our model for the actions of illegal extractors committing crimes in protected regions is based on the Albers model [Alb10]. However, we would like to remove what we deem are overly restrictive assumptions. We are interested in real national parks and hence the protected region will not be symmetric and the benefit and cost functions could be arbitrary, rather than depending only on depth. The extension of the Albers model to arbitrary geometry comes with a few challenges.

In [Alb10], the profit is maximized on a ring with some radius and extractors will enter from any point on the boundary of the region to travel to the maximum profit ring. Without radial symmetry, the profit will, in general, be maximized at an isolated point rather than along a closed curve, and correspondingly there will be a single optimal route from this point to the boundary of the protected region. Hence illegal extractors only ever occupy a set of points of measure zero. In our model, we resolve this issue by assuming that extractors will tolerate extracting anywhere where the profit is close to the maximum possible profit.

Another issue is that the cost function of Albers does not generalize easily to irregular geometries. Instead of using a predefined formula based on distance, we constructively determine the cost by considering the effects that would detract from the final profit with the help of the level set method. Given terrain and elevation information for a real protected

park, the paths taken by extractors will not be simple straight lines. Therefore it is also necessary to find the paths extractors will follow to exit of the protected region. In the ensuing section we describe an algorithm to find the pristine and non-pristine regions in a protected area with arbitrary shape, terrain, benefit distribution and patrol strategy.

3.3.2 Problem Description

Let $\Omega \subset \mathbb{R}^2$ represent the area that needs to be protected and let $\psi(x, y)$ be the patrol density function chosen by the patrol. The patrol budget E is defined as

$$E = \int_{\Omega} \psi(x, y)\, dx\, dy. \tag{3.3}$$

The budget measures how many patrolling resources can be allocated to protect the domain Ω. Effectively, the budget scales ψ.

Extracting at a position (x, y) gives the extractor an amount of benefit $B(x, y)$. For example, the benefit could depend on the quantity, value, or species of trees. A priori, we make no assumptions about B except that it is known. We denote by $C(x, y)$ the expected cost associated with extracting at position (x, y). Our cost function will be based on two factors: the effort involved in traveling from the extraction point to the boundary of the protected region, and the risk of being caught by patrols while inside the domain. The further into the protected region the extractors penetrate, the higher the cost as they must expend more time/energy traveling, and are more likely to be captured by the patrol.

This highlights a difference between our model and Albers': we include the effect of the patrol directly in the cost function, whereas they include it as a modification to the benefit. In our model, the profit an extractor expects from extracting at a point (x, y) in the protected region is simply $P(x, y) = B(x, y) - C(x, y)$. We assume the extractors accept a position to extract if the profit is within some tolerance of the maximum obtainable profit. Finally, we determine the paths that extractors will take when leaving the protected area (where they can be captured by the patrol). We assume the extractors will always take an optimal path from the extraction point to the boundary, and can only be caught after they have finished

extracting and started heading back to the boundary (since no crime has been committed before they have extracted).

One feature that arises from our model is that in order to achieve high profit, extractors should only enter from some sections of the boundary of the protected region. In practice however, extractors start their trips from villages near the edge of the park boundary. This means they may have to travel some distance to get to a position on the boundary of the region which leads to high profit. We do not include the extra costs of travel outside the protected region or travel from the boundary to the extraction location. Although we do not do so in this work, it is possible to model some of these extra costs. For example, an extra cost term applied on the boundary could represent the cost of travelling from a village to that point, and the cost of the inward journey from the boundary to the extraction point can be calculated relatively simply with a level set calculation that ignores patrol strategy.

3.3.3 Calculating the Profit Function

Since the benefit function is known in advance, it remains to calculate the expected cost function. As explained in the previous section, the cost function depends on many factors, including the optimal route to a given point. The level set method avoids most of the difficulties one might expect when calculating optimal paths from all points inside the protected region to the boundary. We use the level set method to find contours of equal cost in the protected region. As the level sets evolve, individual points along the contour follow paths of minimal cost, where the "cost" accounts for both the physical effort to traverse the region and the risk of being captured by patrollers.

The expected cost associated with extracting at a given point (x_0, y_0) is calculated with level set equation

$$\begin{aligned} &\phi_t + \frac{1}{1/v(x,y) + \alpha\psi(x,y)B(x_0,y_0)} |\nabla\phi| = 0, \\ &\phi(x,y,0) = \phi_0(x,y), \end{aligned} \tag{3.4}$$

where ϕ_0 is the signed distance to $\Gamma(0) = \partial\Omega$, taking positive values inside Ω.

As this equation advances, the zero level sets $\Gamma(t)$ represent contours of equal cost $t > 0$. The cost associated with (x_0, y_0) is defined to be the unique $C > 0$ such that $(x_0, y_0) \in \Gamma(C)$. That is, the cost function $C(x_0, y_0)$ is implicitly defined by $\phi(x_0, y_0, C(x_0, y_0)) = 0$. This is illustrated in fig. 3.3.

The important term from a modeling perspective is the coefficient

$$V(x, y; B(x_0, y_0)) = \frac{1}{1/v(x, y) + \alpha\psi(x, y)B(x_0, y_0)} \tag{3.5}$$

of $|\nabla\phi|$ in (3.4), which determines the level set velocity. Here $v(x, y)$ represents the travel velocity, which we use as a proxy for travel cost apart from any consideration of patrol. This could account for travel speed, terrain difficulty or a number of other physical considerations. Increasing $v(x, y)$ will increase the level set velocity meaning that cost decreases. Also in the denominator is the term $\psi(x, y)B(x_0, y_0)$. This will cause cost to increase in regions that are more densely patrolled, and accounts for the extractors unwillingness to be captured. We

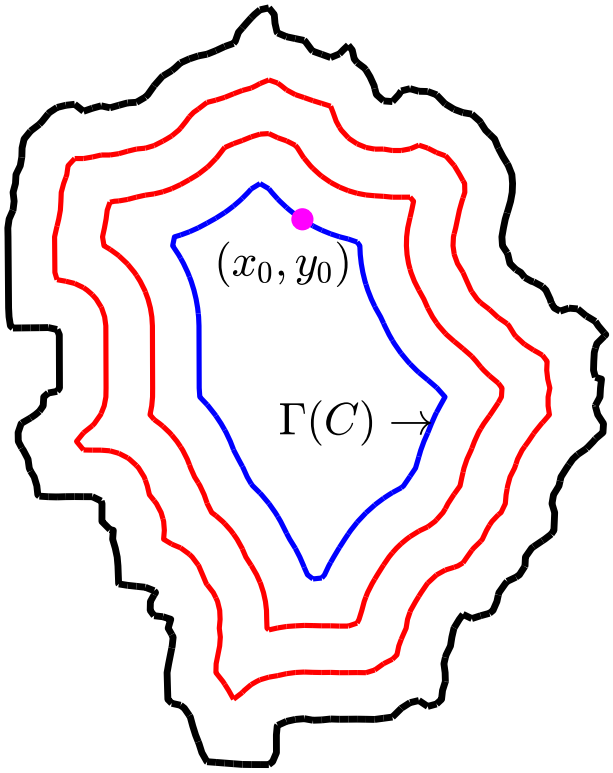


Figure 3.3: Evolve level sets in from the boundary until the $C > 0$ such that $(x_0, y_0) \in \Gamma(C)$. This C is the cost associated with the point (x_0, y_0).

scale the patrol density by $B(x_0, y_0)$ because we assume that if an extractor has gained a larger benefit, they will be more averse to being captured. Increasing $\psi(x, y)$ will decrease the level set velocity meaning that cost increases. The constant α is a dimensional parameter that translates between travel difficulty and aversion to being captured, so we refer to it as the *risk aversion* parameter. Higher α indicates that extractors are willing to travel through more treacherous terrain to avoid patrol. Alternatively, lower α signifies that extractors will assume more risk of being captured so as to travel more easily. An important note is that $V(x, y; B(x_0, y_0))$ includes information about the extraction point (x_0, y_0) via the benefit $B(x_0, y_0)$. This means that the cost function must be calculated separately for each extraction point. Thus we should actually write $\phi(x, y, t; B(x_0, y_0))$ to express that this is the level set function corresponding to the extraction point (x_0, y_0), but we will supress this notation.

3.3.4 The Walking Velocity Function

It remains to decide on a velocity function $v(x, y)$. Tobler gave an early formula for human walking speed based on slope [Tob93]. More recently, Irmischer and Clarke [IC17] gave an improved formula that they claim more accurately predicts travel times for humans walking on roads,

$$v = 0.11 + \exp\left(-\frac{(100s + 2)^2}{1800}\right), \tag{3.6}$$

where $100s$ is the grade in percent, and v is the corresponding speed in meters per second. This formula agrees well with experimental results given in [IC17], but has a drawback for our model, that the speed does not vanish as the slope goes to infinity because they only considered grades up to 100% (signifying a 45° incline/decline). We hence modify the given equation so that the speed drops to zero as the slope becomes more extreme and use

$$v = 1.11 \exp\left(-\frac{(100s + 2)^2}{2345}\right), \tag{3.7}$$

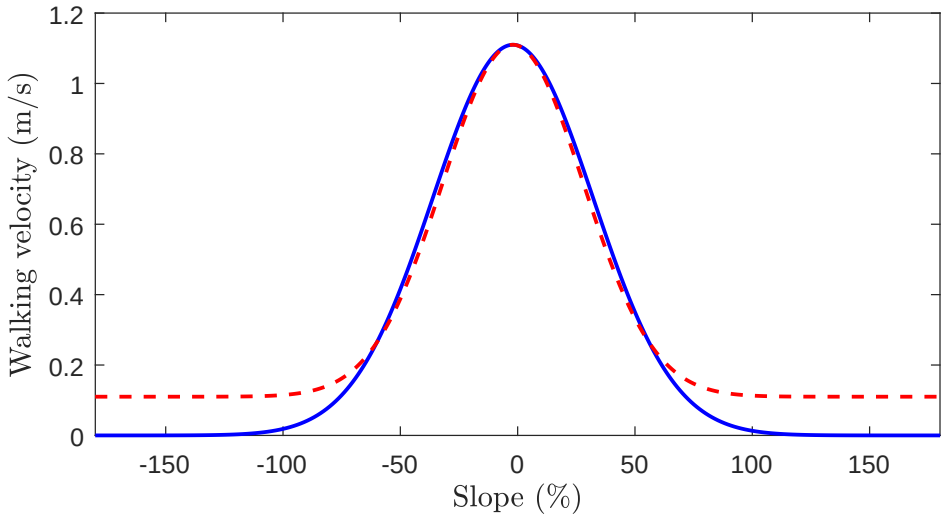


Figure 3.4: Our velocity function eq. (3.7) (solid blue line), and the Irmischer & Clarke function (3.6) (dashed red line). Our velocity function decays to zero for high slopes while the function suggested by [IC17] does not.

which matches the maximum speed, gives similar results for grades less than 60% and decays to zero for more extreme slopes as can be seen in fig. 3.4.

This walking velocity could be easily modified to further account for terrain types, roads, rivers or other geographical and topographical features. We note that the exact form the function is not a crucial piece of the model, and other walking velocity functions could be used, though for the sake of realism, the general shape should be maintained: the maximum velocity should be on the order of 1 and should occur at near zero slope, and the velocity should decay to zero as slope becomes large.

Another note is that the walking speed $v(x, y)$ that appears in the level set equation (3.4) is a function of position only and does not depend on walking direction. This is a simplification since walking speed does depend on walking direction (for example walking down a hill is faster than walking up a hill, or walking along a ridgeline is faster than climbing down the ridge). Assuming v depends only on position reflects the fact that walking in any direction on steeply sloping ground is slow even if the slope in the direction of travel is moderate. We take the maximum slope as the input to the velocity function (that is, for an elevation profile $E(x, y)$, we set $s = ||\nabla E||$). It is possible to account for walking direction with a level set formulation which we discuss in chapter 4, but doing so requires an anisotropic control-theoretic formulation.

3.3.5 The Expected Cost Algorithm

As noted above, the level set equation defining the cost function depends on the extraction point through the benefit function. Thus to compute the cost at different points, one would need to run individual level set simulations for each point. We can partially avoid this by computing the profit function using the following algorithm, which is illustrated in fig. 3.5

1. Since the cost level set equation (3.4) depends on the particular benefit level where extraction takes place, we must solve it for all possible values of B. We pick N benefit values in $[\min_\Omega B, \max_\Omega B]$ and evaluate the cost function using each benefit value B_i, by performing steps 2-4 for each B_i.

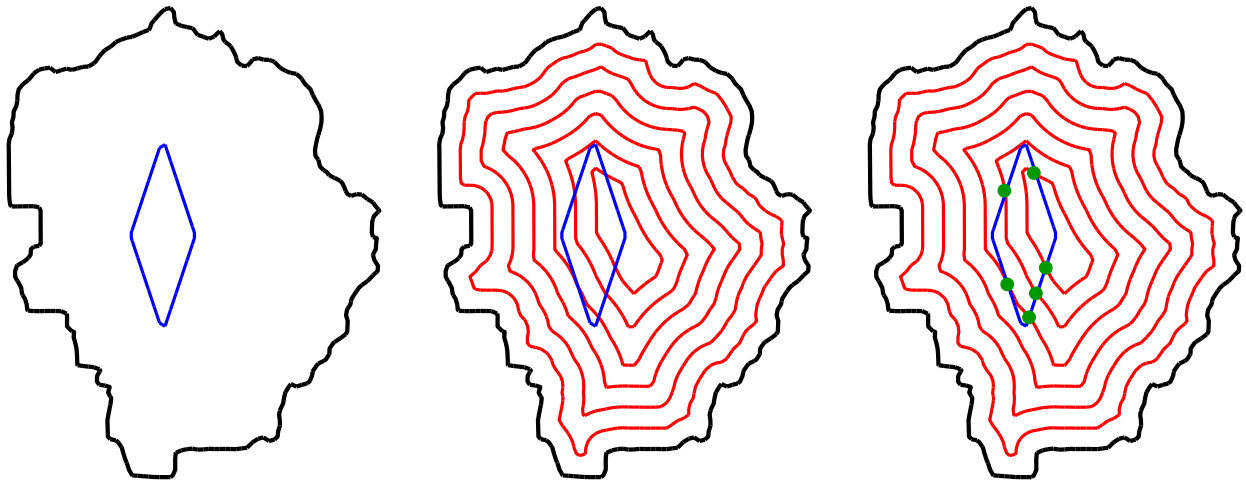
2. For a given B_i, first find the contour(s) of points where $B(x, y) = B_i$. In fig. 3.5a, the black contour is the boundary of the domain, and the blue contour is an equal benefit contour with the same benefit B_i.

3. Evolve the expected cost level set function with the benefit B_i using eq. (3.4). In fig. 3.5b, the red contours are the equal-cost level set contours. Each of the contours represents a different cost value.

4. The cost level sets only apply to the points whose benefit has been used to calculate the level sets. That is, we have calculated the cost assuming the benefit gained is B_i, so our results are only valid at the points (x, y) such that $B(x, y) = B_i$. Therefore we find the intersections between the cost level contours $\phi(x, y, C) = 0$ and the benefit contour $B(x, y) = B_i$. At these intersections (marked in green), we can calculate $C(x, y)$.

5. Repeat steps 2-4 for each benefit level B_i, until values of $C(x, y)$ are known throughout the region, and interpolate the cost value to other points.

By using this algorithm, we can calculate the cost function with N level set simulations, at the expense of discretizing the benefit function into N levels. The algorithm produces a collection of cost functions $C(\,\cdot\,; B) : \Omega \to [0, \infty)$, parametrized by $B \in [\min_\Omega B, \max_\Omega B]$, where $C(x, y; B)$ represents the cost of extracting at (x, y) if the benefit the extractor has



(a) Begin by choosing a contour (blue) of fixed benefit value.

(b) For this benefit value, evolve cost contours (red) in from the boundary.

(c) At intersection points (green), one knows the benefit and cost and thus can calculate the profit.

Figure 3.5: Determining the profit function throughout the region. These images correspond to steps 2, 3 and 4 in the expected cost algorithm (section 3.3.5) respectively.

gained is B. Since $B(x, y)$ represents the benefit accrued at the point (x, y), the cost to extract is given by $C(x, y; B(x, y))$. This value represents the minimal cost required to travel from the extraction point (x, y) to the boundary of the protected region, considering both the physical effort required to travel and the risk of being captured. Once this function is obtained, we can calculate the profit for the extractors $P(x, y) = B(x, y) - C(x, y; B(x, y))$. In what follows, we will suppress some notation, writing only $C(x, y)$ at times, and leaving the dependence of the cost on the benefit implicit.

3.3.6 Finding the Pristine Region

After resolving the expected profit function, the next step is to find the pristine region. In [Alb10], the pristine region is the portion of the circular forest between its center and the ring of maximum profit. In general, however, the pristine region must be defined differently. We assume that extractors will extract anywhere where the profit is close to the maximum possible profit. Denote P_{max} as the maximum expected profit, and $(1-\varepsilon)P_{max}$ as the lowest expected profit that the extractor will accept for some tolerance ε. We also assume that the extractors will take the optimal path away from their extraction point to the boundary of the region. Thus the pristine region is determined using the following algorithm.

1. Find the equal profit contour corresponding to $(1-\varepsilon)P_{max}$. The extractors will only extract within this high-profit region.

2. Pick points inside the high-profit region uniformly at random, and calculate the optimal path from that point to the boundary of the region. The optimal path can be found easily by solving the gradient descent equation

$$\dot{\boldsymbol{x}}(t) = -\nabla C(\boldsymbol{x}(t); B(x_0)), \tag{3.8}$$

   where $\boldsymbol{x}(0) = x_0$ and $C(\,\cdot\,; B(x_0))$ is the cost function wthat was calculated using the benefit level $B(x_0)$. If we advance this equation until $C(\boldsymbol{x}(T); B(x_0)) = 0$, then $\boldsymbol{x}(\cdot)$ represents the minimal cost path from x_0 to the exterior of the protected region.

3. The non-pristine region includes the high-profit region, and all points within some (small) distance of one of the optimal paths found in the previous step. The rest of the region is pristine since it is not traversed by extractors.

3.3.7 Metrics for Measuring Patrol Effectiveness

We suggest two metrics for measuring patrol effectiveness. First, in analogy with [Alb10], we consider the pristine area ratio

$$PA = \frac{\int_\Omega \chi(x,y)\,dx\,dy}{\int_\Omega dx\,dy}, \tag{3.9}$$

where $\chi(x,y)$ is an indicator function which is 1 in the pristine region and 0 in the non-pristine region.

Second, the benefit function $B(x,y)$ can be interpreted as a map of commodities that the patrol would like to protect. Thus we could weight the pristine area by this function and consider the protected benefit ratio

$$PB = \frac{\int_\Omega B(x,y)\chi(x,y)\,dx\,dy}{\int_\Omega B(x,y)\,dx\,dy}. \tag{3.10}$$

Likewise, other functions besides $B(x, y)$ could be used to express relative importance of protecting certain areas more than others.

3.3.8 Equivalence of Our Model and Albers'

Although our model is more sophisticated than that of Albers [Alb10], we note that in certain simple cases the two coincide. Consider an instance of Albers' model in a circular forest of radius 1 with a homogeneous patrol strategy with budget 1, which means that $\psi \equiv 1/\pi$. Further, suppose the benefit function is given by $B(d) = 2d$ and the cost function is given by $C(d) = d$. In this case, the profit for the extractors is given by

$$P(d) = (1 - \Psi(d))B(d) - C(d) = 2\left(1 - \frac{d}{\pi}\right)d - d = \left(1 - \frac{2d}{\pi}\right)d. \tag{3.11}$$

We achieve the same result using our model. Setting $B(d) = 2d$ in [Alb10] is equivalent to $B(x) = 2(1 - |x|)$ in our model, since $1 - |x|$ measures the depth of x into the forest. Setting $C(d) = d$ in [Alb10] is equivalent to setting $v(x, y) \equiv 1$ in our model. Likewise, we set $\psi(x, y) = \frac{1}{\pi}$, and $\alpha = 1$. With these parameters, equation (3.4) reads

$$\phi_t + \frac{1}{1 + 2(1 - |x_0|)/\pi} |\nabla\phi| = 0. \tag{3.12}$$

if we are calculating the cose at the point x_0. The initial condition is given by the signed distance to the boundary $\phi_0(x) = 1 - |x|$. Since the coefficient of $|\nabla\phi|$ is independent of x, and since $|\nabla\phi_0| = 1$ almost everywhere, we see that

$$\phi(x, t) = 1 - |x| - \frac{t}{1 + 2(1 - |x_0|)/\pi}, \tag{3.13}$$

is an almost everywhere solution to (3.12). Indeed, a computation like that performed for the Eikonal equation in section 2.2.3 shows that this $\phi(x, t)$ is the viscosity solution of (3.13).

The cost at x_0 is then implicitly defined by $\phi(x_0, C(x_0)) = 0$ which gives

$$1 - |x_0| - \frac{C(x_0)}{1 + 2(1 - |x_0|)/\pi} \quad \Longrightarrow \quad C(x_0) = \left(1 + \frac{2(1 - |x_0|)}{\pi}\right)(1 - |x_0|), \tag{3.14}$$

whereupon

$$P(x_0) = B(x_0) - C(x_0) = 2d(x_0) - \left(1 + \frac{2d(x_0)}{\pi}\right) d(x_0) = \left(1 - \frac{2d(x_0)}{\pi}\right) d(x_0), \tag{3.15}$$

where $d(x_0) = 1 - |x_0|$. This formula agrees with (3.11).

This equivalence holds slightly more generally. For example, if the patrol is piecewise constant and the cost is linear in [Alb10], then our model will produce the same profit function. If the patrol density and/or cost function are more complicated, then ϕ_t will not be constant, and the PDE solution will be more complicated. Our approach to calculating the cost and benefit functions is therefore equivalent to the approach given by [Alb10] in simple scenarios but is capable of modeling much more general cases.

3.3.9 Control Theoretic Formulation

While we have constructed our model purely employing a level set approach, it can be noted that the extractors are solving a control type problem, and thus we can phrase the model in the language of control theory. Indeed, consider the state dynamics for an extractor given by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{a}(t), \quad t > 0, \tag{3.16}$$

where the control variable $\boldsymbol{a}(\cdot)$ represents the walking direction. If the extraction point is $x_0 \in \Omega$, we can define the cost for a walking path to be

$$C(x; B(x_0)) = \inf_{\boldsymbol{a}(\cdot)} \left\{ \int_0^{T_{\boldsymbol{a}(\cdot)}} \left(\frac{1}{v(\boldsymbol{x}(t))} + \alpha\psi(\boldsymbol{x}(t)) B(x_0) \right) dt \right\}, \tag{3.17}$$

where $T_{\boldsymbol{a}(\cdot)}$ is the time required to leave the region. The steady state Hamilton-Jacobi-Bellman equation for this optimal cost is

$$|\nabla C(x; x_0)| = \frac{1}{v(x)} + \alpha\psi(x)B(x_0). \tag{3.18}$$

Under the equivalence of the time-dependent and steady-state Eikonal equation described in section 2.1.3, we see that (3.4) and (3.18) describe the same cost function. Here $\phi(x, c; B(x_0)) = 0$ if and only if $C(x; B(x_0)) = c$. We emphasize that this control-theoretic formulation was not the inspiration for this model, but it gives an idea of why we can compute optimal paths out of the region using gradient descent on the cost function.

3.3.10 Numerical Implementation

We briefly discuss the numerical implementation of our model. Since we are concerned with two spatial dimensions, grid-based finite difference methods are sufficient. We approximate (3.4) using the Godunov Hamiltonian, and since the set velocity is always positive, the explicit formula (2.66) applies. We use the second-order ENO approximations of ϕ_x and ϕ_y given in (2.67), in concert with second-order total variation diminishing Runge-Kutta time integration [OS91]. Thus our scheme will be second-order accurate near the level set $\Gamma(t)$ so long as the velocity (which depends on the elevation profile) is sufficiently smooth.

There is an implementation issue that is worth mentioning. Note that the initial function $\phi(x, 0)$ gives precisely the signed distance from x to $\Gamma(0)$. As the level sets evolve, we no longer have $\phi(x, t)$ is no longer the signed distance to $\Gamma(t)$, and thus $|\nabla\phi| \neq 1$. When $|\nabla\phi|$ becomes too large or too small near the zero level contour $\Gamma(t)$, the level set will speed up or slow down, and may become difficult to resolve. We can fix this by occasionally replacing ϕ with the signed distance function to $\Gamma(t)$. That is, we occasionally halt the time integration, reset $\phi(x, t) = \text{dist}(x, \Gamma(t))$ and continue. This process is known as *re-distancing*. The typical strategy for computing the distance function to the current contour $\Gamma(t)$ is to set

$\sigma(x) = \text{sign}(\phi(x,t))$, initialize $d(x,0) = \phi(x,t)$, solve the equation

$$d_\tau = \sigma(x)(1 - |\nabla d|), \quad \tau > 0, \tag{3.19}$$

until steady state. The steady state solution will then be the signed distance function to the current contour. One advantage of this method is that we do not need to explicitly compute the contour $\Gamma(t)$; the contour is resolved implicitly by observing the sign of $\phi(x,t)$. If there is no difficulty computing the contour $\Gamma(t)$, and $\phi(\cdot,t)$ is being resolved on a grid (x_i, y_j), one can explicitly calculate the distance from each grid point to the contour, eliminating the need to solve (3.19). However, this will only be computationally feasible in very low dimension. The re-distancing problem is well established in the literature and several schemes have been developed to solve the problem [EZL12, LDO17, SF99, SSO94b, SFS98]. We can more easily demonstrate the necessity of re-distancing in the next chapter, where we observe level sets of equal travel time in mountainous regions, so we put off the illustration until then, but note that we do indeed perform re-distancing while solving (3.4).

In addition to the level set equations, we must also solve the gradient descent equation (3.8) and evaluate the patrol effectiveness metrics from section 3.3.7. We solve the gradient descent equation using a stiff ordinary differential solver (`ode15s` in MATLAB) to find the paths from the protected region to the boundary of the protected region. The pristine proportion metric measures the proportion of the protected region that is pristine. The pristine region and the protected region are defined by polygons in our numerical formulation, and we use a built-in function provided by MATLAB to find the areas of these polygons. To calculate the proportion of the value protected requires the integrals in (3.10) to be explicitly calculated and again a built-in MATLAB function suffices.

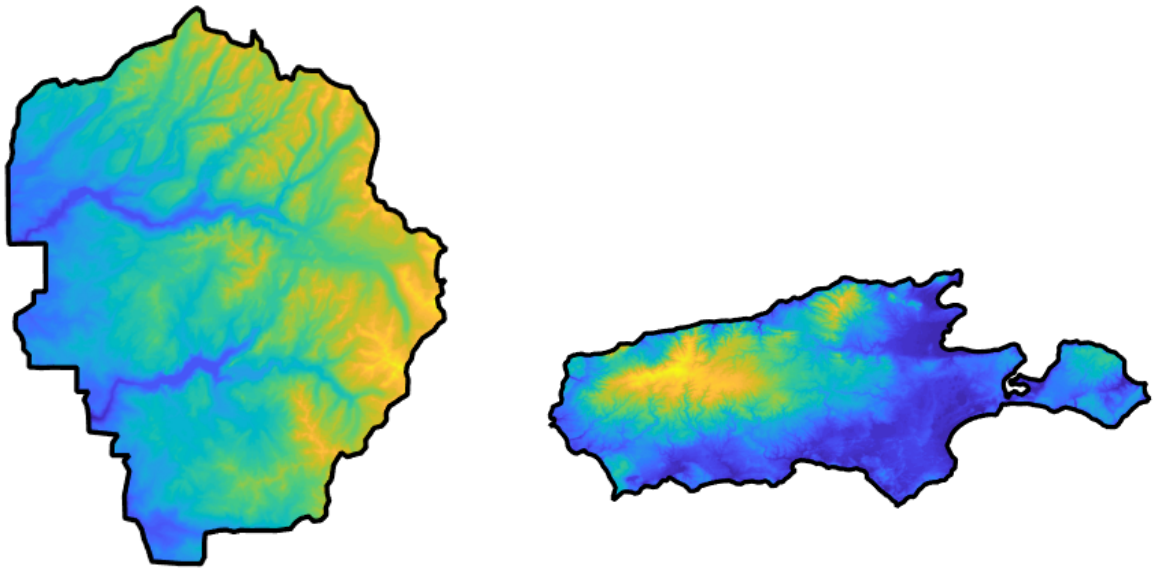
3.4 Results

We present results for our algorithm applied to two real world locations: Yosemite National Park in California, and Kangaroo Island in South Australia, pictured in fig. 3.6. Yosemite

National Park is a mountainous area with steep mountains and long valleys. Kangaroo Island has an interesting shape, with a narrow neck separating the main part of the island from a smaller part at the eastern end. For both locations we use real elevation data, sourced from the United States Geographical Survey (Yosemite National Park data) and the Foundation Spatial Data Framework (Kangaroo Island data). We apply several patrol strategies identified by [Alb10] and [JFT12] before suggesting some simple and more effective patrols that account for the geometry of the regions.

3.4.1 Yosemite National Park: No Patrol

We first consider the case without any patrols, so the cost function depends only on the effort required to travel from the extraction point to the boundary of the park. In fig. 3.7, we present results for two cases with different benefit functions. Both benefit functions have the same form, a quadratic increase from 0 at the boundary to a maximum value at the point furthest from the boundary, but fig. 3.7b has maximum benefit double that as in fig. 3.7a. Explicitly $B(x) = kd(x)(2d_m - d(x))/d_m$ where $d(x)$ is the distance from x to the boundary, and d_m is the maximum distance of any point to the boundary. The parameter k scales the



(a) Yosemite National Park.

(b) Kangaroo Island.

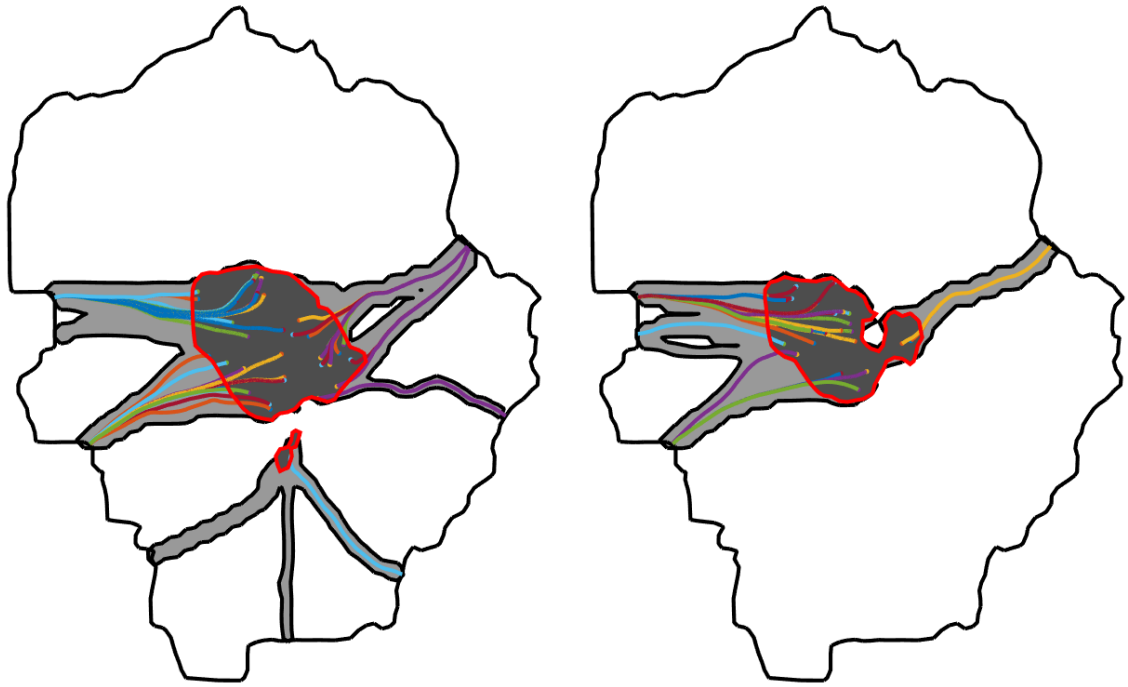
Figure 3.6: Elevation profiles for Yosemite National Park and Kangaroo Island. Yellow corresponds to higher elevation, blue to lower elevation (scales different in each figure).

benefit function.

The lower benefit case has a larger pristine area and smaller high-profit region. In the high-benefit case, there is enough incentive for extractors to expend more effort and extract from more locations within the protected region, obtaining much higher profits (although not doubled). A selection of the optimal paths from extraction points to the boundary of the protected region are also shown. In the sections that follow, we will use $k = 8$, the high-benefit case.

3.4.2 Yosemite National Park: Homogeneous Patrol

In fig. 3.8 we consider the simplest nonzero patrol strategy, a homogeneous patrol in which the entire protected area is patrolled with equal intensity. Two cases are shown, both with the high-benefit case $k = 8$ from the previous section, but with different patrol budgets E. The patrol strategy is simply $\psi \equiv E/A$ where E is the budget and A is the area of the protected



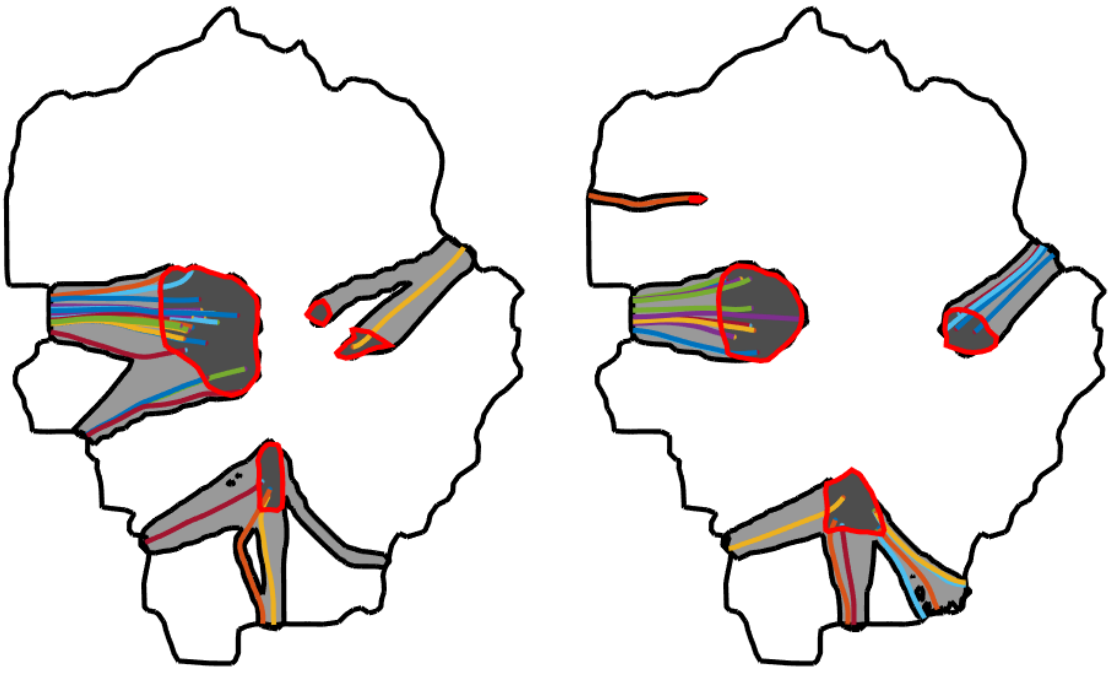
(a) $k = 4$, 24 paths out of the total 192 are shown. Maximum profit is 7.06×10^4 and pristine proportion is 0.867.

(b) $k = 8$, 37 paths out of the total of 294 are shown. Maximum profit is 1.65×10^5 and pristine proportion is 0.784.

Figure 3.7: Two cases for Yosemite National Park with no patrols, benefit based on distance from boundary d as $B(d) = kd(2d_m - d)/d_m$ (where $d_m = \max_\Omega d$) and $k = 4$ in (a) and 8 in (b). The dark gray denotes the high-profit region where extraction occurs, and light gray shows the envelope of paths the extractors follow when exiting the protected region. Some of the individual paths are shown for illustrative purposes.

region. We specify $\alpha = 1$ as the risk aversion parameter, as this choice gives agreement between our model and that of [Alb10] in simple symmetric case (discussed in section 3.3.8). As expected, when the patrol budget increases the pristine proportion increases and the maximum profit obtained by the extractor decreases.

In fig. 3.8, doubling the patrol budget decreases the maximum profit obtained by extractors by roughly 22%, increases the pristine proportion by roughly 4%, and increases the proportion of benefit protected by roughly 8.6%. These are clear improvements, but one perhaps would expect more change upon doubling the patrol density. Johnson, Fang and Tambe [JFT12] claimed that the homogeneous patrol was not optimal, and this agrees with our intuition that some areas of the protected region will never be targeted by extractors, so there is no need to patrol there.



(a) $E = 3 \times 10^4$, 26 paths out of the total of 205 are shown. Maximum profit is 1.27×10^5 and pristine proportion is 0.809.

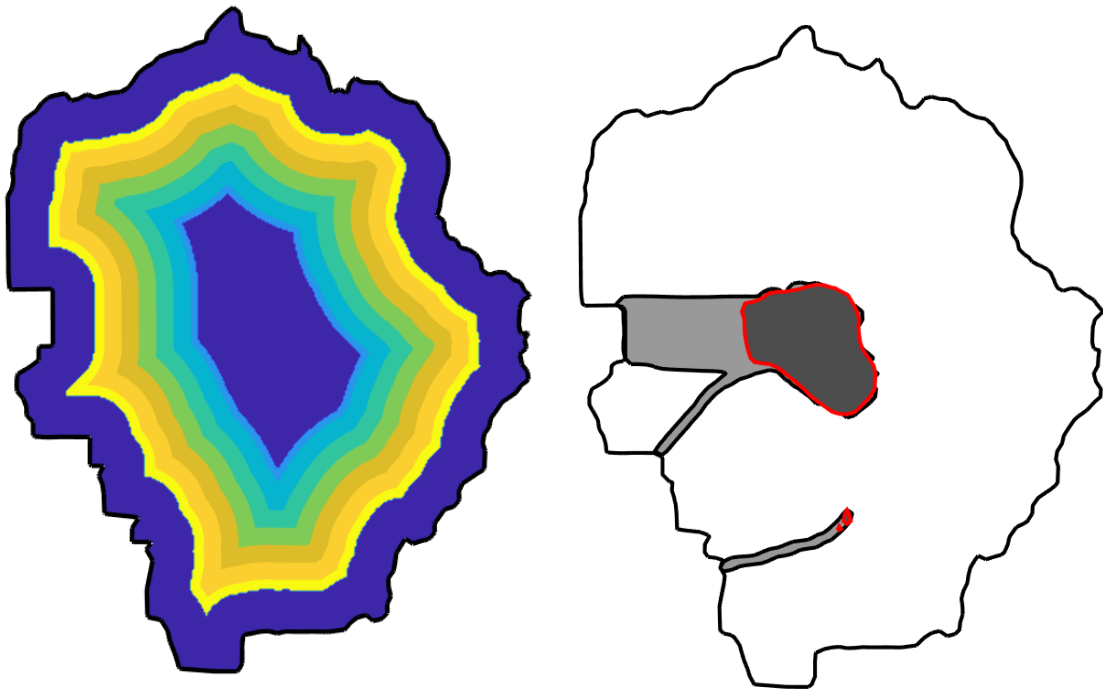
(b) $E = 6 \times 10^4$, 23 paths out of the total of 179 are shown. Maximum profit is 9.86×10^4 and pristine proportion is 0.841.

Figure 3.8: Two cases for Yosemite National Park with homogeneous patrols, benefit based on distance from boundary d as $B(d) = kd(2d_m - d)/d_m$ (where $d_m = \max_\Omega d$ and $k = 8$) and different budgets $E = 3 \times 10^4$ in (a) and $E = 6 \times 10^4$ in (b).

3.4.3 Yosemite National Park: Band Patrols

Johnson, Fang and Tambe[JFT12] identified the band patrol as the optimal strategy in symmetric circular protected regions. In the circular case, the band patrol is an annulus between two distances from the boundary, $0 < d_o < d_i < d_m$, where d_m is the maximum distance from the boundary, with highest patrol density at the outer extent d_o, and decreasing density moving towards the inner-most extent of the band d_i. In the general case, we set up a band patrol by patrolling between $0.3d_m$ and $0.7d_m$ from the boundary. Figure 3.9 shows the patrol strategy and the results of our algorithm. We did not implement the algorithm presented by [JFT12] to find the optimal band patrol, instead testing a number of band patrols and choosing the one that gave the best results.

The patrol budget in fig. 3.9b is 3×10^4, the same as for the homogeneous patrol in fig. 3.8a, but the outcome is significantly better for the patrollers. The homogeneous patrol had a pristine proportion 0.809 and was able to protect 76.9% of the total benefit, whereas



(a) The patrol strategy. Yellow corresponds to more intense patrolling, blue to no patrolling.

(b) The maximum profit is 1.25×10^5 and the pristine proportion is 0.901.

Figure 3.9: A band patrol in Yosemite National Park. The patrol is based on distance from boundary d and decreases linearly from $d = 0.3$ to $d = 0.7$, and is zero elsewhere. The benefit is $B(d) = kd(2d_m - d)/d_m$ (where $d_m = \max_\Omega d$ and $k = 8$) and the budget $E = 3 \times 10^4$.

the band patrol has a pristine proportion of 0.901 and protects 86% of the total benefit. The optimal band patrol of [JFT12] has the property that extractors do not enter the patrolled region, which ours does not. Despite this, it is still superior to the the homogenous patrol (even the homogeneous patrol with twice the budget) and the other band patrols tested.

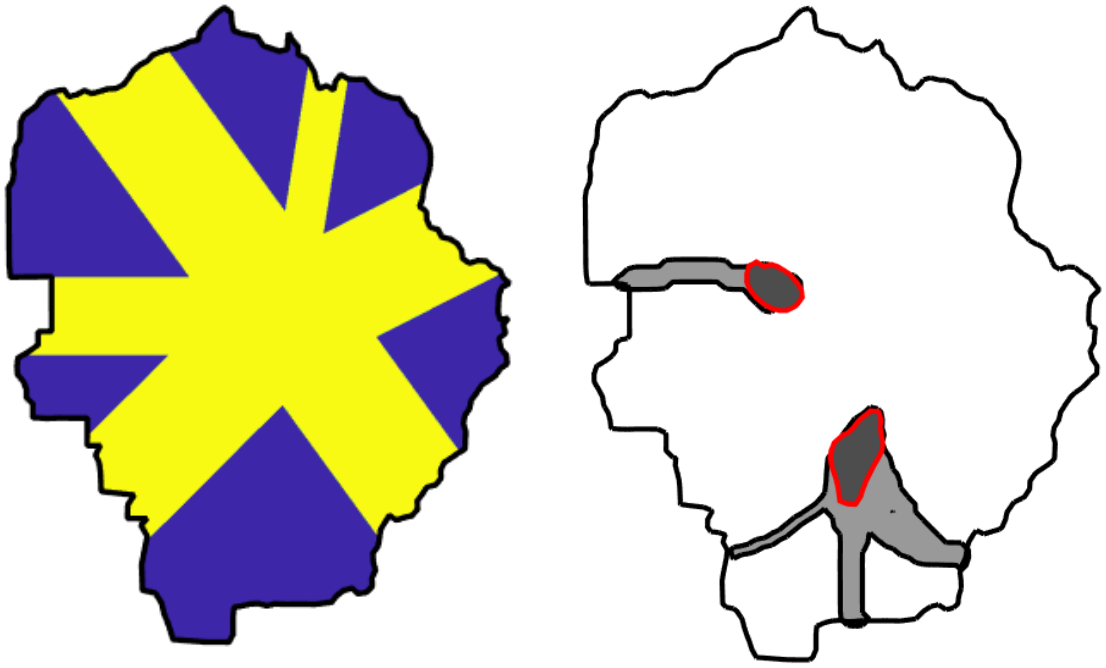
3.4.4 Yosemite National Park: Asymmetric Patrols

The patrol strategies employed above were identified by [Alb10] and are radially symmetric with respect to the geometry of Yosemite National Park. That is, these strategies depended only on the depth d of the point (x, y): $\psi(x, y) = \psi^*(d)$. Our method does not require patrols to be radially symmetric, and we here present an asymmetric patrol that outperforms the symmetric patrols discussed previously. Observing the results of the other patrols, it seems that extractors prefer to enter and leave the park at certain portions of the boundary: the portions that are most concave. This is intuitive, as entering at those areas will ensure less travel distance to reach the center. In response, we can design strategies to preferentially patrol those regions through which extractors are more likely to enter. Figure 3.10 shows just such a patrol strategy.

Exploiting the geometry of the park, rather than patrolling in a depth-dependent way, the pristine proportion is increased to 0.92 and 90.7% of the benefit is protected. These results are better than the band or homogeneous patrols with the same budget, and could likely be improved upon even further by finding even more specialized strategies. This result shows the critical importance of the geometry of the protected region. To a very rough approximation Yosemite is fairly circular, but the relatively small-scale concavities are very important in understanding the behavior of extractors.

3.4.5 Kangaroo Island

We now apply our method to Kangaroo Island, a small island off the southern coast of Australia. We present two patrol strategies (a homogeneous patrol and a custom designed patrol) that once again emphasize the importance of accounting for the geometry of the



(a) The patrol strategy. Yellow corresponds to more intense patrolling, blue to no patrolling.

(b) The maximum profit is 1.52×10^5 and the pristine proportion is 0.920.

Figure 3.10: A custom patrol in Yosemite National Park that is designed to patrol the concavities in the boundary of the park. The benefit is $B(d) = kd(2d_m - d)/d_m$ (where $d_m = \max_\Omega d$ and $k = 8$) and the budget $E = 3 \times 10^4$.

protected area. Kangaroo Island is fairly long and narrow with a portion at the eastern end connected to the rest of the island by a small isthmus.

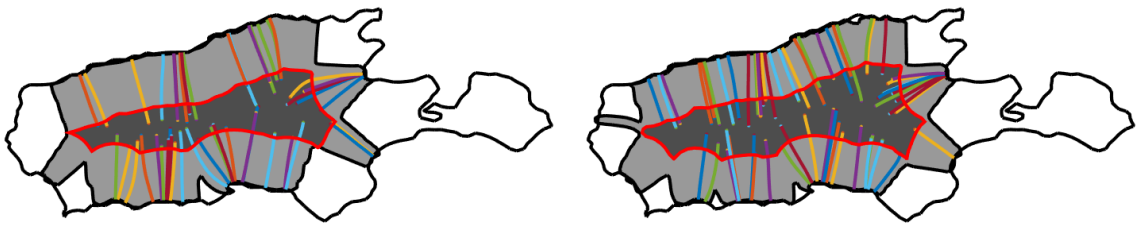
Figure 3.11 shows results of homogeneous patrols applied to Kangaroo Island. In this case, we observe that, somewhat counter-intuitively, the increased patrol budget results in a smaller pristine area proportion. The reason for this is that increasing the patrol budget decreases the profit function but causes the high profit area to spread out. That is, with increased patrol the surface profile of the profit function will look more like a plateau that has a smaller maximum than in the low budget case, but has a larger near-maximal area. Although the pristine region is smaller, the maximum profit obtained by the extractors is much lower, decreasing from 1.34×10^5 to 1.15×10^5, so the patrol does have a significant effect on the extractors. We also observe that a large portion of the non-pristine area comprises the paths leaving the region as opposed to the high-profit region itself. This should inform our decision regarding how to more effectively patrol the region. In any radially symmetric patrol, the high-profit region will also be radially symmetric meaning that it will likely

occupy the middle of the island and the paths will cover a very large area as they enter and exit from the north and south side of the island. If we can force the high profit region out of the middle of the island, then the paths will instead travel throughout the east and west portions of the island, hopefully occupying a smaller area. Another key geographic feature of the island is the peninsula at the eastern tip of the island. The peninsula is connected to the island by an isthmus thin enough that, for our purposes, the peninsula can almost be considered an independent region. Extractors will likely be uninterested by the peninsular region since it has much less depth and thus offers much less benefit than the main body of the island. Hence any patrol in that region is likely wasted effort.

With the homogeneous results in mind, fig. 3.12 shows the results of a more effective patrol, where the middle of the island is patrolled uniformly and the eastern and western ends are not patrolled. This patrol was able to increase the pristine area proportion to 0.875 compared to the homogeneous patrol which gave 0.349 with the same budget. Again, designing this patrol required some simple observations regarding the geometry of the region, and once again shows the importance of explicitly incorporating geographical information into the model.

3.5 Conclusion & Avenues for Future Work

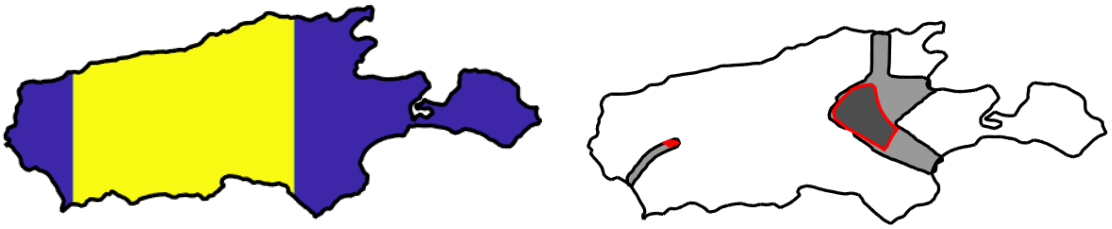
In this chapter, we presented a level set model for illegal deforestation. By treating environmental criminals as a front propagting inward from the boundary of a protected region, we are able to assign a cost to paths that move throughout the region. The cost for extractors



(a) $E = 3 \times 10^4$, 44 paths out of the total of 431 are shown. Maximum profit is 1.34×10^5 and pristine proportion is 0.349.

(b) $E = 6 \times 10^4$, 58 paths out of the total of 579 are shown. Maximum profit is 1.15×10^5 and pristine proportion is 0.305.

Figure 3.11: Two cases for Kangaroo Island with homogeneous patrols, benefit based on distance from boundary d as $B(d) = kd(2d_m - d)/d_m$ (where $d_m = \max_\Omega d$ and $k = 8$) and different budgets $E = 3 \times 10^4$ in (a) and $E = 6 \times 10^4$ in (b).



(a) The patrol strategy. Yellow corresponds to more intense patrolling, blue to no patrolling.

(b) The maximum profit is 1.54×10^5 and the pristine proportion is 0.875.

Figure 3.12: A custom patrol in Kangaroo Island that is designed to push the high profit region away from the center of the island. The benefit is $B(d) = kd(2d_m - d)/d_m$ (where $d_m = \max_\Omega d$ and $k = 8$) and the budget $E = 3 \times 10^4$.

takes into account travel time and capture probability. Previous continuum models relied on overly restrictive assumptions regarding the geometry of the region, and symmetry of all the key quantities. Our model agrees with these simpler models in some symmetric cases, but is able to eschew these assumptions and cover much more general scenarios. We tested our model in domains with irregular geometry and showed that very basic observations regarding the geometry can lead to vast improvements both in terms of pristine area, and in terms of value protected.

There are several avenues for future work in continuum modeling of illegal deforestation. One obvious way in which the model could be improved is by incorporating more geographical features, such as waterways and roads, in order to improve the realism. Additionally, we have only considered benefit functions that depend on distance from the boundary, which is likely not realistic. The proposed model can be evaluated with any benefit function, however defining a realistic and accurate benefit function for a given national park would require input from agencies involved in managing the park. In a similar vein, we could allow the benefit function to change with time, perhaps tracking animal movement for a model of animal poaching. However, this would like require major qualitative changes to the model.

Recently, Cartee and Vladimirsky [CV19] proposed a more control-theoretic model, focusing on the minimax problem arising from the adversarial game between the extractors and the patrol. While this is an improvement, the game is still static. Another improvement would be to develop a time-series version of the model wherein this is considered as one stage in an on-going game, and the patrol strategy can change, either instantaneously or at discrete times. Studying the large-time behavior of such a game could prove very enlightening

as to the best patrol strategies at each stage and the expected amount of deforestation over several years.

One piece of the model that is mildly unsatisfying is the hazy definition of "patrol strategy." As is, the patrol strategy is specified at the outset in the form of a patrol density function, but what exactly this means is open to some interpretation. In order to fully assist a law enforcement agency, one would need to deal with problems of resource allocation and patrol routes necessary to "achieve" a desired patrol density. Formulating this problem mathematically would be crucial to implementing this model in a real-world scenario.

Finally, and perhaps most importantly, the model lacks real world validation. In order to truly evaluate the model, it is crucial to test it against real world data. Identification of such data is difficult. While open source deforestation data sets exist[1][2], it is unclear how much of the data reflects illegal logging of timber, versus natural deforestation or illegal deforestation for other reasons such as agricultural land clearance [SU18]. A full analysis of the data, and an update to the model that incorporates the data in the correct ways, are key steps before the model could be implemented in practice.

3.6 Acknowledgement

Large portions of this chapter were first published in the SIAM Journal on Applied Mathematics, volume 79, number 3, 2019 [AFJ19]. Published by the Society for Industrial and Applied Mathematics (SIAM).

[1]DETER: https://infoamazonia.org/en/datasets/monthly-deforestation-brazil-deter/

[2]PRODES: http://data.globalforestwatch.org/datasets/4160f715e12d46a98c989bdbe7e5f4d6_1

CHAPTER 4

Optimal Human Navigation in Steep Terrain

We consider the problem of determining the optimal walking path between two points given elevation data in a region. If the terrain is fairly flat, this may be very easy as conventional wisdom ("the shortest path between two points is a straight line") will provide a good approximation to the optimal path. However, in mountainous regions, straight line travel is often inefficient or impossible and the optimal path between two points is no longer clear. In this chapter, we introduce a level set method for resolving optimal walking paths. We demonstrate the effectiveness of our method by computing optimal paths that travel throughout mountainous regions of Yosemite National Park. We include details regarding the numerical implementation of our model and, in concert with the previous chapter, we address a specific application of a law enforcement agency patrolling a nationally protected area.

4.1 Introduction & Previous Work

The problem of optimal path planning goes back at least as far as Dijkstra [Dij59] who designed an algorithm for optimally traversing weighted graphs. In the years since, significant effort has been devoted to developing and improving algorithms that find optimal or near-optimal paths in a discrete setting [HNG15, MS04, Pap85]. Others have used modified versions of Dijkstra's algorithm for path planning in a semi-continuous setting [AM06, Tsi95].

As noted in the previous chapter, path planning problems have been largely reframed using control theory and partial differential equations. Examples of these methods include finding geodesic paths on manifolds, [KS98], curvature constrained motion for self-driving

cars [AW01, Dub57, RS90, TTS10, TT13], ordered upwind methods (which can be seen as a continuous version of Dijkstra's algorithm) [AM09, SV01, SV03], reach-avoid games [Lyg04, BMO07, ZDH18], or optimal sailing routes that include random effects to account for weather patterns [FF19, SV15].

There has been some research into path planning in a geographical or terrain-based setting though most previous work is focused on discrete, graph-based methods employing Dijkstra's algorithm and its many variants: so-called A^* and D^* algorithms [KS95, SUA16]. Such methods have long been used for vehicular navigation and can be adapted to include real-time obstacle recognition [MMM16]. This problem is also of particular interest to those working on unmanned aerial vehicles (UAVs) and other autonomous robots [BVM10, Hac08, INY11, LG09]. In a continuous approach, Popović et al. [PVH17] propose a path planning algorithm for UAVs by maximizing an information functional that measures the amount of data a UAV can collect. However, to our knowledge, the methods of control theory and HJB equations have yet to be applied to terrain-based path planning previous to this work.

This chapter will be presented as follows. In section 4.2, we present a model that uses the level set method and a HJB formulation to compute optimal walking paths in a continuous setting where travel direction can be considered dynamically and walking speed is dependent on slope of the local terrain. This is as opposed to other terrain-based path planning methods which are fully or partially discrete and do not account for directional movement. In section 4.3, we discuss the numerical simulation of the model, including implementation concerns that are somewhat specific to terrain based path planning. We begin by testing the model against toy problems using synthetic elevation data specifically designed so that the "correct answer" is somewhat clear *a priori* and move on to use real elevation data of Yosemite National Park. The motivation for this work was to aid law enforcement agencies in efficiently patroling protected areas such as parks or forests, but with small adjustments, our method could be applied to optimal path planning in any number of scenarios.

This chapter reproduces an article previously published in Communications in Mathematical Sciences. A full copyright acknowledgment can be found at the end of the chapter.

4.2 A Level Set Model for Terrain-Based Optimal Path Planning

The primary tool we will use in our path planning approach is the level set method [OS88]. As discussed in chapter 2, the level set method implicitly solves an extremal geometry problem. If $\Gamma(0)$ is some initial contour, and we solve the Eikonal equation (2.6) with velocity $v(x)$ throughout a region, then $\Gamma(t)$ denotes the set of reachable points in this velocity field after traveling for time t. Thus, using the level set equation, one can compute the (approximate) time that it takes to travel from one point to another in our domain. Let $x_0 \in \mathbb{R}^2$ represent a starting point and $x_f \in \mathbb{R}^2$ represent a final point. For some small $\delta > 0$, let $\phi_0(x) = |x - x_0| - \delta$ so that $\Gamma(0) = \{|x - x_0| = \delta\}$ is a small circle around the point x_0. When $\Gamma(t)$ evolves outward with prescribed normal velocity $v(x)$, there will be some time $t^* > 0$ such that $x_f \in \Gamma(t^*)$; that is, at some positive time t^*, the level set will hit our ending point. This time t^* is the time required to travel from point x_0 to point x_f when traveling in the normal direction with velocity $v(x)$ (neglecting the small parameter δ). This approach can be seen as a continuous analog of Dijkstra's algorithm, assigning optimal travel times to points as the level sets sweep through the region. This gives a method for calculating travel times, but this model is too simple for our purposes, only allowing for travel in the normal direction which is potentially far from optimal. For example, if in a physical setting there is a large mountain between the points x_0 and x_f, one may wish to walk around the mountain rather than over the mountain, as normal direction travel may suggest. Thus at each point, one must not only consider the speed of travel, but also the direction of travel. Considering direction, it no longer makes sense to simply specify a velocity $v(x)$ at each point. Instead, we assume that walking velocity depends on both the gradient of the terrain at the current point and the direction of travel as we search for the optimal travel direction.

4.2.1 Our Model

For our model, assume that in addition to the starting and ending points $x_0, x_f \in \mathbb{R}^2$, there is an elevation profile $E(x)$ and a velocity function $v(s)$ that gives human walking velocity as a function of terrain slope s; for our purposes, we will again use the modified Irmischer

& Clarke [IC17] walking function as detailed in section 3.3.4. The control variable will take values $a \in \mathbb{S}^1$, unit vectors that represent walking direction. Now if one is standing at a point x and desires to walk in the direction a, they can walk with velocity $v(a \cdot \nabla E(x))$ since $a \cdot \nabla E(x)$ represents the slope at x in the direction of a. For each $a \in \mathbb{S}^1$, define the directional Hamiltonian $\overline{H}(x, p, a) = V(a \cdot \nabla E(x))\langle a, p\rangle$. Fixing a, and using this Hamiltonian, the equation

$$\phi_t + \overline{H}(x, \nabla\phi, a) = 0 \tag{4.1}$$

models advection in the direction of a, with level set velocity dependent on slope of the terrain in the direction a. To consider optimal travel, we take the supremum over all walking directions. Define the *optimal path Hamiltonian*

$$H(x, p) := \sup_{a \in \mathbb{S}^1} \overline{H}(x, p, a) = \sup_{a \in \mathbb{S}^1} \big\{ v(a \cdot \nabla E(x))\langle a, p\rangle \big\}. \tag{4.2}$$

Then the level sets of the solution to the Hamilton-Jacobi-Bellman equation

$$\phi_t + H(x, \nabla\phi) = 0 \tag{4.3}$$

correspond to contours of equal travel time when considering optimal walking direction. Note that this is indeed a level set equation as discussed in section 2.1.4, since this optimal path Hamiltonian is homogeneous of degree one in the variable p. Now to find the optimal travel time between points x_0 and x_f, one can use the same method described above: letting $\Gamma(0) = \{x \in \mathbb{R}^2 : |x - x_0| = \delta\}$ for small δ, evolve $\Gamma(t)$ using the level set equation with the optimal path Hamiltonian until the time $t^* > 0$ such that $x_f \in \Gamma(t^*)$. This t^* is the minimal time required to travel from x_0 to x_f. This procedure is displayed in fig. 4.1.

What remains is to compute the optimal path from x_0 to x_f: the path that requires time t^* to traverse. In order to do this, one simply needs to follow the characteristics of the Hamilton-Jacobi-Bellman equation. In order to do so, we follow the characteristics backwards from x_f to $\Gamma(0)$, since the initial control value is not resolved. The characteristic

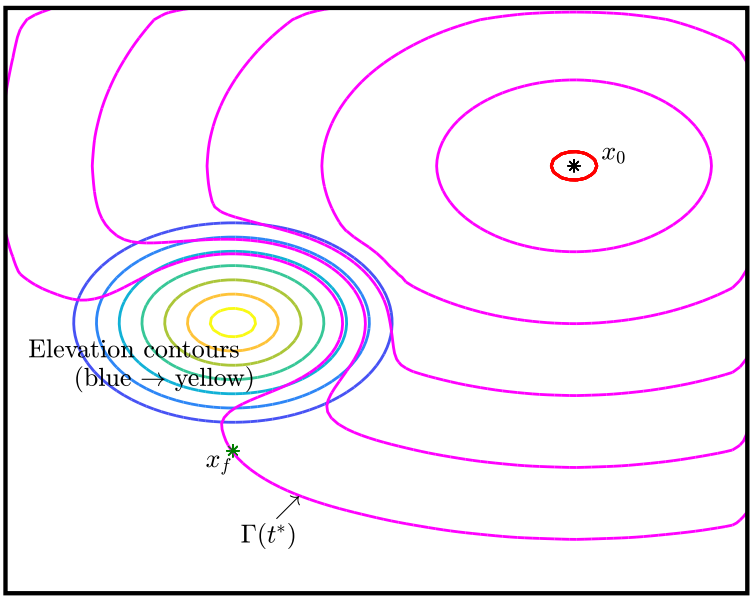


Figure 4.1: To find optimal travel time, begin with a small circle around x_0 (red) and evolve level sets $\Gamma(t)$ (magenta) outward until the time $t^* > 0$ at which $x_f \in \Gamma(t^*)$.

equations, which agree with the Pontryagin Optimality Principle from section 2.3.3, are

$$\begin{aligned} \dot{\boldsymbol{x}} &= -\nabla_p H(\boldsymbol{x}, \boldsymbol{p}), \quad \boldsymbol{x}(0) = x_f, \\ \dot{\boldsymbol{p}} &= \quad \nabla_x H(\boldsymbol{x}, \boldsymbol{p}), \quad \boldsymbol{p}(0) = \nabla\phi(x_f, t^*). \end{aligned} \tag{4.4}$$

Physically, one can imagine starting at the point x_f, considering what was the direction of the optimal step that led to the current point, stepping backwards in that direction and updating the direction in real time as one is walking backwards. Running this system of ODEs to time t^*, one will have backtracked optimally from b to $\Gamma(0)$.

To summarize, once we have defined the optimal path Hamiltonian (4.2), the algorithm for finding the optimal path consists of two steps:

1. Find the optimal travel time by advancing the PDE

$$\begin{aligned} \phi_t + H(x, \nabla\phi) &= 0, \\ \phi(x, 0) &= |x - x_0| - \delta, \end{aligned}$$

until the time $t^* > 0$ such that $x_f \in \Gamma(t^*)$.

2. Find the optimal travel path by advancing the ODE system

$$\begin{aligned}\dot{\boldsymbol{x}} &= -\nabla_p H(\boldsymbol{x}, \boldsymbol{p}), \quad \boldsymbol{x}(0) = x_f, \\ \dot{\boldsymbol{p}} &= \quad \nabla_x H(\boldsymbol{x}, \boldsymbol{p}), \quad \boldsymbol{p}(0) = \nabla\phi(x_f, t^*)\end{aligned}$$

until time t^*.

In another approach, Sethian and Vladimirsky [SV01, SV03] devise a static Hamilton-Jacobi-Bellman formulation for path planning. In doing so, they are able to solve similar problems without the time-dependence that is present in our model. While removing the time-dependence eliminates a dimension, since we are only solving problems in two spatial dimensions, our algorithm is sufficiently efficient for our purposes, and has the advantage that it is simple to implement at any order of accuracy one desires. If efficiency is a concern, local level set methods [AS95, LDM14, Min04, PMO99] could be used to speed up the level set computation.

4.2.2 The Associated Control Problem

Again, as in section 3.3.9, we note that underlying the formalism of section 4.2.1, there is a control problem that is being solved and a payoff function that is being maximized. As above, let $x_0 \in \mathbb{R}^2$ be the initial point. If one is standing at the point $x \in \mathbb{R}^2$, then traveling optimally away from the point x_0 for a time t is the same as maximizing the distance $|\boldsymbol{x}(t) - x_0|$, where $\boldsymbol{x}(\cdot)$ is a path with $\boldsymbol{x}(0) = x$. At each time along the path, denote the direction of travel by $\boldsymbol{a}(\cdot)$. As discussed above, the travel velocity at the point $\boldsymbol{x}(\cdot)$ and in the direction $\boldsymbol{a}(\cdot)$ is given by $v(\boldsymbol{a}(\cdot) \cdot \nabla E(\boldsymbol{x}(\cdot)))$. Thus, the problem can be phrased as

$$\sup_{\boldsymbol{a}(\cdot)} P_{x,t}[\boldsymbol{x}(\cdot), \boldsymbol{a}(\cdot)] = |\boldsymbol{x}(t) - x_0| \tag{4.5}$$

where the supremum is taken among measurable functions $\boldsymbol{a} : [0,t] \to \mathbb{S}^1$, and the system is subject to the state dynamics

$$\begin{aligned} \dot{\boldsymbol{x}}(s) &= v\big(\boldsymbol{a}(s) \cdot \nabla E(\boldsymbol{x}(s))\big)\boldsymbol{a}(s), \quad 0 < s \leq t \\ \boldsymbol{x}(0) &= x. \end{aligned} \tag{4.6}$$

Computing formally, one sees that the HJB equation associated with the value function $\phi(x,t) = \sup_{\boldsymbol{a}(\cdot)} P_{x,t}[\boldsymbol{x}(\cdot), \boldsymbol{a}(\cdot)]$ for this control problem is (4.3) with the optimal path Hamiltonian (4.2) and initial condition $\phi_0(x) = |x - x_0|$. We then make the slight modification $\phi_0(x) = |x - x_0| - \delta$ for small positive δ so that we can resolve the zero level sets $\Gamma(t)$ of $\phi(x,t)$ as the boundary of the reachable set at time t. To make this computation rigorous, one would need to require that the map $x \mapsto v(a \cdot \nabla E(x))$ is Lipschitz with a Lipschitz constant that is uniform in a. When this holds, the value function will be the viscosity solution of the HJB equation (4.3). However, this requirement will, in turn, depend upon the smoothness of the elevation data E, which is something we cannot guarantee, so we emphasize that, in our case, the connection between the value function for the optimal control problem and our HJB equation is merely formal.

4.2.3 Accounting for Uncertainty in the Starting Location

The above algorithm will compute a path for one who wishes to travel optimally throughout a region. We would like to incorporate some uncertainty into the model to account for a real world situation that law enforcement agents may encounter. Consider a scenario wherein a law enforcement agency has knowledge that environmental criminals (for example, poachers or illegal loggers) are operating within a protected region but can only identify the criminals' location with some uncertainty. Supposing that the criminals perpetrate a crime within the region and then travel to a known final destination, the law enforcement agency may want to predict which paths the criminals will take.

In this situation, assume that rather than a starting point x_0, we have a compact set Ω_0 of possible starting points along with a probability distribution from which one can sample

elements of Ω_0. The algorithm described above requires a starting point drawn from Ω_0 upon which one could calculate an optimal path to the end point x_f. However, we wish to calculate the optimal path to x_f from each point in Ω_0 and according to our procedure, this would require solving (4.3) individually for each point $x_0 \in \Omega_0$. Computationally, this would be very inefficient, but we can avert this need by turning the problem around: rather than starting from a point $x_0 \in \Omega_0$ and evolving level sets outward toward x_f, we can evolve level sets outward from x_f. As the level sets $\Gamma(t)$ evolve outward from x_f, they sweep through the region Ω_0 so that for each $x_0 \in \Omega_0$, we find a time $t^*(x_0)$ such that $x_0 \in \Gamma(t^*(x_0))$. The computation can be stopped when Ω_0 is inside $\Gamma(t)$ and for each $x_0 \in \Omega_0$, we will have found the time $t^*(x_0)$ required to travel from x_0 to x_f. This is pictured in fig. 4.2. Having done this, we can draw points N points $\{x_{0,n}\}_{n=1}^N$ from Ω_0 and calculate the optimal paths using (4.4) with starting values $\boldsymbol{x}(0) = x_{0,n}$, $\boldsymbol{p}(0) = \nabla\phi(x_0^{(n)}, t^*(x_{0,n})$. In this way, we can calculate optimal paths to any points in Ω_0 with only one level set computation.

As a minor note, according to Irmischer & Clarke [IC17], walking velocity is maximal when one is walking on a slight decline. Thus reversing the direction of the level set evolution means we must also reverse our sense of slope since the walking direction is now opposite

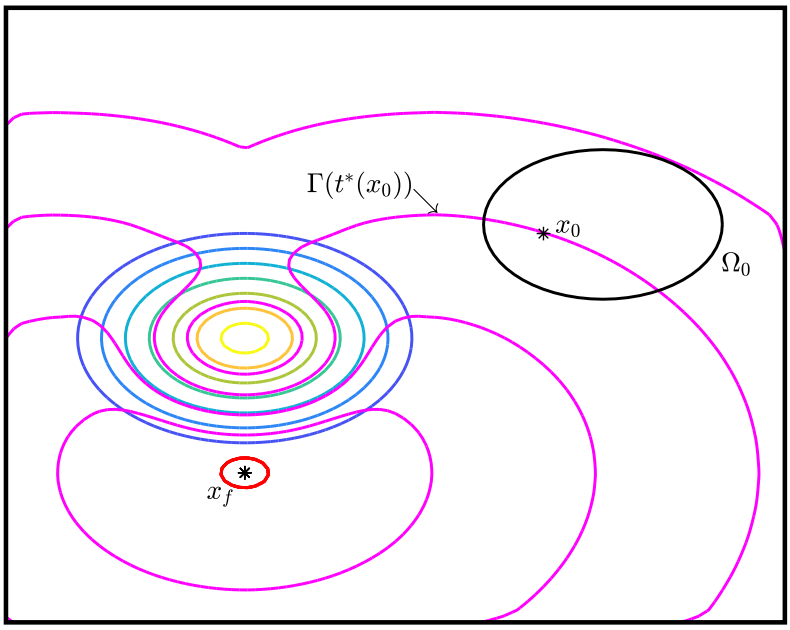


Figure 4.2: If there is uncertainty in the location of the starting point, we evolve level sets outward from b until they cover A, recording optimal times for each $a \in A$ as we go.

the direction of the level set evolution. Hence, we replace E with $-E$. In doing so, when we evolve level sets outward from x_f, we are actually calculating optimal travel as if one was traveling inward toward x_f. Thus we can still compute the optimal path from x_0 to x_f even though we use x_f as the "starting point" for the level sets.

4.3 Implementation, Results & Discussion

To simulate the model numerically, we would like to use the Godnuov Hamiltonian discussed in section 2.4. However, there are a few issues that arise. The first obstacle is deciding how to calculate our Hamiltonian since this requires a maximization over $a \in \mathbb{S}^1$. If the velocity function v is sufficiently simple, it may be possible to resolve this maximization explicitly using calculus. When this is not possible (as with our simulations), one can maximize H discretely. That is, rather than maximize over $a \in \mathbb{S}^1$, we maximize $\overline{H}(x, p, a)$ on a uniform discretization of $\mathbb{S}^1$. This causes some approximation error, but as long as $v(a \cdot \nabla E(x))$ is continuous in a for fixed x, this discrete maximization will tend to the exact supremum as the discretization of $\mathbb{S}^1$ becomes more precise.

Second, as seen in (2.63), the Godunov Hamiltonian is computed by performing a sequence of minimizations or maximizations. Again, we need to perform this computationally. In chapter 2, the motion was isotropic, meaning that the Hamilton-Jacobi-Bellman equation reduced to an Eikonal equation, and the Godunov Hamiltonian could be resolved explicitly. Here we are dealing with anisotropic motion (wherein the level set velocity depends on the direction of movement), and thus the explicit formula (2.66) is no longer available. Accordingly, we perform these optimizations discretely as well.

We use the second order ENO approximations for the derivatives of ϕ, and second order total variation diminishing Runge-Kutta time integration to simulate (4.3). One can solve the optimal trajectory ODE system (4.4) using any method one wishes. For relatively jagged elevation data E, the equation can become stiff, so it is recommended that one uses a stiff solver with accuracy that matches that of the numerical solution to (4.3).

While this describes the basics of the numerical implementation, there are some minor

adjustments required to obtain our results, which we discuss in section 4.3.2. Some of these issues are caused by roughness in the terrain data. Indeed, rough terrain data may violate the smoothness conditions required to ensure second order accuracy, and for this reason, first-order schemes may be sufficient to solve this problem. However, there is no significant difficulty in implementing the above scheme at second-order. This will provide second-order accuracy in regions where terrain is relatively smooth, and in some cases we observed slightly improved results (in terms of optimal travel time) using the second-order scheme.

4.3.1 Results

To test our code, we first ran simulations with synthetic (and very simple) elevation data. This allows us to gauge whether our model aligns with our intuition. When we were confident that our model and numerical methods were correct, we were able to download real elevation data from the United States Geological Survey and run simulations in a real national park. For our simulations, we chose Yosemite National Park and we ran optimal path simulations in the direct vicinity of the mountain El Capitan. Specifically, we use data spanning longitude 119°W - 120°W and lattitude 38°N - 39°N with 1/3 arcsecond resolution. This was obtained from the USGS National Map Viewer.

As in fig. 4.1 above, in the following images, the starting point x_0 is represented by the black asterisk surrounded with a red circle which denotes the starting contour $\Gamma(0)$. Next, the magenta contours represent several steps in the evolution of the contours $\Gamma(t)$. The green asterisk represents the point x_f and the thick black line represents the optimal path from x_0 to x_f. The elevation contours are plotted in colors ranging from blue representing low elevation to yellow representing high elevation. In our first simulations, we place mountains in certain areas and our intuition tells us that the optimal path should likely bend around the mountains since it would require too much effort to climb up the mountain. Our simulations do indeed reflect this, as seen in fig. 4.3.

Next, we use actual elevation data from the area surrounding El Capitan. Before showing the result of the simulation, we show the elevation profile and the starting and ending points

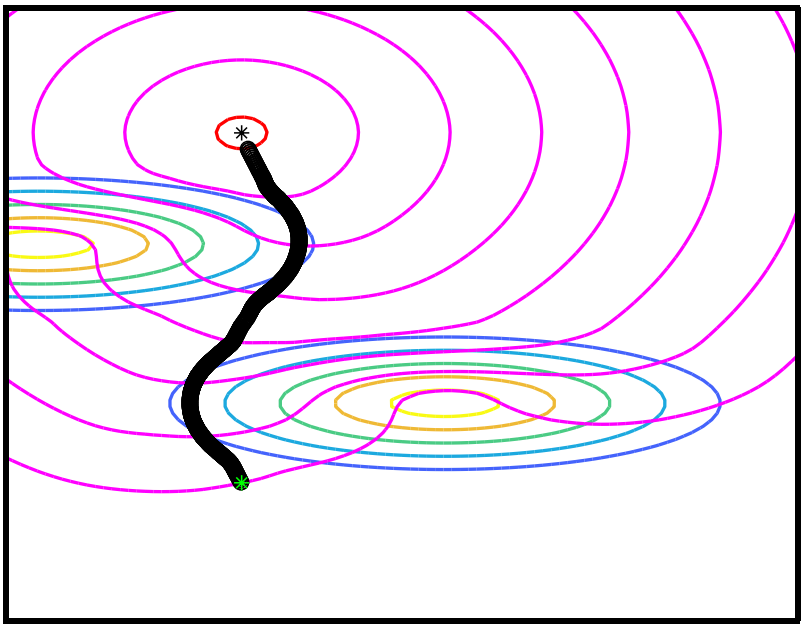


Figure 4.3: Optimal path winding around two mountains (toy problem).

in fig. 4.4. Note that directly above the endpoint, there is a very steep cliff face which should be nearly impossible to traverse. Thus we would expect the optimal path to travel to the east or west, descending down a gully rather than a cliff. Indeed, this is shown to happen in section 4.3.1, wherein the path travels down the eastern or western slope depending on the location of the initial point.

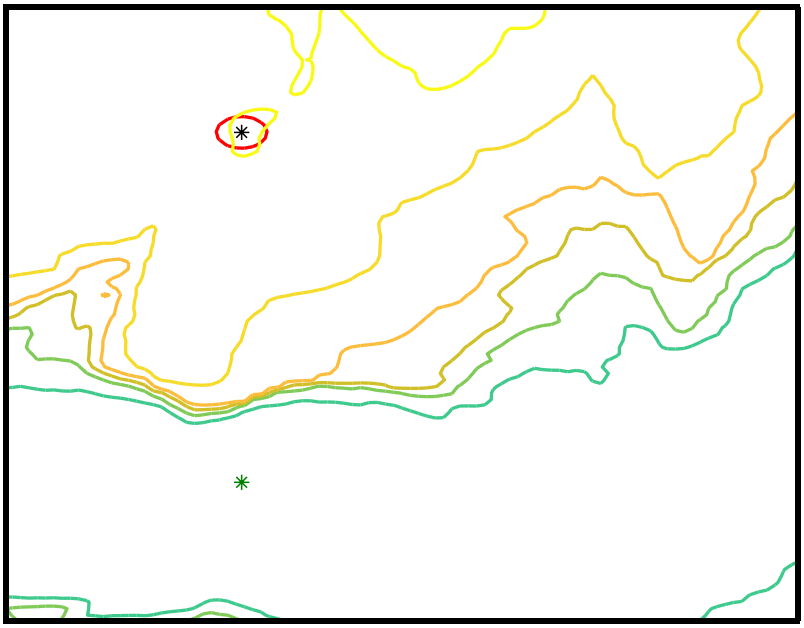
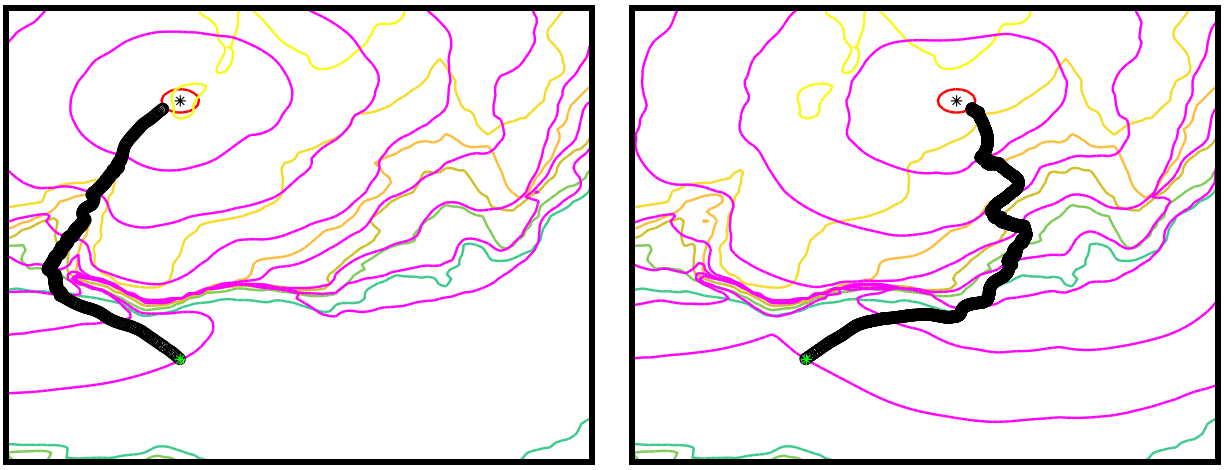


Figure 4.4: Elevation profile of El Capitan.



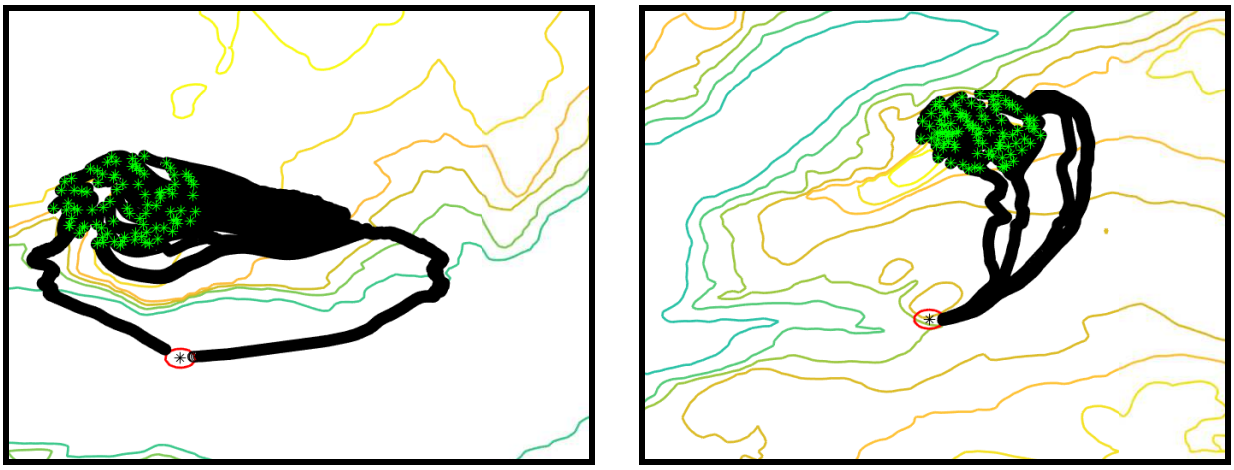
(a) Optimal path down the western slope.

(b) Optimal path down the eastern slope.

Figure 4.5: Optimal paths down from El Captian avoid the steep cliff.

Finally, we ran the algorithm that accounts for uncertainty in the location of the starting point. Before we display our results, we remind the reader of the distinction here. In all of these above results, we are calculating one optimal path from the point x_0 to the point x_f. Now we wish to calculate several optimal paths from the region Ω_0 to the point x_f. Whereas previously we evolved level sets outward from point x_0 until they reach the point x_f, now we evolve level sets outward from x_f until they subsume the region Ω_0 and record the optimal travel time for each $x_0 \in \Omega_0$ as the level sets sweep through the region. This is shown in fig. 4.2. We ran our algorithm in two different areas within Yosemite National Park. We let Ω_0 be a circle near the summit of El Capitan and calculated the optimal path down the mountain from 100 random points drawn uniformly from Ω_0. We then did the same thing but using the elevation profile of Half Dome, another peak in Yosemite National Park. The results are pictured in fig. 4.6.

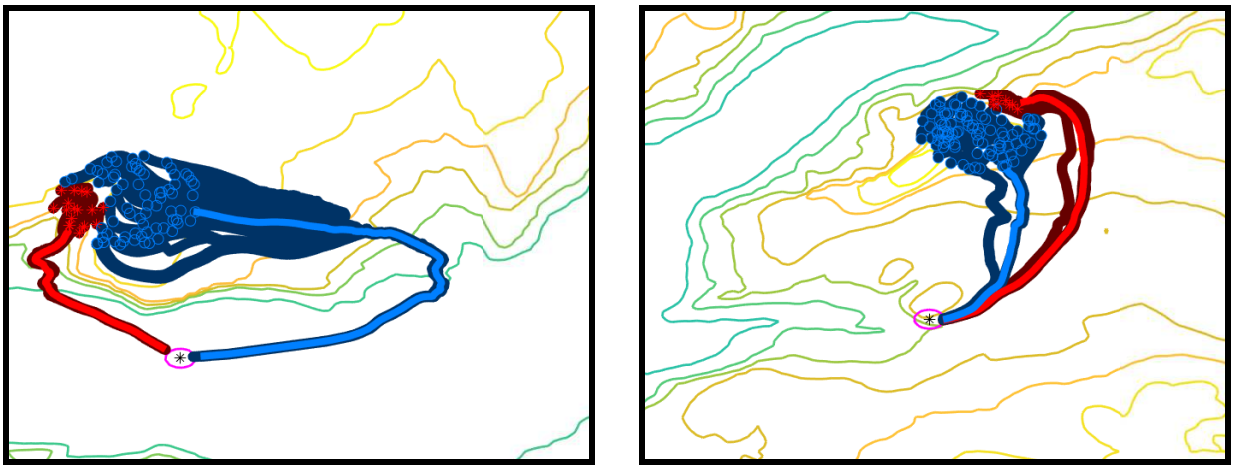
Note that in both cases, while there are 100 randomly chosen starting positions, all of the paths eventually conform to one of very few routes. We seek to quantify this similarity between some paths. Suppose we have calculated several paths $\{P_n\}_{n=1}^N$ starting from different locations. For each path, we know the time $T_n > 0$ required to traverse the path, so that $P_n : [0, T_n] \to \mathbb{R}^2$ and the paths are oriented so that $P(0) = x_f$ and $P(T_n) = x_{0,n}$, the n^{th} starting location. However, we are not concerned with where a path began, we would like to classify the route that the path eventually takes. Thus, we only consider points along the



(a) 100 optimal paths traveling down from the summit of El Capitan. **(b)** 100 optimal paths traveling down from the summit of Half Dome.

Figure 4.6: Calculation of optimal paths accounting for uncertainty in the initial location.

path that are outside of the set Ω_0; that is, if we define $T_n^* = \sup\{t : P_n(t) \notin \Omega_0\}$, we would only like to compare the paths on the intervals $[0, T_n^*]$. After making this restriction, one can then re-parametrize using $\tau = t/T_n^*$ so that for each n, $P_n : [0,1] \to \mathbb{R}^2$ denotes a path from the ending point x_f to the set Ω_0. Then it is easy to define a metric to judge whether two such paths lie nearby each other: for two paths P, Q, define $d(P,Q) = \int_0^1 |P(\tau) - Q(\tau)| \, \mathrm{d}\tau$. With this metric, one can evaluate the pairwise distances between our paths, $\{d(P_n, P_m)\}_{n,m=1}^N$. Now, using basic clustering algorithms, one can categorize the paths into collections that are morally the same, in the sense that they eventually collapse onto the same route. We performed this clustering for the above two examples. Specifically, we used k-means clustering with $k = 2$ clusters in each case, though other clustering methods could be used. The results are included in fig. 4.7, where the first cluster of paths is depicted in red and the second in blue. Here, we have plotted each of the 100 paths as well as the mean path for each cluster. Thus any initial point that is marked with a blue circle has a corresponding optimal path that eventually closely resembles the bright blue path and any initial point marked with a red asterisk has a corresponding optimal path that eventually closely resembles the bright red path. Returning to our original motivation, these graphics could be of great interest to law enforcement agencies who are tracking criminal movement. For example, in the case of El Capitan (fig. 4.7a), we observe that 20% of the paths travel down the western slope while 80% travel down the eastern slope. This may suggest to law enforcement that they



(a) Clustering the paths down from El Capitan into two collections.

(b) Clustering the paths down from Half Dome into two collections.

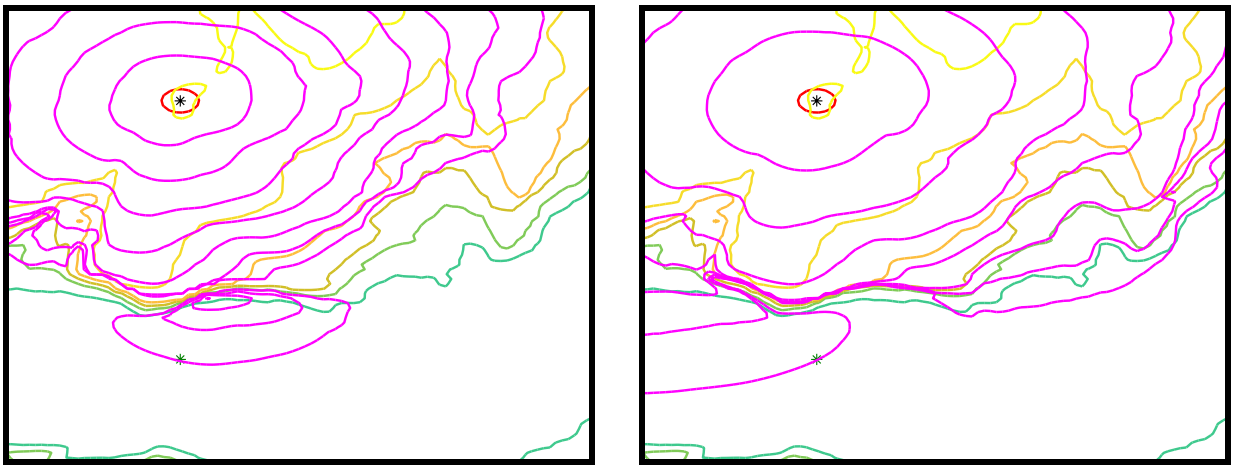
Figure 4.7: Clustering can help us identify which paths are morally the same. The bright blue path is the representative path for the blue points and the bright red path is the representative path for the red points.

should patrol the eastern slope with roughly four times the resources that they devote to the western slope.

4.3.2 Implementation Notes

There are a few specific issues that arise when implementing the model numerically. We mention three such issues (and their resolutions) and demonstrate their effects in fig. 4.8, fig. 4.9, and fig. 4.10. The first issue is the aforementioned re-distancing problem, discussed in section 3.3.10. To summarize, as the level sets evolve, $\phi(x,t)$ can become too flat or too steep near $\Gamma(t)$ which causes the level sets to become unreliable. To prevent this issue, we occasionally halt the time integration of (4.3) and replace the level set function with the distance function to the current level set. Figure 4.8 demonstrates the necessity for re-distancing. When we solve the equation without re-distancing, the level sets "jump" over the cliff (fig. 4.8a).

Next, as mentioned before, the system (4.4) used to find the optimal path becomes very stiff when non-smooth elevation profiles are used. Even when using a stiff solver, the results were unreliable in that the value of $\boldsymbol{p}(t)$ corresponding to a location $\boldsymbol{x}(t)$ was straying far from the theoretically correct value $\nabla\phi(\boldsymbol{x}(t),t)$. This was causing the "optimal path" that our code found to be wildly inaccurate, often times not even connecting x_f to x_0, opting instead



(a) Level sets without re-distancing "jump over" the cliff.

(b) Level sets with re-distancing wrap around the cliff.

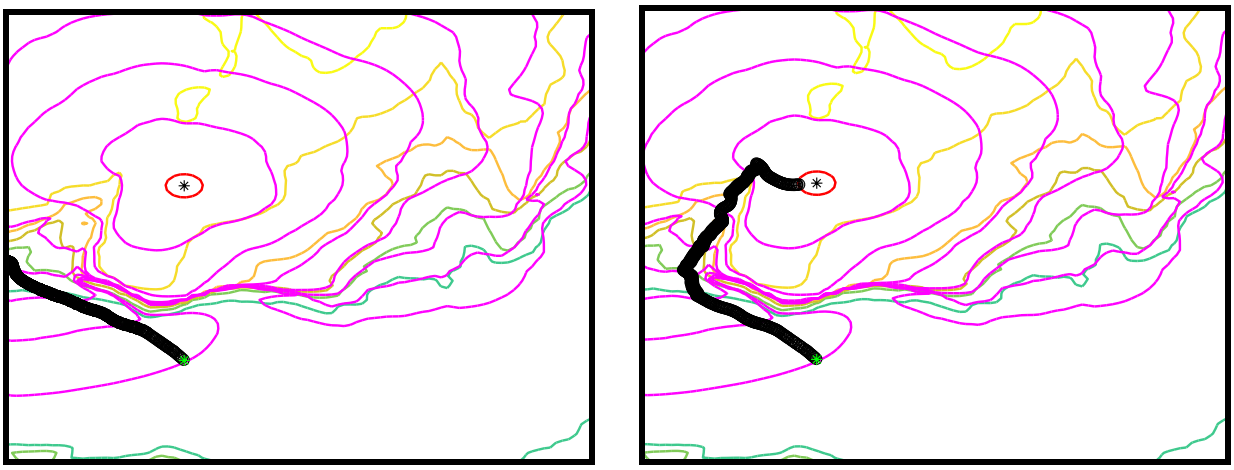
Figure 4.8: Level sets without, (a), and with, (b), re-distancing.

to wander off in some seemingly random direction. To fix this, we do something similar to the above fix: we occasionally stop the time integration, re-initialize $\boldsymbol{p}(t) = \nabla\phi(x(t), t)$ and then restart the time stepping. We refer to this as *re-initialization* and the effect is shown in fig. 4.9. If one can resolve ϕ at all values of (x, t) (as opposed to resolving ϕ only on a discrete grid), one could replace the system (4.4) with the single equation

$$\dot{\boldsymbol{x}} = -\nabla_p H(\boldsymbol{x}, \nabla\phi(\boldsymbol{x}, t)), \quad \boldsymbol{x}(0) = x_f. \tag{4.7}$$

When ϕ is not available at all points, one can solve for $\boldsymbol{p}$ using (4.4) and re-initialize as often as possible which is akin to solving (4.7).

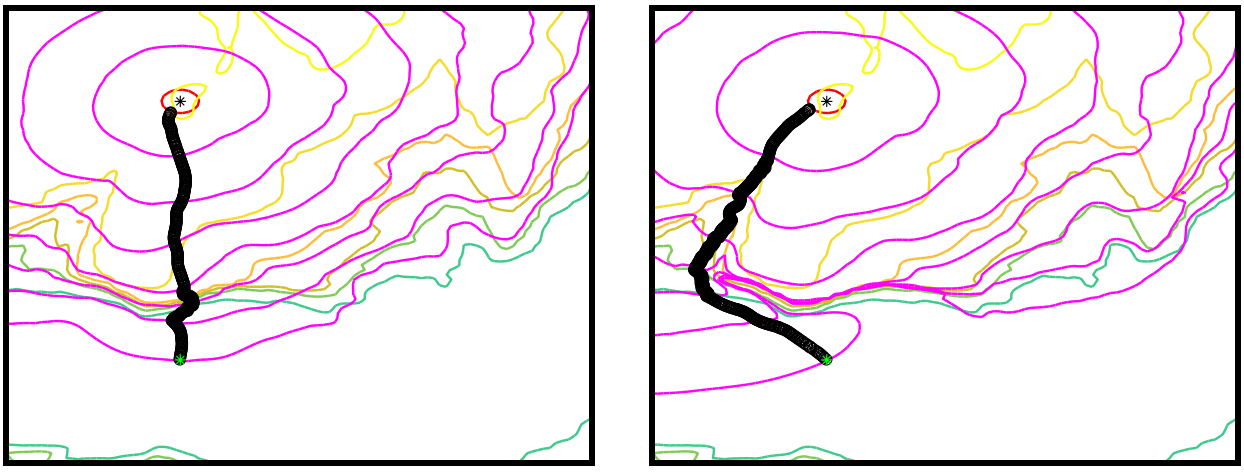
Finally, there is still one shortcoming of our Hamiltonian with respect to directional movement: the slope in the direction of travel and its orthogonal are completely decoupled. For example, consider a situation where there is a steep cliff face in the north-south direction while the slope in the east-west direction is very mild. Our model would allow an individual to walk east-west in this situation even though they may be standing on an prohibitively steep slope. To fix this problem, we add a penalty for walking in locations where the maximum slope in *any* direction is very large. This is as simple as multiplying our Hamiltonian by a pre-factor that is approximately 1 for low slopes and approximately zero for high slopes. We have chosen the penalization function $P(S) = \frac{1}{2} - \frac{1}{2}\tanh(S - 1)$ where $S = \frac{\text{rise}}{\text{run}}$ is



(a) Optimal path without re-initialization veers of the map.

(b) Optimal path with re-initialization finds its target.

Figure 4.9: Optimal paths without, (a), and with, (b), re-initilialization.



(a) Optimal path without high-slope penalization zig-zags up the cliff.

(b) Optimal path with high-slope penalization avoids the cliff.

Figure 4.10: Optimal paths without, (a), and with, (b), the high-slope penalization.

the slope. Thus we actually solve the Hamilton-Jacobi-Bellman equation with Hamiltonian $P(|\nabla E(x)|)H(x,p)$ where $H(x,p)$ is the optimal path Hamiltonian. The effect of this high-slope penalization is seen in fig. 4.10

4.4 Conclusions & Further Work

In this chapter, we presented a method for resolving optimal walking paths given terrain data. The key element of the model is a generalization of the level set equation. By representing the direction of travel for the level sets with a control variable, we constructed a

Hamiltonian whose corresponding level set equation models optimal travel. Using this, we described a simple algorithm for calculating the optimal walking path between a starting and ending point that consists of numerically solving a Hamilton-Jacobi-Bellman (HJB) equation and then a system of ordinary differential equations. Further, we suggest a method for incorporating uncertainty into the location of the starting point: by modifying the algorithm slightly, we can compute several optimal paths while only solving one HJB equation. To test our algorithm, we simulated our model first using artificial elevation data and then using the actual elevation data in certain regions of Yosemite National Park. In both cases, results aligned very well with our physical intuition. Finally, we sampled several different starting locations and calculated optimal paths that travel down from the summits of El Capitan and Half Dome and noticed that in both cases, the paths can be naturally clustered into collections of paths that follow the same basic route. We performed k-means clustering to separate the paths into such collections. Such clustering could suggest simple yet effective patrol strategies for a law enforcement agency tasked with patrolling nationally protected areas.

While this algorithm handles uncertainty in the location of the starting point, uncertainty could arise in many other places. In the ensuing chapter, we discuss adding uncertainty in the equation of motion (4.4). In doing so, the state dynamics are represented by a stochastic differential equation which leads to a slightly different Hamilton-Jacobi-Bellman equation.

4.5 Acknowledgement

This work was first published in Communications in Mathematical Sciences, vol. 17, num. 1, 2019 [PAB19], and is reproduced here by courtesy of International Press of Boston, Inc.

CHAPTER 5

Optimal Human Navigation with Uncertainty

In this chapter, we return to the problem of optimal path planning in mountainous terrain, using a control theoretic formulation and a Hamilton-Jacobi-Bellman equation. We reconstruct the previous path planning model to incorporate a stochastic component that can account for uncertainty in the problem, and thus includes a Hamilton-Jacobi-Bellman equation with viscosity. We discuss the model in the presence and absence of stochastic effects, and suggest numerical methods for simulating the model. In particular, we discuss two different notions of an optimal path when there is uncertainty in the problem.

5.1 Introduction & Previous Work

In section 4.1, we gave a short survey of path planning methods. One feature of most of those models is that they are completely deterministic. Some exceptions are [FF19, SV15] where some randomness is included to account for weather effects on a sailboat. But, for example, in the simplest reach-avoid games, the strategy of each team is known to the opposing team. Assumptions like this may not be realistic in practice, and thus it is important to incorporate some uncertainty into the models, and this can be done in a number of ways. In the case of reach-avoid games, Gilles and Vladimirsky suggest paths for the attackers or defenders that minimize or maximize risk in different ways [GV]. These type of stochastic effects are of special interest to those working on self-driving vehicles wherein misaligned axles, miscalibrated sensors or a host of other variables could significantly perturb an optimal driving path, and these can be modeled using randomness in a number of different ways [AM01, LR12, LLM14, Mar99, Mar15]. Others have suggested optimal path planning for

underwater unmanned vehicles using a stochastic drift term to account for the effects of the ocean's current [LJL15, SWL18, SL16].

In this chapter, we propose a method for optimal path planning where the walking velocity depends on the local slope in the direction or motion (and is thus anisotropic), and there is uncertainty in the equation of motion of the traveler. Rather than an ordinary differential equation, our equation of motion is a stochastic differential equation where Brownian motion can account for uncertainty in human walking speed, the ambient elevation data, terrain traversibility or other factors. In doing so, we follow a similar formulation as in chapter 4. However, since the Hamilton-Jacobi-Bellman equation for the stochastic case has viscosity, the level set method is no longer applicable, so we opt for a model more rooted in control theory.

This chapter is organized as follows: in section 5.2, we discuss the setup of our model, including some of the underlying mathematical formalism and how it pertains to optimal path planning. In section 5.3, we discuss the numerical methods that we used to simulate the model. In section 5.4, we discuss the results of our simulations. Specifically, we once again test our model against both synthetic data and real elevation data taken from the area surrounding El Capitan. We compare two different notions of optimal paths when stochastic effects are present. Finally, we observe that as the size of the uncertainty tends to zero—and thus the solution of the stochastic HJB equation tends to the viscosity solution of the ordinary HJB equation [CL83]—the optimal path tends toward the deterministic optimal path.

5.2 Mathematical Model

We discuss the construction of our model, recalling very briefly the components of our deterministic optimal control problem, and then detailing the extension to the stochastic case.

5.2.1 The Deterministic Path Planning Model

We recall the formulation of our path planning model from chapter 4, as discussed in section 4.2.2. There are some slight modifications since we used a level set formulation in the previous chapter, which we will abandon here since the stochastic Hamilton-Jacobi-Bellman equation is no longer a level set equation.

We imagine that a hiker is standing at a point $x_0 \in \mathbb{R}^2$, and wishes to walk to a point $x_f \in \mathbb{R}^2$. We are given the elevation profile $E : \mathbb{R}^2 \to \mathbb{R}$, and the modified Irmischer-Clarke [IC17] walking velocity function $v : \mathbb{R} \to [0, \infty)$ discussed in section 3.3.4. Let $\boldsymbol{x} : [0, T] \to \mathbb{R}^2$ represent the current position of the hiker, where $T > 0$ is the total walking time. Note, this terminal time T is a parameter that one specifies at the outset. Our control variable will be the walking direction along the path, $\boldsymbol{a} : [0, T] \to \mathbb{S}^1 := \{a \in \mathbb{R}^2 \,:\, |a| = 1\}$. Given all of this, the equation of motion for our hiker is

$$\begin{aligned} &\dot{\boldsymbol{x}}(t) = v\big(\nabla E(\boldsymbol{x}(t)) \cdot \boldsymbol{a}(t)\big)\boldsymbol{a}(t), \quad 0 < t \leq T, \\ &\boldsymbol{x}(0) = x_0. \end{aligned} \tag{5.1}$$

Note that $\nabla E(\boldsymbol{x}) \cdot \boldsymbol{a}$ represents the local slope in the walking direction. For the cost functional, we choose the Euclidean distance to the end point

$$C[\boldsymbol{x}(\cdot), \boldsymbol{a}(\cdot)] = |x - x_f| \,. \tag{5.2}$$

The dynamic programming principle discussed in section 2.3.2 shows that the value function ϕ for this optimal control problem satisfies the terminal valued Hamilton-Jacobi-Bellman equation

$$\begin{aligned} &\phi_t(x, t) + \inf_{a \in \mathbb{S}^1} \big\{ v\big(\nabla E(x) \cdot a\big) \langle a, \nabla \phi(x, t) \rangle \big\} = 0, \\ &\phi(x, T) = |x - x_f| \,. \end{aligned} \tag{5.3}$$

After (5.3) is solved, the optimal control plan is given by

$$\boldsymbol{a}^*(x,t) = \underset{a\in\mathbb{S}^1}{\operatorname{argmin}} \left\{ v\big(\nabla E(x)\cdot a\big)\langle a, \nabla\phi(x,t)\rangle \right\}. \tag{5.4}$$

and the optimal trajectory is given by

$$\begin{aligned} &\dot{\boldsymbol{x}}(t) = v\big(\nabla E(\boldsymbol{x}(t))\cdot \boldsymbol{a}^*(\boldsymbol{x}(t),t)\big)\boldsymbol{a}^*(\boldsymbol{x}(t),t), \quad 0 < t \le T, \\ &\boldsymbol{x}(0) = x_0. \end{aligned} \tag{5.5}$$

It was mentioned above that the terminal time T is a parameter that one must specify beforehand. This is the main departure from the fully deterministic model in chapter 4, where the algorithm provided the optimal travel time. It is important to note what this means for the model. Notice that for our cost functional, we have chosen a lump sum cost which is the distance from the end of the path $\boldsymbol{x}(T)$ to the desired end point x_f. The optimal control problem attempts to minimize this cost. Thus the path that we observe will be the path $\boldsymbol{x}$ whose end point is as close to x_f as possible, given time $T > 0$. If we select T too small, the path will not reach the end point, and in some cases this can lead to some interesting decisions regarding how the path should be constructed. We discuss this further in section 5.4.3.

5.2.2 The Stochastic Path Planning Model

In the above, we assume that all data is known perfectly and that there is no uncertainty. In reality, weather effects, instrumentation noise, incomplete elevation data or any number of other things could cause uncertainty in the equation of motion. Accordingly, we can account for random effects by considering a stochastic equation of motion

$$\begin{aligned} &d\boldsymbol{X}_t = v\big(\nabla E(\boldsymbol{X}_t)\cdot \boldsymbol{A}_t\big)\boldsymbol{A}_t dt + \sigma(\boldsymbol{X}_t, \boldsymbol{A}_t) d\boldsymbol{W}_t, \quad 0 < t \le T \\ &\boldsymbol{X}_0 = x_0, \end{aligned} \tag{5.6}$$

where the coordinates of $\boldsymbol{W}_t$ are independent one-dimensional Brownian motions, and σ is some function that determines the uncertainty.

Because the state is a stochatic process, we define the expected cost function

$$C[\boldsymbol{X}, \boldsymbol{A}] = \mathbb{E}\left(|\boldsymbol{X}_T - x_f|\right). \tag{5.7}$$

As in section 2.3.2, we fix x, t and restrict to trajectories $\boldsymbol{X}$ such that $\boldsymbol{X}_t = x$. The value function is then defined by

$$\phi(x,t) = \inf_{\boldsymbol{A}} C_{x,t}[\boldsymbol{X}, \boldsymbol{A}]. \tag{5.8}$$

Because there is no running cost in this problem, the dynamic programming principle for this problem takes the form

$$\phi(x,t) = \inf_{\boldsymbol{A}} \mathbb{E}\{\phi(\boldsymbol{X}_{t+\Delta t}, t+\Delta t)\}. \tag{5.9}$$

This states that the optimal remaining cost is constant along optimal trajectories. Rearranging and invoking the fundamental theorem of (stochastic) calculus yields

$$0 = \inf_{\boldsymbol{A}} \mathbb{E}\left\{\phi(\boldsymbol{X}_{t+\Delta t}, t+\Delta t) - \phi(x,t)\right\} = \inf_{\boldsymbol{A}} \mathbb{E}\left\{\int_t^{t+\Delta t} d\phi(\boldsymbol{X}_s, s)\right\}. \tag{5.10}$$

From here, assuming ϕ is smooth, we would like to resolve the infinitessimal $d\phi(\boldsymbol{X}_s, s)$. To do so, we recall the Itô formula for the stochastic chain rule. Suppose $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is smooth and $\boldsymbol{Y}_t$ is an n-dimensional stochastic process. Then

$$df(\boldsymbol{Y}_t, t) = f_t(\boldsymbol{Y}_t, t)dt + \langle \nabla f(\boldsymbol{Y}_t, t), d\boldsymbol{Y}_t \rangle + \frac{1}{2}\sum_{i,j=1}^{n} f_{x_i x_j}(\boldsymbol{Y}_t, t) dY_t^{(i)} dY_t^{(j)}, \tag{5.11}$$

where $dY_t^{(i)}$ are the components of $\boldsymbol{Y}_t$. Specific to our case, recalling the fundamental relationship $(dW_s)^2 \sim ds$ for a one-dimensional Brownian motion and operating formally, we see

$$dX_s^{(i)} dX_s^{(j)} = \sigma(\boldsymbol{X}_s, \boldsymbol{A}_s)^2 dW_s^{(i)} dW_s^{(j)} + o(ds). \tag{5.12}$$

Because the coordinates of $d\boldsymbol{W}_s$ are independent, we have $dW_s^{(i)}dW_s^{(j)} = \delta_{ij}ds$. Further,

$$\langle \nabla\phi(\boldsymbol{X}_s, s), d\boldsymbol{X}_s\rangle = v\big(\nabla E(\boldsymbol{X}_s)\cdot \boldsymbol{A}_s\big)\langle \nabla\phi(\boldsymbol{X}_s, s), \boldsymbol{A}_s\rangle ds + \sigma(\boldsymbol{X}_s, \boldsymbol{A}_s)\langle \nabla\phi(\boldsymbol{X}_s, s), d\boldsymbol{W}_s\rangle. \tag{5.13}$$

Inserting these into (5.10), we arrive at

$$\begin{aligned}\inf_{\boldsymbol{A}} \mathbb{E}\bigg\{ \int_t^{t+\Delta t} &\Big(\phi_t(\boldsymbol{X}_s, s) + v\big(\nabla E(\boldsymbol{X}_s)\cdot \boldsymbol{A}_s\big)\langle \nabla\phi(\boldsymbol{X}_s, s), \boldsymbol{X}_s\rangle\Big) ds \\ &+ \int_t^{t+\Delta t} \sigma(\boldsymbol{X}_s, \boldsymbol{A}_s)\langle \nabla\phi(\boldsymbol{X}_s, s), d\boldsymbol{W}_s\rangle \\ &+ \int_t^{t+\Delta t} \frac{1}{2}\sigma^2(\boldsymbol{X}_s, \boldsymbol{A}_s)\Delta\phi(\boldsymbol{X}_s, s) ds\bigg\} = 0.\end{aligned} \tag{5.14}$$

The middle term is zero in expectation since $\boldsymbol{X}_t$ is non-anticipating. Thus, sending $\Delta t \to 0$, (5.14) reduces to the stochastic Hamilton-Jacobi-Bellman equation

$$\begin{aligned}&\phi_t(x,t) + \inf_{a\in\mathbb{S}^1}\left\{ v\big(\nabla E(x)\cdot a\big)\langle \nabla\phi(x,t), a\rangle + \frac{1}{2}\sigma(x,a)^2\Delta\phi(x,t)\right\} = 0, \\ &\phi(x,T) = |x - x_f|\,.\end{aligned} \tag{5.15}$$

Note that the positive sign on the diffusion is correct since the equation runs backwards in time.

As in the deterministic case, one can solve (5.15) and determine the optimal control plan according to

$$\boldsymbol{A}_t^*(x) = \operatorname*{argmin}_{a\in\mathbb{S}^1}\left\{ v(\nabla E(x)\cdot a)\langle \nabla\phi(x,t), a\rangle + \frac{1}{2}\sigma(x,a)^2\Delta\phi(x,t)\right\}. \tag{5.16}$$

At this point, there is a choice as to how to construct a trajectory. One can use the optimal control computed from the stochastic HJB equation but simulate the deterministic path equation

$$\begin{aligned}&\dot{\boldsymbol{x}}(t) = v\big(\nabla E(\boldsymbol{x}(t))\cdot \boldsymbol{A}_t^*(\boldsymbol{x}(t))\big)\boldsymbol{A}_t^*(\boldsymbol{x}(t)), \quad 0 < t \le T, \\ &\boldsymbol{x}(0) = x_0,\end{aligned} \tag{5.17}$$

to arrive at an optimal path. Alternatively, to compute a single instance of a path, we could simulate the equation

$$\begin{aligned} d\boldsymbol{X}_t &= v\big(\nabla E(\boldsymbol{X}_t)\cdot \boldsymbol{A}_t^*(\boldsymbol{X}_t)\big)\boldsymbol{A}_t^*(\boldsymbol{X}_t)dt + \sigma(\boldsymbol{X}_t, \boldsymbol{A}_t^*(\boldsymbol{X}_t))d\boldsymbol{W}_t, \quad 0 < t \leq T \\ \boldsymbol{X}_0 &= x_0. \end{aligned} \tag{5.18}$$

Because this is only one instance and is subject to randomness, the path will not necessarily connect the points x_0 and x_f. However, we can average over many realizations to arrive at an expected optimal path. This is summarized in table 5.1.

These methods have different interpretations and may be better suited to modeling different physical scenarios. In using method (i), the uncertainty is factored into the planning of the route, but upon traversing the route, there is no uncertainty in the velocity. This could model a hiker walking through a forest. The hiker does not feel random perturbations in the walking velocity at each step; rather, the uncertainty is in the exact form of the landscape that lies ahead. Method (ii) may be of more practical use to a company shipping goods from one port to another, wherein each boat that makes the trip will actually feel random perturbations in velocity due to wind or currents. We will use both methods to compute paths and compare the results in section 5.4.

To begin simulating the model, it remains to decide the exact form of the uncertainty σ in the stochastic equation of motion. For our purposes, we take σ constant, so that we model uncertainty in the walking velocity in a general sense without specifying the exact nature of the uncertainty. As a consequence, if we reconsider (5.15), we notice that the viscosity term

Method (i)	Method (ii)
Stochastic HJB Equation (5.15) $\rightarrow$ control values	Stochastic HJB Equation (5.15) $\rightarrow$ control values
Deterministic equation of motion (5.17) $\dot{\boldsymbol{x}} = v\big(\nabla E(\boldsymbol{x})\cdot\boldsymbol{a}\big)\boldsymbol{a}$ $\rightarrow$ optimal path	Stochastic equation of motion (5.18) $\mathbb{E}\big(\ d\boldsymbol{X} = v\big(\nabla E(\boldsymbol{X})\cdot\boldsymbol{A}\big)\boldsymbol{A}dt + \sigma d\boldsymbol{W}\ \big)$ $\rightarrow$ expected optimal path

Table 5.1: The two options for how to compute optimal paths with uncertainty.

is independent of the control variable s, and thus the equation can be re-written

$$\begin{aligned} &\phi_t(x,t) + \frac{\sigma^2}{2}\Delta\phi(x,t) + \inf_{a\in\mathbb{S}^1}\{v(\nabla E(x)\cdot a)\langle\nabla\phi(x,t), a\rangle\} = 0, \\ &\phi(x,T) = |x - x_f|\,. \end{aligned} \tag{5.19}$$

This case is interesting because now the optimal control is resolved exactly as in the deterministic case, and (5.19) is reminiscent of the viscous Hamilton-Jacobi equation considered by Crandall and Lions [Eva80, CL83]. Thus, as discussed in section 2.2, if our Hamiltonian

$$H(x,p) \coloneqq \inf_{a\in\mathbb{S}^1}\{v(\nabla E(x)\cdot a)\langle p, a\rangle\} \tag{5.20}$$

is continuous, the solution $\phi^{(\sigma)}$ to equation (5.19) will converge to the viscosity solution ϕ of the inviscid equation (5.3) as $\sigma \searrow 0$. Continuity of $H(x,p)$ will depend on the nature of the elevation data, but we can observe this convergence empirically by considering the optimal path constructed by the the method at varying levels of uncertainty σ.

Other modeling decisions could be made. For example one could consider a running cost, allow for infinite horizon time, or allow σ to depend on x and s (perhaps accounting for local slope as with the velocity function). We chose our cost functional and finite horizon time in analogy with the previous level set model for deterministic path planning [PAB19]. With these decisions, the HJB equation remains time-dependent, and can thus be approximated very simply at high-order accuracy using techniques like ENO and WENO [OS91, LOC94]. As stated above, we will discuss the ramifications of this decision in section 5.4.3. Likewise, choosing σ constant simplifies the numerics in that it allows the diffusion to be resolved implicitly as seen in section 5.3.1. Alternatively, if one wishes to have $\sigma(x,s)$, it will likely be difficult to resolve the diffusion implicitly, and a steady-state, infinite horizon time formulation like those in [MT, ZDH18] may be necessary. This would also require more involved numerical schemes; one could likely use a modified fast marching [Tsi95, SV01, SV03] or fast sweeping method [KOQ04, TCO03]. Indeed, in the latter reference [TCO03], a fast sweeping scheme is specifically applied to an anisotropic Eikonal equation which describes geodesic distance on the graph of a smooth function.

5.3 Numerical Implementation

In this section, we discuss the numerical methods used to solve (5.19). In the $\sigma = 0$ case, we can use the same methods discussed in section 2.4. When $\sigma \neq 0$, the presence of the diffusion introduces some additional challenges.

5.3.1 A Semi-Implicit Scheme for (5.19)

In section 2.4 and section 4.3, we described an explicit scheme which is sufficient for solving (5.3). In order to numerically simulate the reaction-diffusion equation (5.19), one could simply insert the centered difference approximation to $\Delta\phi$ and implement the same explicit Euler time stepping. However, this will require exceedingly small time discretization, since the stability condition for forward Euler time stepping for a diffusion operator is of the form $\Delta t = O((\Delta x)^2, (\Delta y)^2)$. Instead, we resolve the diffusion implicitly.

Specifically, we let (x_i, y_j, t_n) be a uniform discretization of space and time, and consider the equation

$$\phi_t - \frac{1}{2}\sigma^2\Delta\phi + H(\nabla\phi) = 0, \tag{5.21}$$

with a prescribed initial condition. To transform (5.19) into this form, we simply need to make the transformation $\tau = T - t$. We solve (5.21) numerically using the scheme

$$\phi_{ij}^n - \frac{\sigma^2\Delta t}{2\Delta x}(\phi_x^+ - \phi_x^-)_{ij}^n - \frac{\sigma^2\Delta t}{2\Delta y}(\phi_y^+ - \phi_y^-)_{ij}^n = \phi_{ij}^{n-1} - \Delta t\hat{H}^G(\phi_x^+, \phi_x^-; \phi_y^+, \phi_y^-)_{ij}^{n-1}, \tag{5.22}$$

where $\hat{H}^G$ denotes the Godunov Hamiltonian.

Since implicit Euler time stepping for diffusion is unconditionally stable, our discretization is still only bound by the CFL condition (2.61). For larger values of σ, the resulting diffusion will smooth the solution ϕ, and thus sophisticated numerical Hamiltonians are probably no longer necessary. However, we have implemented this scheme as stated, so that as $\sigma \searrow 0$, the semi-implicit scheme (5.22) reverts to the explicit scheme used previously, and there are no stability issues. We note that this semi-implicit scheme is only available

when the uncertainty σ in the equation of motion is independent of the control variable. If σ depends on a, then the Hamilton-Jacobi-Bellman equation takes the form (5.15), and in this case the Hamiltonian cannot be decoupled from the diffusion term.

Before moving on to discuss the results of our simulations, we include some further implementation notes. First, in this application it is especially necessary to use the Godunov Hamiltonian (or some other essentially non-diffusive numerical Hamiltonian). The Lax-Friedrichs Hamiltonian, for example, is highly inadvisable since the strategy of the Lax-Friedrichs Hamiltonian is to add diffusion to the Hamilton-Jacobi equation. In this application, adding diffusion at level $\varepsilon = O(\Delta x, \Delta y)$ to the Hamilton-Jacobi equation is akin to adding uncertainty in the equation of motion at level $\varepsilon^{1/2}$. In the discretization we will use, the numerical diffusion would be on the order of 0.01, which would correspond to uncertainty in the equation of motion on the order of 0.1 m/s. This is a nontrivial level of uncertainty, representing roughly one tenth of the maximum walking velocity.

Secondly, the computational boundary in this problem is entirely artificial. At the boundary nodes, one cannot compute $\phi_x^+, \phi_x^-, \phi_y^+, \phi_y^-$. In the preceding work, this was never an issue and the boundary nodes could simply remain fixed. This is because characteristics of the Hamilton-Jacobi equations were flowing out of the boundary, and the Godunov Hamiltonian would naturally "choose" not to use those nodes. However, in this case, information will be flowing in from the computational boundary due to the diffusion, and we need to prevent this. To do so, after performing the semi-implicit update (5.22) at interior nodes, we enforce the boundary conditions suggested by [KOQ04] to update the boundary nodes:

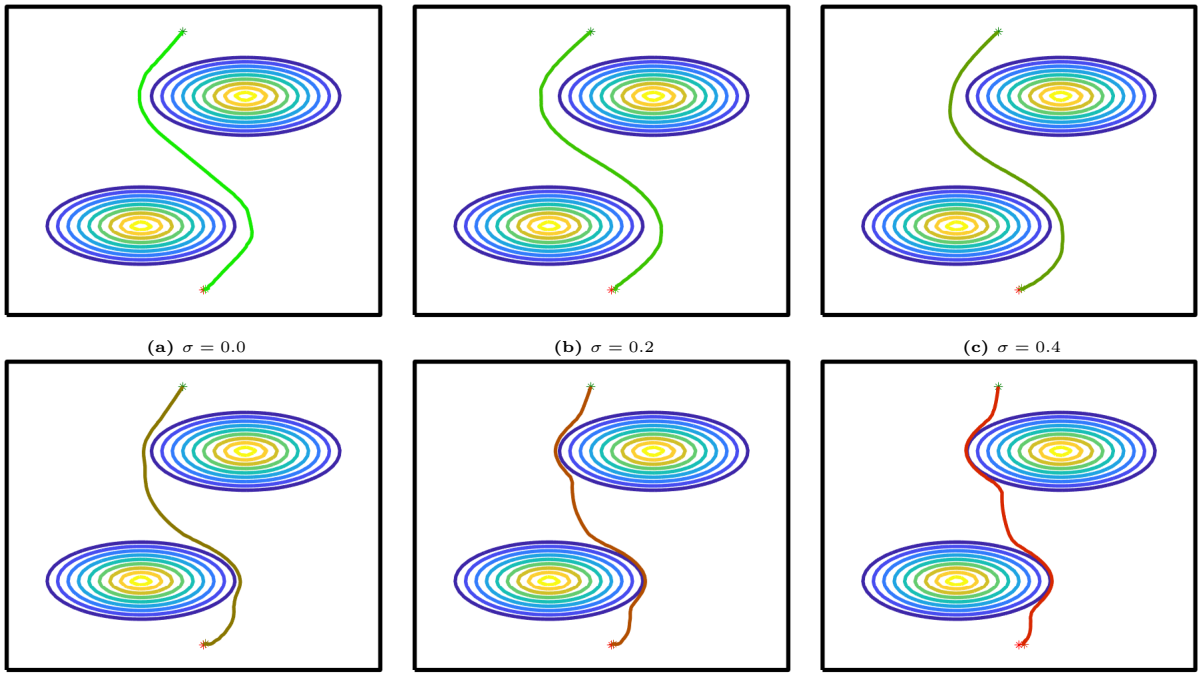
$$\begin{aligned}
\phi_{0,j}^n &= \min(\max(2\phi_{1,j}^n - \phi_{2,j}^n, \phi_{2,j}^n), \phi_{0,j}^{n-1}), \\
\phi_{I,j}^n &= \min(\max(2\phi_{I-1,j}^n - \phi_{I-2,j}^n, \phi_{I-2,j}^n), \phi_{I,j}^{n-1}), \\
\phi_{i,0}^n &= \min(\max(2\phi_{i,1}^n - \phi_{i,2}^n, \phi_{i,2}^n), \phi_{i,0}^{n-1}), \\
\phi_{i,J}^n &= \min(\max(2\phi_{i,J-1}^n - \phi_{i,J-2}^n, \phi_{i,J-2}^n), \phi_{i,J}^{n-1}).
\end{aligned} \tag{5.23}$$

These conditions attempt to extrapolate the solution outward from nearby nodes, while also enforcing zero flux inward from the boundary.

Lastly, in order to implement the Godunov scheme for our Hamiltonian, we must resolve three minima or maxima: the minimum involved in the definition of the Hamiltonian, and the two minima/maxima involved in the scheme itself. As discussed in section 4.3, we resolve all this minima and maxima discretely, by simply sampling points and choosing the correct ones. As long as the error from this descrete optimization remains on the order of Δx and Δy, the scheme will remain accurate to first order. The level of resolution needed for the discrete optimization depends somewhat on the problem, but empirically it appears that the most important facet is the regularity of the elevation data E. This makes intuitive sense: for less smooth elevation, the minimization in (5.20) taken with respect to the walking direction will require finer resolution to resolve. Likewise, for less smooth elevation, the discontinuities in the derivative of the solution ϕ become more severe. Thus the optimization sets in the Godunov scheme, which have the form $I(u_x^+, u_x^-)$ and $I(u_y^+, u_y^-)$, become larger. When σ is small and these extrema are taken on too coarse a grid, instabilities may appear. This was not a problem in section 4.3 because any instabilities that began to form were removed during the redistancing step. However, (5.15) is no longer a level set equation, and thus redistancing no longer applies.

5.4 Results & Observations

The model was implemented in MATLAB using the numerical schemes above to solve the Hamilton-Jacobi-Bellman equations, and the forward Euler method to solve the equation of motion. In the following figures, we will display elevation contours ranging from blue signifying low elevation to yellow signifying high elevation. The starting point x_0 will be marked with a green star and the desired end point x_f will be marked with a red star. The lines representing the walking paths will be plotted in colors ranging from green, symbolizing simulations with smaller σ values, to red, symbolizing larger σ values.



(a) $\sigma = 0.0$ (b) $\sigma = 0.2$ (c) $\sigma = 0.4$

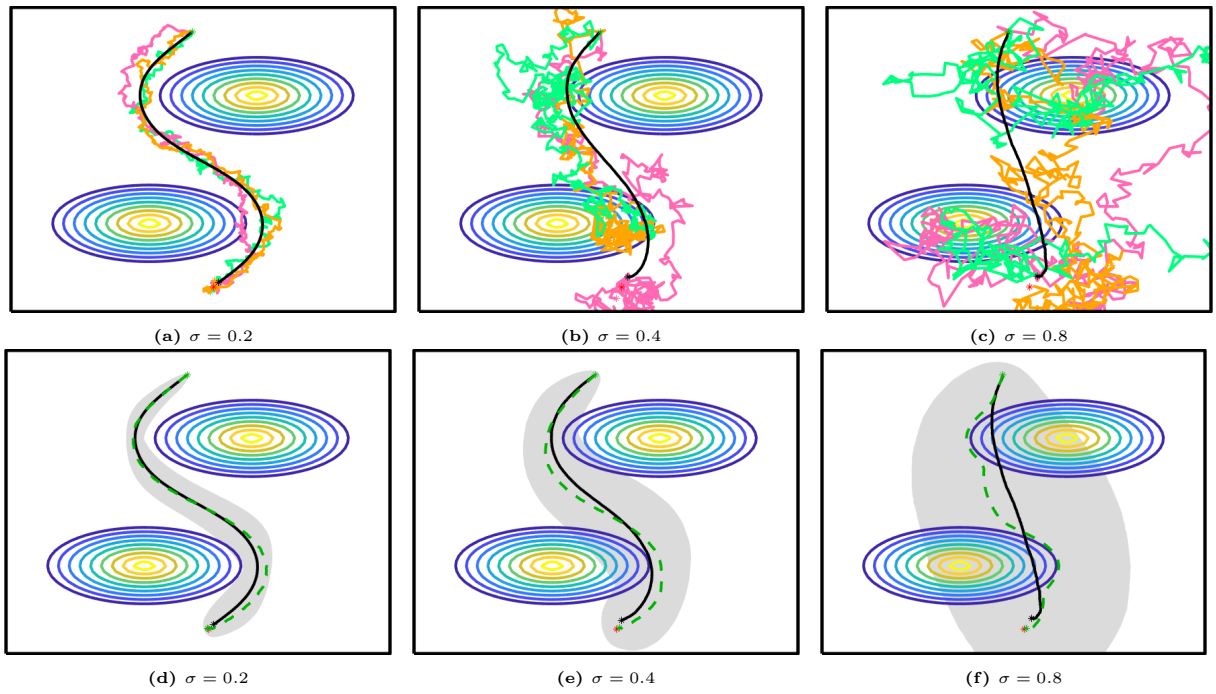
(d) $\sigma = 0.6$ (e) $\sigma = 0.8$ (f) $\sigma = 1.0$

Figure 5.1: Optimal paths using different σ values. End time $T = 3.8$ for each plot.

5.4.1 Synthetic Elevation Data

We began by testing our model against simple synthetic data. In fig. 5.1, we have computed optimal paths with several different levels of uncertainty σ. Referring to table 5.1, we are using method (i) to compute the optimal paths. That is, we are using the stochastic HJB equation to determine the optimal control values, but then computing the path using the deterministic equation of motion. In this example, the elevation is flat except for two large mountains that lie between the starting point and end point.

In the deterministic case, plotted in fig. 5.1a, the path curves around the mountains as one would expect: the walking velocity is significantly hampered by the change in elevation, so it is more efficient to avoid those regions. In this case, we see that the optimal path suggested by our algorithm is not particularly sensitive to small changes in σ. The path in fig. 5.1b which has $\sigma = 0.2$ looks very similar to that in fig. 5.1a which has $\sigma = 0$. However, as σ becomes larger, we do see significant changes in the path. The path in fig. 5.1f where $\sigma = 1$ is significantly different from the deterministic case. Here the uncertainty is on roughly the same order as the walking velocity. With this level of uncertainty, one could imagine



(a) $\sigma = 0.2$ **(b)** $\sigma = 0.4$ **(c)** $\sigma = 0.8$

(d) $\sigma = 0.2$ **(e)** $\sigma = 0.4$ **(f)** $\sigma = 0.8$

Figure 5.2: (a) - (c) Average path over 10000 trials (black), and three realizations of the stochastic equation of motion (colored) at different levels of uncertainty σ. (d) - (f) Average path (black) with standard deviation (grey), and the path computed using method (i) (dotted green).

walking through a forest in a very dense fog. In planning the path, this algorithm suggests that you walk directly toward your destination and adjust as necessary when obstacles arise.

Next, we consider method (ii) from table 5.1; that is, we simulate the stochastic equation of motion many times and compute the average path. Specifically, we simulate the equation L times, resulting in paths $\{\boldsymbol{x}_\ell(t)\}_{\ell=1}^{L}$ which are resolved at the same discrete time points, but with some randomness due to the Brownian motion. We then define the average path $\overline{\boldsymbol{x}}(t) = \frac{1}{L}\sum_{\ell=1}^{L} \boldsymbol{x}_\ell(t)$. We are still using the forward Euler method for the stochastic ODE and since the coefficient in front of the Brownian motion is independent of $\boldsymbol{X}_t$, this corresponds with the Milstein method which exhibits strong and weak convergence at first order [KP92]. In each of the following results, we simulated the equation of motion 10000 times and took the average path. Results are displayed in figures 5.2a-5.2c. The black line represents the average optimal path and the colored lines represent three individual realizations of the stochastic equation of motion. Here as σ gets larger, the individual realizations become less meaningful, but the average path is still somewhat smooth and roughly connects the starting point to the ending point.

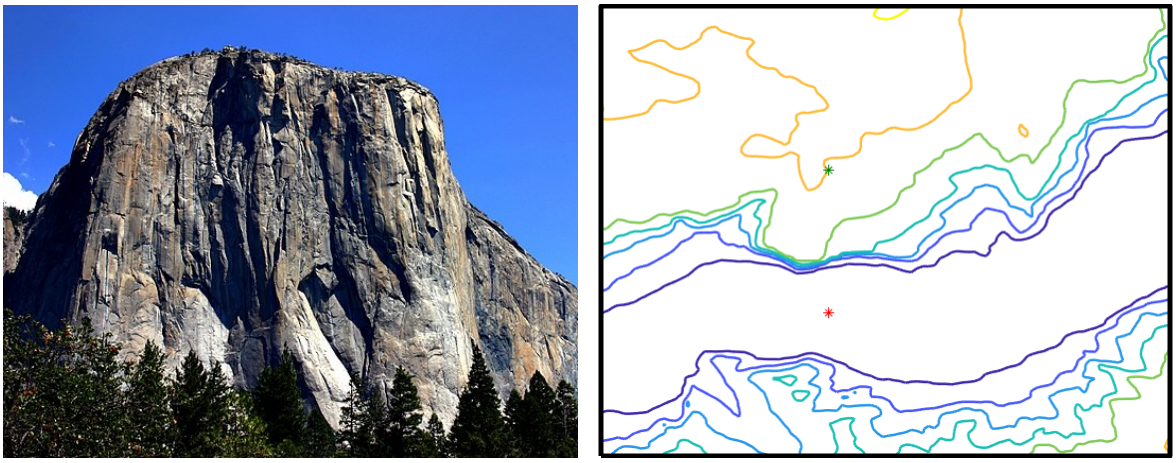
We also calculate a form of confidence interval to evaluate how close a single realization is likely to be to the average. To do this, at each point $(\overline{x}, \overline{y})$ along the average path, we calculate the standard deviation (δ_x, δ_y) in each of the coordinates. Then at each point, we plot in light grey the ellipse centered at $(\overline{x}, \overline{y})$ with radii (δ_x, δ_y) in the x or y direction, respectively. As we travel along the path plotting these ellipses, the grey envelope represents the set of points within one standard deviation of the average path. This is seen in figures 5.2d-5.2f. In these plots the average path is plotted as a solid black line. Now we also display the path that was computed using method (i) using a dotted green line. For small σ, the average path and the determistic path match fairly well. For larger σ, they begin to diverge, but the walking strategy seems similar: for larger σ, the average paths take a much more direct approach, cutting corners more closely, or walking directly over the mountains. In each case, the deterministic path stays well within one standard deviation of the average path. Notice that as σ gets larger, the standard deviation grows very quickly so that in fig. 5.2f the set of possible paths within one standard deviation of the average is quite large, and it may simply be that method (i) provides a more reasonable solution in this application.

5.4.2 Real Elevation Data

Seeing that our model works correctly for simplified elevation data, we tested the model against real elevation in the area surrounding the mountain El Capitan in Yosemite National Park. The elevation profile of El Capitan, along with the starting an ending points, is pictured in fig. 5.3. Notice that directly in between the starting and ending points, the contour lines lie close together, indicating a sheer cliff face. The starting point is near the summit of the mountain, and the ending point is in the valley to the south of the mountain, so any walking path should choose the gentler grades to the east or west of the cliff face.

Indeed, this is exactly what we observe, as seen in fig. 5.4. These paths were determined using method (i), the deterministic equation of motion. In these simulations all the scales in the problem are completely genuine. The region displayed in these figures is a rectangle

[1]Image courtesy of Mike Murphy, uploaded to Wikipedia Commons under Share-Alike license: https://commons.wikimedia.org/wiki/File:Yosemite_El_Capitan.jpg



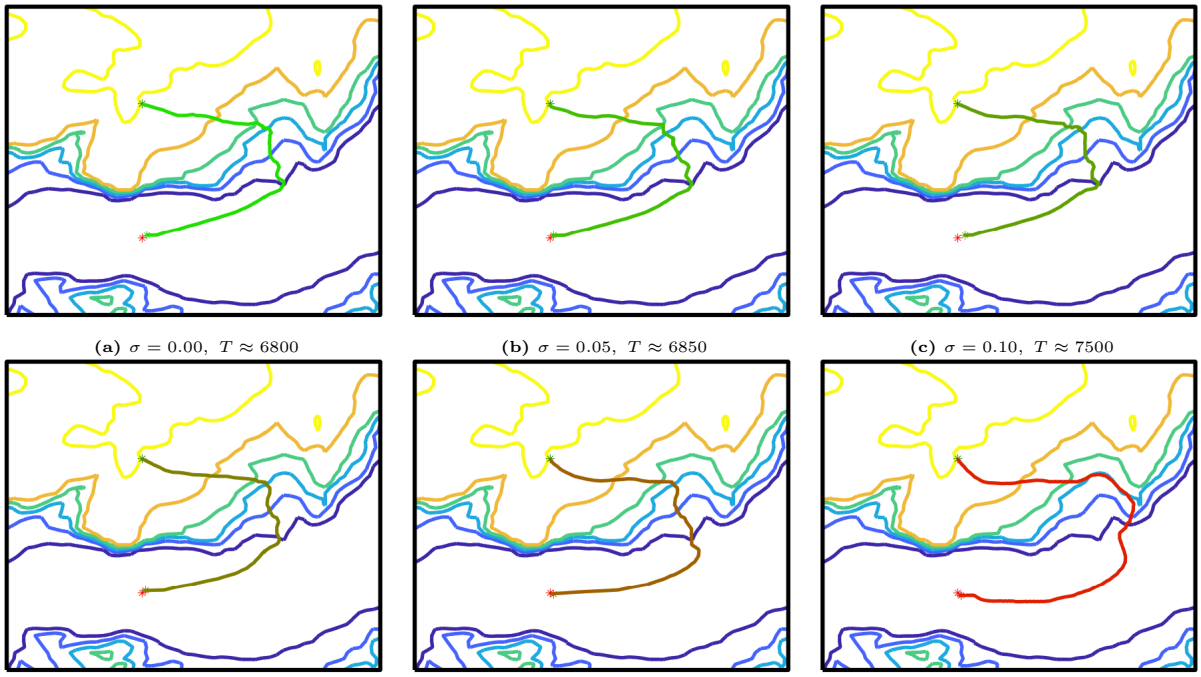
(a) The south-facing cliff face of El Capitan

(b) The elevation profile of El Capitan.

Figure 5.3: El Capitan, Yosemite National Park, California[1]

roughly 5 kilometers east-to-west and 6 kilometers north-to-south. The starting and ending points are roughly 2 kilometers apart and the terrain is mountainous, so several thousand seconds are required to traverse a path connecting the points. Here, the elevation data is much less smooth, and this leads to a greater sensitivity to small changes in σ. Figure 5.4b shows that at $\sigma = 0.05$, the optimal path looks largely the same as in the deterministic case, displayed in fig. 5.4a. However, when $\sigma = 0.3$, the optimal path is quite different, as seen in fig. 5.4f.

One significant note here: at different levels of σ, there are different optimal terminal times T. Recall, the parameter T must be chosen before simulating the model. In the case of the synthetic data in fig. 5.1, the terminal time T is not particularly sensitive to changes in σ, since qualitatively the paths are all similar and the amount of time that is "wasted" by taking a non-optimal path is not significant. In that case, we set $T = 3.8$ and any path with $\sigma \in [0, 1]$ had sufficient time to reach the endpoint. This is not the case in fig. 5.4, where small changes in σ lead to more significant qualitative changes in the paths. Indeed, the greedy strategy of taking a more direct route and then adjusting as necessary can be very costly in the case of El Capitan where it is very easy to get stuck in regions of severe grades, and be nearly unable to move. In this case, if one is reasonably uncertain about walking velocity as in fig. 5.4f, the algorithm suggests one should allow ample time, and take a safer route that more deliberately avoids regions with large changes in elevation.



(a) $\sigma = 0.00,\ T \approx 6800$ (b) $\sigma = 0.05,\ T \approx 6850$ (c) $\sigma = 0.10,\ T \approx 7500$

(d) $\sigma = 0.15,\ T \approx 7550$ (e) $\sigma = 0.20,\ T \approx 8950$ (f) $\sigma = 0.30,\ T \approx 15550$

Figure 5.4: Optimal paths descending El Capitan using different levels of uncertainty σ.

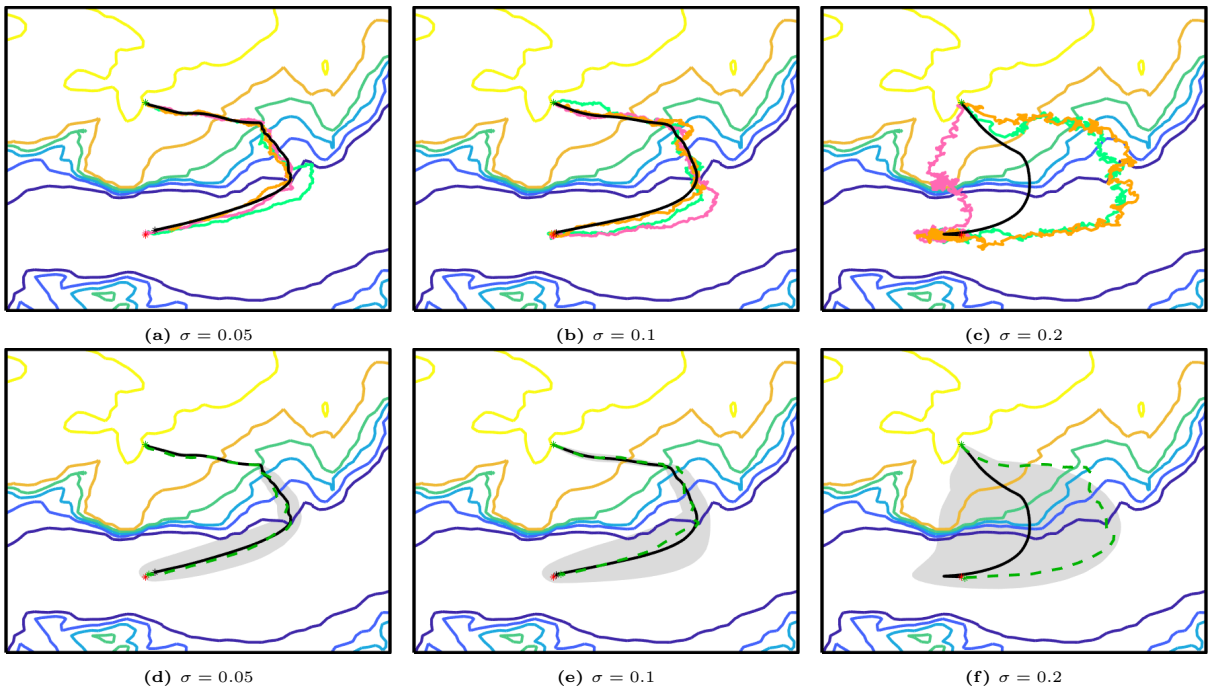
Because this route is significantly different, it requires a terminal time of roughly $T = 15550$ seconds, as opposed to a terminal time closer to $T = 6800$ seconds as in fig. 5.4a. For larger values of σ (for example $\sigma > 0.5$), the path will not make it down the mountain even given exorbitantly large terminal time, because it will walk too close to the cliff, become stuck, and have insufficient time to adjust. We say more about the role of the parameter T, especially as it pertains to impassable obstacles such as the El Capitan cliff face, in section 5.4.3.

As in the previous section, we would also like to use method (ii) to construct a path. In figures 5.5a-5.5c, we plot the average path along with three realizations in the case that $\sigma = 0.05, 0.1$ and 0.2. When $\sigma = 0.05, 0.1$, each of these realizations is fairly close to the average path, and the results are similar to those obtained using method (i). We have also included the region that is one standard deviation away from the average path, as seen in figures 5.5d-5.5f. Even when σ is very small, the variance in how the paths descend the mountain is fairly large. This is because small perturbations in that region will more qualitatively change the course of path. The results for $\sigma = 0.2$—displayed in figures 5.5c,5.5f—do not seem particularly meaningful. In this case, the uncertainty in the walking velocity is large enough

that if the path approaches the large cliff face, the random perturbation can cause the path to move down the cliff. In this region, the walking velocity is approximately zero, and so the random effects are the driving force for the movement. In those simulations, a large enough portion of the paths descended the cliff in this manner, leading to a skewed average path, and an enormously large standard deviation. Similar problems may arise whenever there are regions where the walking velocity is very small. In such cases, it seems that using the deterministic equation of motion with the stochastic control values (as in fig. 5.4) will give a much more meaningful result.

5.4.3 Impassable Obstacles and the Role of the Parameter T

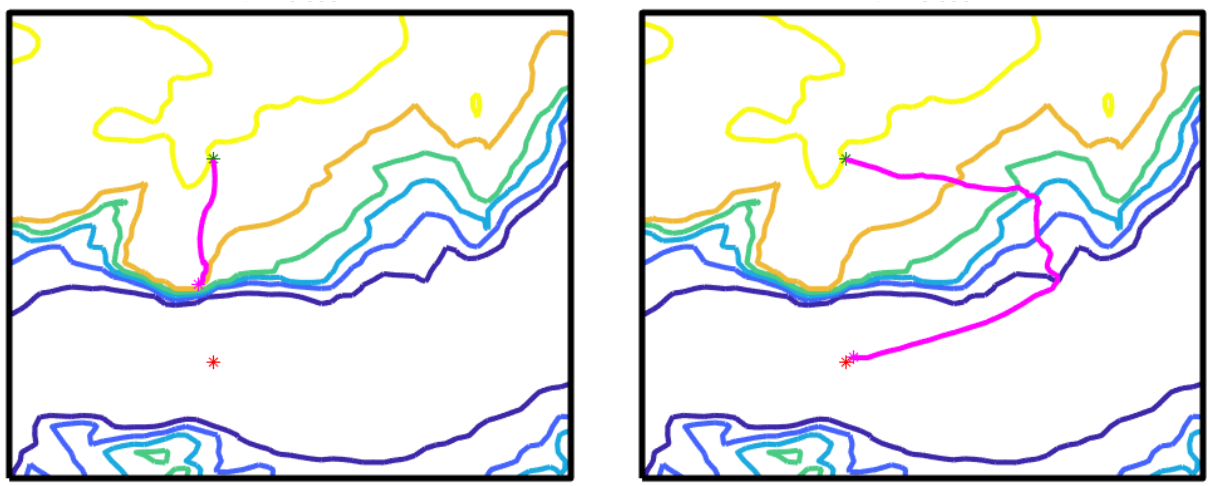
We remarked in section 5.2.1 that different choices for the terminal time T can lead to qualitative changes in how the path is constructed. We can observe this in the example of El Capitan. In fig. 5.6, we used $\sigma = 0$ so that the model is fully deterministic, and we simulated the model with two different terminal times T. In fig. 5.6a, we see that with terminal time



(a) $\sigma = 0.05$ **(b)** $\sigma = 0.1$ **(c)** $\sigma = 0.2$

(d) $\sigma = 0.05$ **(e)** $\sigma = 0.1$ **(f)** $\sigma = 0.2$

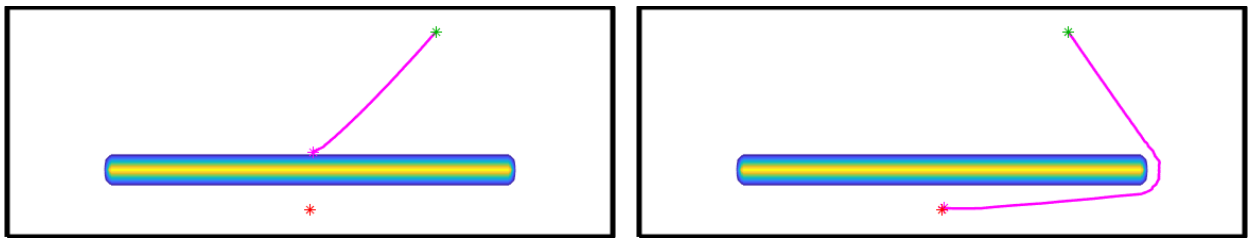
Figure 5.5: (a) - (c) Average path over 10000 trials (black), and three realizations of the stochastic equation of motion (colored) at different levels of uncertainty σ. (d) - (f) Average path (black) with standard deviation (grey), and the path computed using method (i) (dotted green).



(a) Optimal path given $T = 2000$ seconds.

(b) Optimal path given $T = 6800$ seconds.

Figure 5.6: Optimal paths in the vicinity of El Capitan with different terminal times.



(a) Optimal path given $T = 2$ seconds.

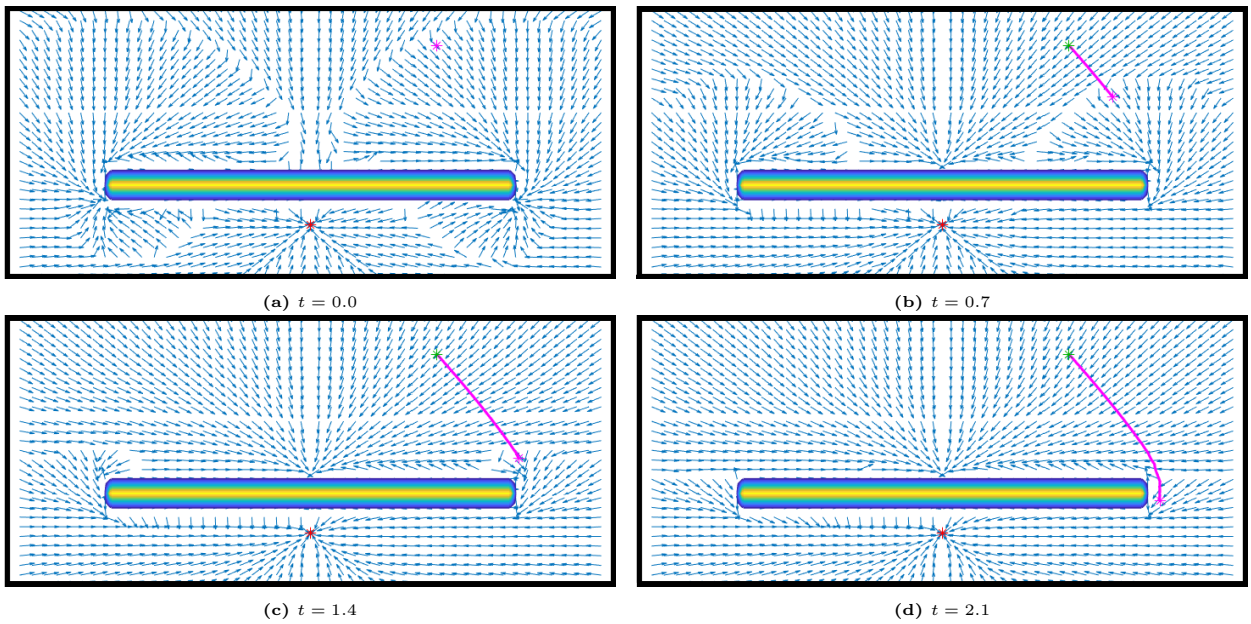
(b) Optimal path given $T = 4.25$ seconds.

Figure 5.7: Optimal paths using different end time values. The colored region is a wall.

$T \approx 2000$ seconds, the path simply walks to the cliff face and stays put. However, given $T \approx 6800$ seconds, the path descends the eastern slope and finds the desired end point as seen in fig. 5.6b

We can recreate this scenario using synthetic elevation data by placing a large wall directly between the starting point and end point as in fig. 5.7. The elevation is incredibly steep in the colored region, meaning that any optimal path would surely avoid the wall. In fig. 5.7b, where $T = 4.25$, this is exactly the behavior we observe; the path curves around the obstacle. However, in fig. 5.7a, where $T = 2$, the path walks toward the obstacle, stopping at the edge because velocity is near zero there.

Recall, our model constructs the path that ends as close (in Euclidean distance) to the desired end point as possible in the time allotted. When $T = 2$ in fig. 5.7a, there is not enough time to walk around the wall and instead, to get as close to the desired end point



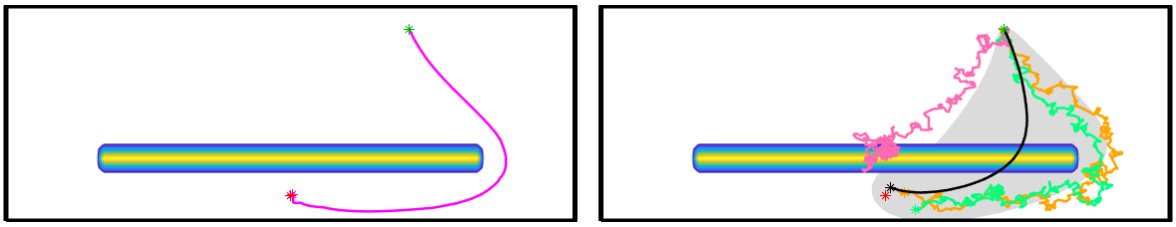
(a) $t = 0.0$ **(b)** $t = 0.7$ **(c)** $t = 1.4$ **(d)** $t = 2.1$

Figure 5.8: The discontinuity in the control value $\boldsymbol{a}^*(x,t)$ propagates as time increases

as possible, the path walks directly toward the wall. When situations like this arise, there is some critical amount of time $T^* > 0$ such that, given $T > T^*$, the path will walk around the obstacle, but given $T < T^*$, the path will walk toward the obstacle because it will not be able to get close enough to the end point if it walks around.

We can see this more explicitly if we plot the actual control values $\boldsymbol{a}^*(x,t)$ as well, as is done in fig. 5.8. In this example, the critical time is roughly $T^* = 3.4$, so we have plotted the path created by the algorithm with final time of $T = 3.5$, but we have plotted the path at times $t = 0, 0.7, 1.4, 2.1$. The arrows in the pictures are the values of $\boldsymbol{a}^*(x,t)$. Notice that in fig. 5.8a, there appears to be a discontinuity in the optimal control value. The deciding factor for whether the path will walk around the wall or walk toward the wall is where the starting point lies relative to this discontinuity. As time advances, the discontinuity in $\boldsymbol{a}^*(x,t)$ propagates, and since the starting point lies below the discontinuity, the path follows the arrows and walks around the obstacle. In the case when $T = 2$, the starting point is above the discontinuity, and thus the path walks toward the obstacle, rather than around it.

Discontinuities in $\boldsymbol{a}^*(x,t)$ are to be expected and relate to non-uniqueness of the optimal path. If x_0 lay directly on the discontinuity in $\boldsymbol{a}^*(x,0)$, then either walking around the obstacle or toward it would be equally optimal, since both would result in a path that



(a) Method (i), $\sigma = 0.3$, $T = 4.75$.

(b) Method (ii), average path, standard deviation and three realizations.

Figure 5.9: Optimal paths around the wall with uncertainty.

ends the same distance from the desired end point. Mathematically, one reason to expect discontinuities in $\boldsymbol{a}^*(x,t)$ is because $\boldsymbol{a}^*(x,t)$ is closely related to the gradient $\nabla\phi(x,t)$ of the solution to the HJB equation. Indeed, as discussed in section 2.3.4, in the case of isotropic motion, we have the explicit relationship $\boldsymbol{a}^*(x,t) = -\nabla\phi(x,t)/\left|\nabla\phi(x,t)\right|$. When the motion is anisotropic, as is the case in our model, the relationship between $\boldsymbol{a}^*(x,t)$ and $\nabla\phi(x,t)$ is no longer so explicit, but we can still anticipate that discontinuities in $\nabla\phi(x,t)$ will give rise to discontinuities in $\boldsymbol{a}^*(x,t)$. We will discuss this further in chapter 6 in the context of simple self-driving cars, where we give examples of non-unique optimal paths.

When we add uncertainty, the path planning strategy becomes more greedy, walking directly toward the end point and adjusting to avoid obstacles as is seen in fig. 5.1. When there is a wall, this strategy is costly because if one walks toward the wall, there may be insufficient time to adjust the route, and thus the critical time T^* required to walk around the wall increases rapidly. This is why the large increase in T is necessary in the example of El Capitan in fig. 5.4. We observe the same behavior in this synthetic example with the wall, though the increase in T as not as pronounced as in the case of El Capitan. In fig. 5.9a, we set $\sigma = 0.3$ and notice that to wrap around the wall and reach the end point, the optimal path computed using method (i) requires an end time of $T = 4.75$ rather than $T = 4.25$ in the deterministic case. In fig. 5.9b, we use method (ii), computing the average path over 10000 trials, and the path cuts through the wall since enough individual paths were pushed off course due to the random perturbations, as is seen with the pink path in the figure. As in fig. 5.5f, an individual could not realistically traverse the average path, since the wall is impassable. Thus, it seems that method (i) gives to a more practical result.

The dependence of the model on the parameter T is a major qualitative difference between this optimal path planning model and the model presented in chapter 4. That model deals only with the fully deterministic case, and uses a level set formulation wherein level sets representing optimal travel evolve outward from the starting point, and the terminal time T is defined as the time required for the level sets to envelop the end point. However, when we add uncertainty to the model, we introduce diffusion in the HJB equation and lose the level set intepretation of the equation. Thus while the model in chapter 4 has the advantage of not depending on T, this model is more generally applicable.

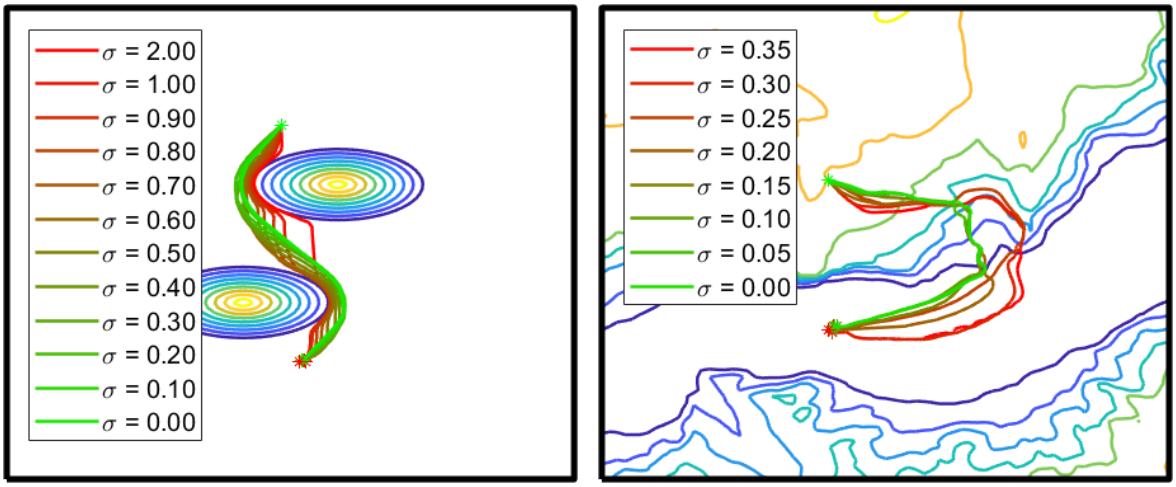
5.4.4 Convergence to the Deterministic Path as $\sigma \searrow 0$

As stated in sections 2.2,5.2.2, given mild regularity conditions on our Hamiltonian, the solution to the stochastic HJB equation (5.19) will converge to the viscosity solution to the ordinary HJB equation (5.3) as $\sigma \searrow 0$. We can see this empirically, not by observing the solution itself, but by examining the optimal path suggested by our algorithm at different levels of σ. This is shown in fig. 5.10. Here we have plotted many paths on the same figure, each computed using method (i) with a different σ value. As before, paths plotted in green were computed using smaller σ values, and those in red were computed using larger σ values. In fig. 5.10a, we see a very clear color gradient: the red paths computed with larger σ clearly tend toward the green path as σ decreases to zero. In fig. 5.10b, this is less obvious, especially since, for larger σ, the path takes a qualitatively different route. However, we do see that for smaller σ (greener paths), there is a tendency toward the deterministic optimal path.

5.5 Conclusions & Future Work

Path planning algorithms have wide-reaching applications in self-driving vehicles, reach-avoid games, pedestrian flow modeling and many other areas. Many previous models for path planning are completely deterministic, while in reality stochastic effects may be present and can significantly alter the motion along the path.

In this chapter, we developed a method for optimal path planning of human walking



(a) Convergence of paths with synthetic elevation data.

(b) Convergence of paths at El Capitan.

Figure 5.10: As $\sigma \searrow 0$, the stochastic optimal path converges back to the deterministic optimal path.

paths in mountainous terrain using a control theoretic approach and a Hamilton-Jacobi-Bellman (HJB) equation and allowing for uncertainty in the controlled equation of motion. The walking speed in our model depends on local slope in the direction of travel, giving rise to an anisotropic control problem. In the HJB equation, the uncertainty presents itself in the form of diffusion, leading to a viscous Hamilton-Jacobi-type equation. We suggest numerical methods for solving these equations, opting for a semi-implicit numerical scheme with a minimally diffusive numerical Hamiltonian, since any spurious numerical diffusion could be intepreted as nontrivial amounts of uncertainty in the equation of motion. After solving the HJB equation numerically, we suggest two methods for resolving the optimal path. First, we use the optimal control values resolved via the stochastic HJB equation, but simulate a deterministic equation of motion. This could simulate a person walking through a dark room or a dense forest, wherein they are cognizant of some uncertainty as they are planning the route, but do not feel random perturbations in velocity as they walk along a path. Second, we integrate the stochastic differential equation many times and arrive at a single path by averaging the results. This could model scenarios such as underwater unmanned vehicles, wherein the traveler actually feels the stochastic effects on the travel velocity.

We test our algorithm, including both methods for resolving the path, with synthetic elevation data first, and then with real elevation data in the area surrounding El Capitan. We compare these two notions of optimal path, and conclude that in the case of real elevation

data or impassable barriers, the first notion gives a more meaningful result. We also observe that in these cases, there will be discontinuities in the optimal control parameter and, especially in the presence of large walls, the position of these discontinuities can determine the walking strategy. Finally, we simulate the model at different levels of uncertainty in the equation of motion and observe that as uncertainty tends to zero, the optimal path path suggested by the model converges back to the deterministic optimal path.

One avenue of future work, which is explored in the ensuing chapter, would be to use similar methods to model path planning when the motion is more complicated. For example, here the motion is anisotropic, but is still simplified in that the control variable $a \in \mathbb{S}^1$ changes instantaneously. In many realistic path planning scenarios, omnidirectional motion is difficult or impossible, and the orientation of motion needs to be represented by a state variable, rather than a control variable.

In another direction, while this model considers uncertainty and suggests an expected optimal path, it has the shortcoming that there is no quantification of what "expected" optimal path means. One avenue for future work would be to update the model to answer probabilistic questions in more quantitative ways. For example, if there is uncertainty in the model, can we construct the set of points that one could *probably* reach by a given time? Under a given amount of uncertainty, what is the set of points one has a %50 chance of reaching? Similarly, if a hiker is observed at point A at time 0, and at point B at time $T > 0$, can we put a probability distribution on the paths linking A to B and requiring time less than T to traverse in order to determine which path the hiker probably followed? Questions like this most likely require significant updates to the model and stronger analytic tools.

CHAPTER 6

Time-Optimal Path Planning for Simple Self-Driving Cars

The previous three chapters present level set and control theoretic models for human movement. One feature of human navigation is that, at least in an approximate sense, walking direction can change instantaneously. A hiker can stop, pivot, and walk in a different direction than before. This is in contrast to even the most simple vehicles, such as a unicycle, where there is some notion of orientation of the vehicle that determines the direction of motion. The Hamilton-Jacobi-Bellman formulation for optimal path planning is fully capable of modeling this type of motion, and in this chapter we give one such example.

6.1 Introduction

As autonomous vehicle technology becomes more and more prevalent, it is important to develop robust and widely applicable trajectory planning algorithms. Many such vehicles—planetary exploration rovers [Tom05], flying drones [NME19], or remote-controlled submarines [AP01]—are subject to motion constraints which are *nonholonomic*, depending not only on the configuration, but the velocity of the vehicle. Accordingly, much effort has been devoted to trajectory planning for general nonholonomic mechanical systems [CGB15, GZ18, VF18].

One important example of a nonholonomic vehicle is a simple self-driving car. To track the motion of such a car, we model the current configuration using variables (x, y, θ): the spatial coordinate (x, y) is the position of the center of mass of the car, and the orientation θ is the angle between the rear wheels and the horizontal, increasing in the counterclockwise

direction. The car drives using actuators attached to the rear wheels that supply torque to each wheel individually, and steers using some mechanism separate from the rear wheels. The car has a rear axel of length $2R$, and a distance d between the center of the rear axel and the center of mass, as pictured in fig. 6.1. The motion of the car is constrained by a minimum turning radius which is equivalent to bounding the angular velocity $|\dot{\theta}| \leq W$. This bound could be resolved in terms of d, R and other parameters inherent to the steering mechanism. Because of this, models like those in chapters 4,5, where the direction of motion is the control variable cannot appropriately describe the navigation here.

The problem of path planning for simple self-driving cars goes back to Dubins [Dub57] who envisioned a vehicle that could move forward along paths constrained by a minimum turning radius. Later, Reeds and Shepp [RS90] generalized the Dubins car to one that could also reverse direction. In both these cases, the problem was analyzed in a geometric and combinatorial fashion, discretizing the path into regions of straight-line movement and arcs of circles. The paths were designed to minimize length, and no obstacles were considered. Barraquand and Latombe [BL93] added obstacles to the model, and devised a method of growing a reachability tree outward from the desired final configuration. Based on similar analysis, Agarwal and Wang [AW01] assumed polygonal obstacles and presented an efficient algorithm for resolving paths that are robust to perturbation and nearly optimal.

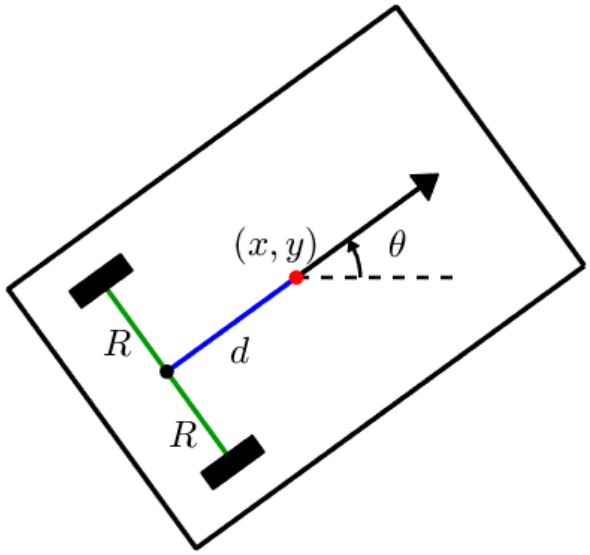


Figure 6.1: A simple self-driving car.

Since then, there has been increased effort to resolve optimal trajectories for such cars (and similar robots) using methods rooted in optimization and control. In this case, the nonholonomic constraint is

$$\dot{y}\cos\theta - \dot{x}\sin\theta = d\dot{\theta} \tag{6.1}$$

which ensures rolling without slipping and motion in the direction parallel to the rear wheels [BL93, FL98]. There have been several discrete and variational models of motion planning for these vehicles [FL98, AWS16, LO95, SSW13, WCW00]. One advantage of models based on optimization is that they can seemlessly account for paths that are not only time-optimal, but consider energy consumption as well [DS03, KNH19, WWW08, VDS08]. Models of this sort are often hierarchical, relying on a global path planner and a local collision avoidance algorithm [ABS18, LJO17].

Finally, a model for curvature constrained motion based on dynamic programming and a Hamilton-Jacobi-Bellman equation was introduced by [TTS10, TT13], where obstacles are included, but the car is simplified to a point mass. We present a model for optimal path planning of nonholonomic self-driving cars based on a Hamilton-Jacobi formulation, but considering the geometry of the vehicle. Our approach is akin to that of [TTS10, TT13]. However, those authors simplify the car to a point mass and accordingly must either create a buffer region around an obstacles [TTS10] or opt for a semi-Lagrangian path planning approach [TT13]. If we do not make the simplification, we can maintain the Hamilton-Jacobi approach. The Hamilton-Jacobi formulation has the natural advantage that it averts the need for hierarchical planning algorithms. Additionally, since our method resolves the value function representing optimal travel time, this approach can provide optimal trajectories from all starting configurations to a desired final configuration, as opposed to variational methods which typically resolve a single trajectory. We also present an upwind sweeping scheme that traces the characteristics outward from a desired final configuration.

This chapter is laid out as follows. In section 6.2, we discuss the Hamilton-Jacobi formulation for optimal path planning of the nonholonomic vehicle above. In section 6.3, we design an upwind, fast-sweeping method to solve the Hamilton-Jacobi-Bellman equation for

the travel time function. In section 6.4, we test our algorithm in the presence and absence of obstacles. Finally, in section 6.5 we discuss several avenues for future work on this and similar problems.

6.2 The Hamilton-Jacobi Formulation for Nonholonomic Cars

We discuss the control theoretic formulation of path planning for our simple car. This includes choosing a model for motion and a dynamic programming approach similar to those used in previous chapters.

6.2.1 Kinematics & Control Problem

The most honest model of the movement of a car would be a *dynamic* model that sums the forces on the car and determines movement via basic mechanics [PLM06]. In such a model, the control variables would be the torques τ_ℓ, τ_r that are supplied by the actuators to the left and right wheels, respectively. This will produce a maximal acceleration, which in turn, when taking into account drag and other damping forces proportional to velocity, translates into a maximal velocity. Accordingly, if the movement of the vehicle is all we are concerned with, we can neglect some of the dynamics, and opt for a *kinematic* model, wherein the control variables are the velocities v_ℓ, v_r of the left and right wheel, respectively [WCW00]. There is a bijective transformation between these velocities, and the tangential velocity v and normal (angular) velocity ω of the car: $v = (v_r + v_\ell)/2$, $\omega = (v_r - v_\ell)/2R$. It is most natural to describe the kinematics using these as control variables.

The equations of motion for the center of mass of the vehicle are

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{v}(t)\cos\boldsymbol{\theta}(t) - \boldsymbol{\omega}(t)Wd\sin\boldsymbol{\theta}(t), \\ \dot{\boldsymbol{y}}(t) &= \boldsymbol{v}(t)\sin\boldsymbol{\theta}(t) + \boldsymbol{\omega}(t)Wd\cos\boldsymbol{\theta}(t), \\ \dot{\boldsymbol{\theta}}(t) &= \boldsymbol{\omega}(t)W, \end{aligned} \tag{6.2}$$

where $W > 0$ is the bound on the angular velocity of the vehicle, and $\boldsymbol{v}(\cdot), \boldsymbol{\omega}(\cdot) \in [-1, 1]$. By

using these control values, we are implicitly assuming instantaneous changes in velocities, which is the same as assuming infinite acceleration.

We formulate a control problem for optimal paths using these kinematics. Suppose the car moves in some domain $\Omega \subset \mathbb{R}^2$ that is disjointly segmented into free space and obstacles: $\Omega = \Omega_{\text{free}} \cup \Omega_{\text{obs}}$. For any configuration $(x, y, \theta) \in \Omega \times [0, 2\pi)$ let $D(x, y, \theta) \subset \mathbb{R}^2$ denote the space occupied by the car. The shape could be arbitrary, but for our car, this will be a rectangle of height $2R$ and width $2d$ that is centered at (x, y) and then rotated by θ. We call a configuration *admissable* if $D(x, y, \theta) \cap \Omega_{\text{obs}} = \varnothing$. Next, suppose we are given a desired final configuration (x_f, y_f, θ_f) that is admissable. We call a trajectory $(\boldsymbol{x}(\cdot), \boldsymbol{y}(\cdot), \boldsymbol{\theta}(\cdot))$—defined for $t \in [0, T]$—*admissable* if

1. it obeys (6.2) for all $t \in (0, T]$,
2. $(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{\theta}(t))$ is an admissable configuration for all $t \in [0, T]$, and
3. $(\boldsymbol{x}(T), \boldsymbol{y}(T), \boldsymbol{\theta}(T)) = (x_f, y_f, \theta_f)$.

Given a starting point (x, y, θ), the goal is to choose $(\boldsymbol{v}(\cdot), \boldsymbol{\omega}(\cdot))$ so as to minimize travel time T among all admissable trajectories.

6.2.2 Value Function & Hamilton-Jacobi-Bellman Equation

Denote by $\mathcal{A}(x, y, \theta, T; x_f, y_f, \theta_f)$ the set of admissable trajectories beginning at the configuration (x, y, θ), ending at the configuration (x_f, y_f, θ_f) and requiring less time than T to traverse. The value function for this control problem is the optimal travel-time function:

$$\phi(x, y, \theta) = \inf_{\boldsymbol{v}(\cdot), \boldsymbol{\omega}(\cdot)} \{T \,:\, \mathcal{A}(x, y, \theta, T; x_f, y_f, \theta_f) \neq \varnothing\}. \tag{6.3}$$

That is, we are using the cost functional

$$C[\boldsymbol{v}(\cdot), \boldsymbol{\omega}(\cdot)] = T = \int_0^T 1 \, dt. \tag{6.4}$$

Because neither the equation of motion (6.2) nor the cost functional (6.4) depend explicitly on time, we can formulate a steady-state Hamilton-Jacobi-Bellman equation for this problem.

In this case, the dynamic programming principle takes the form

$$\phi(x, y, \theta) = \Delta t + \inf_{\boldsymbol{v}(\cdot), \boldsymbol{\omega}(\cdot)} \{\phi(\boldsymbol{x}(t + \Delta t), \boldsymbol{y}(t + \Delta t), \boldsymbol{\theta}(t + \Delta t))\}. \tag{6.5}$$

Intuitively, equation (6.5) states that traveling optimally for time Δt will decrease the remaining travel time by exactly Δt. Using the same arguments as before, we arrive at

$$-1 = \inf_{v,\omega} \left\{ \dot{\boldsymbol{x}}\phi_x + \dot{\boldsymbol{y}}\phi_y + \dot{\boldsymbol{\theta}}\phi_\theta \right\}. \tag{6.6}$$

Inserting the equation of motion (6.2) results in

$$-1 = \inf_{v,\omega} \{(v\cos\theta - \omega W d \sin\theta)\phi_x + (v\sin\theta + \omega W d\cos\theta)\phi_y + \omega W \phi_\theta\}, \tag{6.7}$$

where $v, \omega \in [-1, 1]$. Collecting terms with v and ω separately yields

$$-1 = \inf_{v,\omega} \{(\phi_x \cos\theta + \phi_y \sin\theta)v + W(-d\phi_x \sin\theta + d\phi_y \cos\theta + \phi_\theta)\omega\}. \tag{6.8}$$

Since the quantity being minimized is linear in v and ω, this results in a bang-bang controller

$$\begin{aligned} v &= -\text{sign}(\phi_x \cos\theta + \phi_y \sin\theta), \\ \omega &= -\text{sign}(-d\phi_x \sin\theta + d\phi_y \cos\theta + \phi_\theta), \end{aligned} \tag{6.9}$$

and ϕ solves the Hamilton-Jacobi-Bellman equation

$$1 = |\phi_x \cos\theta + \phi_y \sin\theta| + W\,|-d\phi_x \sin\theta + d\phi_y \cos\theta + \phi_\theta|\,. \tag{6.10}$$

This equation is paired with the boundary condition $\phi(x_f, y_f, \theta_f) = 0$. Note, this equation only holds at admissable configurations. The value assigned to any inadmissable configuration is $\phi(x, y, \theta) = +\infty$.

6.2.3 Small Time Local Controllability & Continuity of the Value Function

Again, the above computation only holds formally when ϕ is non-smooth. As long as ϕ remains continuous, it will be the unique viscosity solution of (6.10). Any value functions in the preceding chapters will be continuous under mild conditions on the input data. However, with nonholonomic robots, it is easy to devise realisitic models of motions that give rise to discontinuous value functions. For example, Dubins considered a self-driving car with $d = R = 0$ and $v \equiv 1$ so that the car only drives forward [Dub57]. Suppose the minimum turning radius is 1 and let ϕ^* be the minimum travel time function for the Dubins car with ending configuration $(x_f, y_f, \theta_f) = (0, 1, \pi)$. Then $\phi^*(1, 0, \pi/2) = \pi/2$ since the optimal path only requires traversing a quarter circle. However for any $x \in (0, 1)$, we have $\phi^*(x, 0, \pi/2) \geq 3\pi/2$ since the car will need to traverse at least three quarters of a circle to orient itself correctly. The minimum travel time function will always be lower-semicontinuous [KQ01], and in the case that it is discontinuous, one can define a weaker notion of solution and maintain existence and uniqueness for the Hamilton-Jacobi-Bellman equation [CS03, TT13].

For nonholonomic motion, continuity of the value function is intimately tied with a property called *small time local controllability*. For a configuration (x, y, θ) and a time $t > 0$, define the reachability set

$$\Sigma(x, y, \theta, t) = \{(x_0, y_0, \theta_0) \ : \ \mathcal{A}(x, y, \theta, t; x_0, y_0, \theta_0) \neq \varnothing\}. \tag{6.11}$$

This represents the set of points that can be reached after starting from (x, y, θ) and traveling for time t. The motion is said to be small time locally controllable (STLC) at (x, y, θ) if

$$(x, y, \theta) \in \operatorname{int} \Sigma(x, y, \theta, t) \quad \text{for all } t > 0. \tag{6.12}$$

Roughly speaking, the vehicle is STLC if the time required to steer to configurations in an ε-neighborhood of the current configuration is no greater than $O(\varepsilon)$. The Dubins car is not STLC; because it can only drive forward, to achieve a position slightly behind the current position, it will need to traverse an entire circle. Thus, for example, in the case

of the Dubins car described above, $(x, y, \theta) \notin \operatorname{int} \Sigma(x, y, \theta, 1)$. Conversely, when the car is allowed to move forward and backward, it is STLC [ST91], and when the motion is STLC, the value function remains continuous [TT13]. In general, there is an important interplay between small time local controllability and optimal trajectory planning, and accordingly, much effort has been devoted to analysis of STLC conditions for broad classes of control problems [Kaw90, Kra98, KQ01, AL12, Jaf20].

6.3 Numerical Methods

To solve (6.10) numerically, we would like to develop a sweeping scheme similar to those in [TT13, TCO03]. The primary concerns for such a scheme are that it should be upwind and monotone [CL83, CM80, Obe06, OS91]. We note that from (6.9), there are a finite number of values assumed by v, ω. We will design an update scheme that accounts for each pair and decides which pair to use depending on which case we fall into. Since there is no incentive for the car to stop moving, it is sufficient to consider $v \in \{-1, 1\}$. However, if the car is already oriented in the optimal direction, then normal velocity should be zero, and hence we need to allow $\omega \in \{-1, 0, 1\}$. Analytically, the case that $\omega = 0$ corresponds precisely to $-d\phi_x \sin\theta + d\phi_y \cos\theta + \phi_\theta = 0$. However, due to numerical error, we almost never have $-d\phi_x \sin\theta + d\phi_y \cos\theta + \phi_\theta = 0$ computationally, which means this case should be handled separately. For the Reeds-Shepp car [RS90] (when $d = 0$), one must discritize $|\phi_\theta|$, and interestingly, using the standard monotone discretization $|\phi_\theta| \approx -\min\{\phi_\theta^+, \phi_\theta^-, 0\}$, the case when $\omega = 0$ corresponds to the case that the minimum chooses 0 [TT13].

6.3.1 An Upwind Sweeping Scheme for (6.10)

For simplicity, we describe the numerics on a uniform discretization of $[a, b] \times [c, d] \times [0, 2\pi]$, though in principle, the scheme could be adapted to non-standard grids. Fixing $I, J, K \in \mathbb{N}$,

let

$$\begin{aligned} x_i &\coloneqq a + i\Delta x, \quad \Delta x = \frac{b-a}{I}, \quad i = 0, 1, \ldots, I, \\ y_j &\coloneqq c + j\Delta y, \quad \Delta y = \frac{d-c}{J}, \quad j = 0, 1, \ldots, J, \\ \theta_k &\coloneqq k\Delta\theta, \qquad \Delta\theta = \frac{2\pi}{K}, \qquad k = 0, 1, \ldots, K. \end{aligned} \tag{6.13}$$

Let ϕ_{ijk} be the numerical approximation to $\phi(x_i, y_j, \theta_k)$. To discretize (6.10) in a fully upwind manner, suppose that the particular pair $(v, \omega) \in \{-1, 1\} \times \{-1, 0, 1\}$ represents the correct control values at (x, y, θ). Then from (6.7), the equation reads

$$-1 = (v\cos\theta - \omega W d\sin\theta)\phi_x + (v\sin\theta + \omega W d\cos\theta)\phi_y + \omega W\phi_\theta. \tag{6.14}$$

Accordingly, we define

$$\begin{aligned} A_k(v,\omega) &= |v\cos\theta_k - \omega W d\sin\theta_k|\,, \\ B_k(v,\omega) &= |v\sin\theta_k + \omega W d\cos\theta_k|\,, \\ a_k(v,\omega) &= \operatorname{sign}(v\cos\theta_k - \omega W d\sin\theta_k), \\ b_k(v,\omega) &= \operatorname{sign}(v\sin\theta_k + \omega W d\cos\theta_k). \end{aligned} \tag{6.15}$$

Then the upwind approximation to each derivative term in (6.14) at (x_i, y_j, θ_k) is given by

$$\begin{aligned} \big([v\cos\theta - \omega W d\sin\theta]\phi_x\big)_{ijk} &= A_k(v,\omega)\left(\frac{\phi_{i+a_k(v,\omega),j,k} - \phi_{ijk}}{\Delta x}\right), \\ \big([v\sin\theta + \omega W d\cos\theta]\phi_y\big)_{ijk} &= B_k(v,\omega)\left(\frac{\phi_{i,j+b_k(v,\omega),k} - \phi_{ijk}}{\Delta y}\right), \\ (\omega W\phi_\theta)_{ijk} &= |\omega|\, W\left(\frac{\phi_{i,j,k+\operatorname{sign}(\omega)} - \phi_{ijk}}{\Delta\theta}\right). \end{aligned} \tag{6.16}$$

Inserting these into (6.7) or (6.10), we can isolate ϕ_{ijk}. Indeed, at internal grid nodes, the discrete equation reads

$$-1 = A_k(v,\omega)\left(\frac{\phi_{i+a_k(v,\omega)} - \phi_{ijk}}{\Delta x}\right) + B_k(v,\omega)\left(\frac{\phi_{j+b_k(v,\omega)} - \phi_{ijk}}{\Delta y}\right) + |\omega|\, W\left(\frac{\phi_{k+\operatorname{sign}(\omega)} - \phi_{ijk}}{\Delta\theta}\right), \tag{6.17}$$

where any missing index is i, j or k. This shows that

$$\phi^*_{ijk}(v,\omega) = \frac{1 + \frac{A_k(v,\omega)}{\Delta x}\phi_{i+a_k(u,v),j,k} + \frac{B_k(v,\omega)}{\Delta y}\phi_{i,j+b_k(v,\omega),k} + \frac{|\omega|W}{\Delta\theta}\phi_{i,j,k+\mathrm{sign}(\omega)}}{\frac{A_k(v,\omega)}{\Delta x} + \frac{B_k(v,\omega)}{\Delta y} + \frac{|\omega|W}{\Delta\theta}} \tag{6.18}$$

is a first-order upwind approximation of (6.10) in the case that (v,ω) attain the infima in (6.7). This approximation and the "boundary" conditions—$\phi(x_f, y_f, \theta_f) = 0$ and $\phi(x,y,\theta) = +\infty$ at inadmissable configurations—suggest the following sweeping scheme, which we summarize here and detail explicitly in algorithm 1.

We initialize ϕ^0_{ijk} to be zero at the nodes surrounding (x_f, y_f, θ_f) and set $\phi^0_{ijk} = +\infty$ at other nodes. Next, for $n = 1, 2, 3, \ldots$, we advance from ϕ^{n-1}_{ijk} to ϕ^n_{ijk} by sweeping through the indices (i,j,k), computing $\phi^*_{ijk}(v,\omega)$ according to (6.18) and taking the minimum over (v,ω). We sweep through the indices in alternating directions, backward and forward, until all combinations of sweeping directions have been covered. There should a total of 8 sweeps corresponding to [i-forward,j-forward,k-forward], [i-forward,j-forward,k-backward], [i-forward,j-backward,k-forward], etc. To ensure monotonicity of the scheme, we only update ϕ^n_{ijk} if the value suggested by (6.18) is less than ϕ^{n-1}_{ijk}. After completing the 8 sweeps, we enforce periodicity in θ (ensuring that $\phi^n_{i,j,0} = \phi^n_{i,j,K}$ for all n). We repeat this scheme until the norm of successive iterations $\|\phi^n - \phi^{n-1}\|$ is smaller than some specified tolerance, indicating convergence. This algorithm returns the values ϕ_{ijk}, which approximate $\phi(x_i, y_j, \theta_k)$, as well as the optimal control values v_{ijk}, ω_{ijk} which can be resolved during the sweeping as detailed below.

After solving (6.10) via algorithm 1, one can compute the optimal path by discretizing time and evolving (6.2) beginning at $(\boldsymbol{x}(0), \boldsymbol{y}(0), \boldsymbol{\theta}(0)) = (x_0, y_0, \theta_0)$ until $(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{\theta}(t))$ are sufficiently close to (x_f, y_f, θ_f). The optimal travel time is given theoretically by $\phi(x_0, y_0, \theta_0)$ and thus should be approximated by ϕ_{i_0,j_0,k_0} where $(x_{i_0}, y_{j_0}, \theta_{k_0})$ is the nearest grid point to (x_0, y_0, θ_0).

We include several practical implementation notes. First, referring to algorithm 1, it is important that we begin the while-loop by assigning $\phi^n_{ijk} \leftarrow \phi^{n-1}_{ijk}$, and operate only with ϕ^n_{ijk} in the ensuing computations. This way, as we sweep through and ϕ^n_{ijk} is updated, we are

Algorithm 1 A sweeping scheme to solve (6.10)

Initialization: Input a desired ending pose (x_f, y_f, θ_f), a grid discretization as in (6.13) and a small error tolerance $\varepsilon > 0$. Initialize $\phi^0_{ijk} = 0$ for the grid nodes immediately surrounding (x_f, y_f, θ_f) and $\phi^0_{ijk} = +\infty$ for all other grid nodes. Initialize v^0_{ijk} and ω^0_{ijk} arbitrarily. Initialize $\phi^1_{ijk} = 0$ at all grid poits and set $n = 1$. Sweep through all indices (i, j, k) and record the admissable indices (those corresponding to admissible configurations).

while $\|\phi^n - \phi^{n-1}\| > \varepsilon$ **do**

Assign $\phi^n_{ijk} \leftarrow \phi^{n-1}_{ijk}$ for all (i, j, k).

for $i = 1$ **to** $I - 1$ **do**
for $j = 1$ **to** $J - 1$ **do**
for $k = 1$ **to** K **do**
if (i, j, k) is admissable **then**

For each pair $(v, \omega) \in \{\pm 1\} \times \{0, \pm 1\}$, compute $\phi^*_{ijk}(v, \omega)$ according to (6.18).

Assign $\phi^n_{ijk} \leftarrow \min\{\min_{v,\omega} \phi^*_{ijk}(v, \omega), \phi^{n-1}_{ijk}\}$

if $\phi^n_{ijk} = \phi^{n-1}_{ijk}$ **then**
Assign $(v^n_{ijk}, \omega^n_{ijk}) \leftarrow (v^{n-1}_{ijk}, \omega^{n-1}_{ijk})$
else
Assign $(v^n_{ijk}, \omega^n_{ijk}) \leftarrow \text{argmin}_{v,\omega} \phi^*_{ijk}(v, \omega)$
end if
end if
end for
end for
end for

Repeat the above **for** loops, sweeping in alternating directions until all combinations of sweeping directions have been completed (a total of 8 sweeps).

Enforce periodicity in θ: $\phi^n_{i,j,0} \leftarrow \phi^n_{i,j,K}$ for all (i, j)
Assign $n \leftarrow n + 1$

end while

Assign $\phi_{ijk} \leftarrow \phi^n_{ijk}$, $v_{ijk} \leftarrow v^n_{ijk}$, $\omega_{ijk} \leftarrow \omega^n_{ijk}$

return $\phi_{ijk}, v_{ijk}, \omega_{ijk}$ for all (i, j, k)

constantly using the newest information, in the same manner as in Gauss-Seidel iteration [KOQ04]. For example, if we are sweeping forward in i, we will first compute $\phi^n_{1,j,k}$ and then use this newly computed values to resolve $\phi^n_{2,j,k}$. Second, some care needs to be taken with the "boundary" nodes in k. When $k = K$, we could have $k + \text{sign}(\omega) = K + 1$, but this index should be identified with $k = 1$ by periodicity. Third, to slightly reduce the computational load, one can pre-compute $A_k(v,\omega), B_k(v,\omega), a_k(v,\omega), b_k(v,\omega)$ and also exploit the symmetries $A_k(-v,-\omega) = A_k(v,\omega), a_k(-v,-\omega) = -a_k(v,\omega)$ (and similarly for $B_k(v,\omega), b_k(v,\omega)$) to avoid looking up redundant values. Fourth, while algorithm 1 suggests a method for resolving the optimal control values v, ω during the sweeping scheme; this only needs to be done during the final sweep, not in each of the 8 sweeps. Alternatively, one can ignore these values during the sweeping and use the values of ϕ_{ijk} to compute the optimal control values from (6.9) afterwards. If one opts for this method, special consideration should be made for the case that $\omega = 0$; for example $\omega = 0$ when $|-d\phi_x \sin\theta + d\phi_y \cos\theta + \phi_\theta| < \delta$ for some small δ, and $\omega = \text{sign}(-d\phi_x \sin\theta + d\phi_y \cos\theta + \phi_\theta)$ otherwise. Fifth, when initializing ϕ^0_{ijk}, one will likely need to choose a large positive number for the grid nodes away from (x_f, y_f, θ_f) instead of using $+\infty$. This number is arbitrary but needs to be larger than any actual value of $\phi(x,y,\theta)$ in the domain. Sixth, one may notice that the nodes at the computational boundary ($i = 0, I$ and $j = 0, J$) are never updated and thus will remain large. Practically, this forces the car to remain in the computational domain. Further, since the scheme is fully upwind and characteristics flow out of the boundary, there is no danger of this affecting the solution at the interior nodes (the sweeping scheme will naturally "choose" not to use the boundary nodes). Thus there is no need for special consideration at the boundary as was necessary in section 5.3, and is generally necessary when using diffusive schemes [KOQ04]. Seventh, the norm one uses to evaluate convergence is somewhat arbitary since the computational solution space is finite dimensional. However, in light of the ensuing notes, we suggest the supremum norm: $\|\phi^n - \phi^{n-1}\| = \sup_{ijk} \left|\phi^n_{ijk} - \phi^{n-1}_{ijk}\right|$. Eighth, the algorithm returns an approximation to the value function at grid nodes corresponding to the entire computational domain $[a,b]\times[c,d]\times[0,2\pi]$. Thus the value function can be used to determine the optimal path from any point in the domain to the final configuration (x_f, y_f, θ_f), which

is to say that the initial configuration (x_0, y_0, θ_0) is not an integral part of the model. If the goal is only to resolve the path from (x_0, y_0, θ_0) to the final configuration, one does not need to wait for convergence of ϕ^n_{ijk} at all nodes (i, j, k), but rather only at the node (i_0, j_0, k_0) closest to the inital configuration. Empirically, the values ϕ^n_{ijk} at any nodes corresponding to configurations visited along the path between (x_0, y_0, θ_0) and (x_f, y_f, θ_f) will converge before the value of $\phi^n_{i_0,j_0,k_0}$. This makes intuitive sense since characteristics are flowing outward from (x_f, y_f, θ_f). Thus the terminal condition can be reduced to $\left|\phi^n_{i_0,j_0,k_0} - \phi^{n-1}_{i_0,j_0,k_0}\right| < \varepsilon$ (perhaps performing one extra sweep for safety, since the grid point $(x_{i_0}, y_{j_0}, \theta_{k_0})$ is only an approximation of (x_0, y_0, θ_0)). Ninth and finally, in light of the previous observation, to reduce computational load, we can remove nodes (i, j, k) from the sweeping when the value of ϕ^n_{ijk} has converged.

6.3.2 Proof of Concept: Eikonal Equation

In principle a sweeping scheme analogous to the one above could be developed for any equation of the form

$$r(x) = \inf_a \left\{ \sum_{\ell=1}^{d} f_\ell(x, a) u_{x_\ell} \right\}. \tag{6.19}$$

For example, this includes any Hamilton-Jacobi-Bellman equation for a time-independent control problem where the running cost does not depend explicitly on the control variables. Two difficulties may arise when running such a scheme: evaluating the infimum on the right hand side may be non-trivial, and dimensions $d > 3$ are susceptible to the curse of dimensionality. However, as a proof of concept, we can run this same algorithm for the two dimensional steady-state Eikonal equation. Suppose that $\phi : \mathbb{R}^2 \to \mathbb{R}$ solves

$$1 = |\nabla \phi| \quad \text{in } \mathbb{R}^2 \tag{6.20}$$

and $\phi(0, 0) = 0$. As stated in section 2.3.4, this can be realized as the Hamilton-Jacobi equation for the minimum travel time function, when the equation of motion is $\dot{\boldsymbol{x}} = \boldsymbol{a}$ where $\boldsymbol{a}(\cdot) \in \mathbb{S}^1$ is the control variable. By analysis similar to that in sections 2.2, 2.3, the exact

solution to this equation is given by $\phi(x, y) = \sqrt{x^2 + y^2}$.

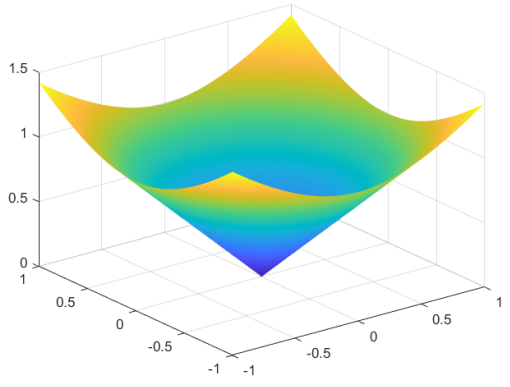
Re-writing (6.20) in the form

$$-1 = \inf_{a \in [0,2\pi]} \{\phi_x \cos(a) + \phi_y \sin(a)\} \tag{6.21}$$

shows that it is amenable to our algorithm. In this case, the upwind discretization takes the form

$$\phi^*_{ij}(a) = \frac{1 + \frac{|\cos(a)|}{\Delta x}\phi_{i+\text{sign}(\cos(a)),j} + \frac{|\sin(a)|}{\Delta y}\phi_{i,j+\text{sign}(\sin(a))}}{\frac{|\cos(a)|}{\Delta x} + \frac{|\sin(a)|}{\Delta y}}, \tag{6.22}$$

and we can run the same algorithm as before. Results of this simulation are included in fig. 6.2. The computational domain is $[-1, 1]^2$, and we run the simulation with increasingly fine discretization. The table in fig. 6.2b shows convergence at order slightly less than 1. The discretization (6.21) is first order accurate. However, error estimates for numerical solutions of Hamilton-Jacobi equations depend not only on the order of approximation, but also on the regularity of the solution. Classical results establish convergence at order 1/2 if the discretization is first order accurate and the solution remains Lipschitz continuous [CL84, Sou85]. It is difficult to improve on the order of convergence, though one can improve the accuracy by considering additional approximations to the derivatives of ϕ along different directions [TTS10, Par20]



(a) Numerical Solution with $I = J = 1280$ grid points.

I, J	Error ($\|\cdot\|_\infty$)	Conv.
10	0.1758	—
20	0.0970	0.8574
40	0.0556	0.8024
80	0.0322	0.7886
160	0.0187	0.7870
320	0.0107	0.7993
640	0.0061	0.8165
1280	0.0034	0.8335

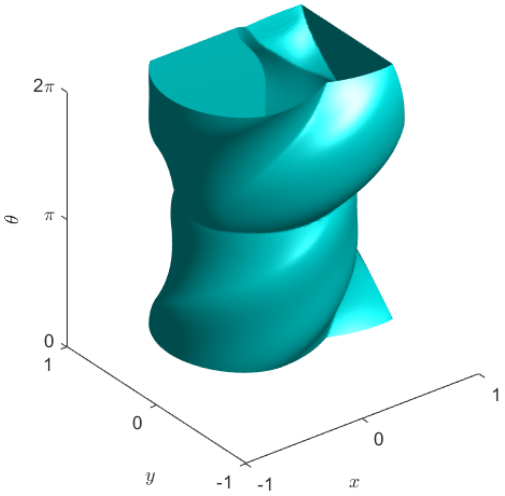
(b) Convergence table.

Figure 6.2: Application of our algorithm to the two-dimensional Eikonal equation.

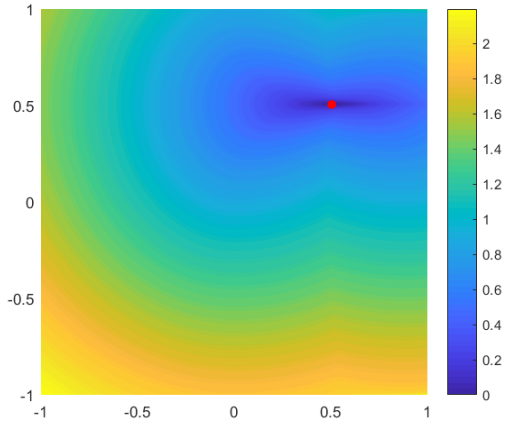
6.4 Results & Observations

We tested our algorithm using the spatial domain $\Omega = [-1, 1] \times [-1, 1]$. In our tests, we set $R = 0.04$, $d = 0.07$ and $W = 4$, meaning that the minimum radius of a circle that the car can traverse is $1/4$. Note these are all dimensionless parameters, used solely for testing purposes. In our simulations, we used a uniform discretization with 200 grid nodes in each direction. Depending on the simulation, the sweeping scheme required roughly 25 iterations to converge with tolerance $\varepsilon = 10^{-4}$, though it took longer in simulations with obstacles.

In the first simulation, we computed the value function $\phi(x, y, \theta)$ for the final orientation is $(x_f, y_f, \theta_f) = (\frac{1}{2}, \frac{1}{2}, 0)$ with no obstacles. Figure 6.3a displays the isocontour $\{(x, y, \theta) \in \Omega \times [0, 2\pi] \ : \ \phi(x, y, \theta) = 1\}$. One interesting note here is the approximate symmetry across the line $\theta = \pi$, which is an expression of periodicity in the value function. When $d = 0$, we do indeed have π-periodicity: $\phi(x, y, \theta) = \phi(x, y, \theta + \pi)$ [TT13]. When $d \neq 0$, this is only approximate. Figure 6.3b shows a contour map of the function $\phi(x, y, 0)$, which gives the travel time from different starting positions if the car is already facing in the positive x-direction. The final position $(x_f, y_f) = (\frac{1}{2}, \frac{1}{2})$ is represented by the red dot. As a sanity check, we note that along the line $y = \frac{1}{2}$, the value is given by $\phi(x, \frac{1}{2}, 0) = |x - x_f|$ since the optimal path merely includes pulling forward or reversing into the spot.



(a) The isocontour $\phi(x, y, \theta) = 1$, $(x_f, y_f, \theta_f) = (\frac{1}{2}, \frac{1}{2}, 0)$.



(b) The contour map of $\phi(x, y, 0)$, $(x_f, y_f, \theta_f) = (\frac{1}{2}, \frac{1}{2}, 0)$.

Figure 6.3: Visualization of the travel time function.

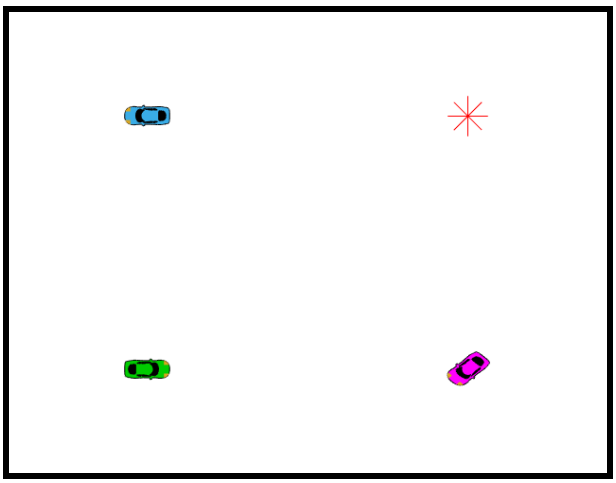
We used the value function from fig. 6.3b to compute optimal paths. Figure 6.4 displays optimal paths from three different starting configurations to the final configuration $(x_f, y_f, \theta_f) = (\frac{1}{2}, \frac{1}{2}, 0)$. The final location is marked by the red star, and the initial locations are marked by colored dots. This figure shows that the best strategies for the blue and pink car involve traveling large portions of the path in reverse, before pivoting and achieving the final configuration while moving in the forward direction. By contrast, the green car travels only forward. Note, these optimal paths were computed independently and are simply plotted on top of each other; the paths will require different amounts of time to traverse and there is no interaction between the cars.

Reeds and Shepp [RS90] analyzed this problem in the case that $d = 0$ so that the car is a point mass. They proved that the optimal path between two points consists of a finite number of straight lines and arcs of circles of minimum radius. Further, they proved that while kinks will occur as the car switches driving direction (as seen in the path of the blue car in fig. 6.4), the optimal path requires no more than two kinks. Our simulations empirically confirm this; in the examples in fig. 6.4, none of the paths required more than one kink. For an example of an optimal path with two kinks, consider the parallel parking problem displayed in fig. 6.5. In this example $(x_0, y_0, \theta_0) = (x_f + 2d, y_f + 3R, \theta_f)$ and the car is plotted at four points along the path: the initial position, the two kinks, and the final position.

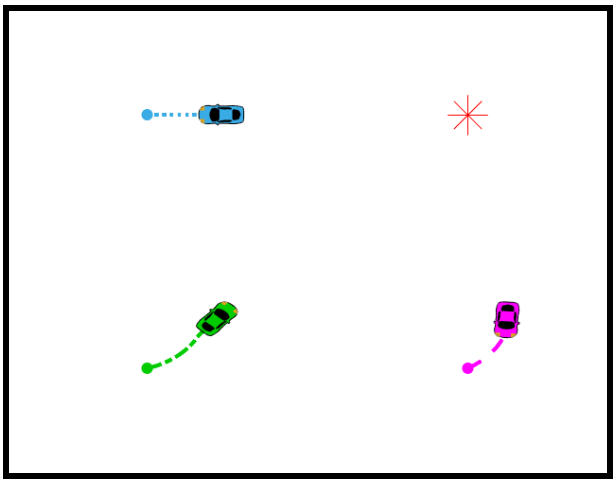
An interesting phenomenon in optimal path-planning is that optimal paths are are not always unique. We see this even in the simplest path-planning models. For example, consider the case of a particle moving isotropically through a velocity field $\nu(x, y) \geq 0$ in two dimensions. Following the computation in section 2.3.4, the optimal travel time function ϕ for this problem satisfies the steady-state Eikonal equation

$$\nu(x, y) \left|\nabla \phi(x, y)\right| = 1, \quad \phi(x_f, y_f) = 0 \tag{6.23}$$

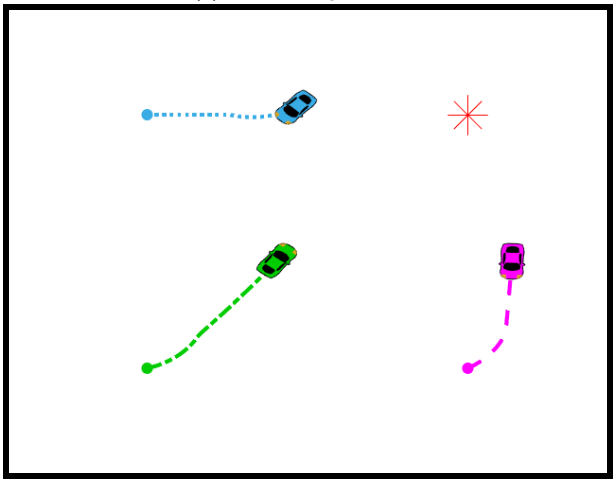
where (x_f, y_f) is the desired end point, and the optimal control value is given by the normal to the level sets. It is easy to design ν so as to produce non-unique optimal paths. In fig. 6.6, we compute the value function where $\nu(x, y) = 0$ when $\max\{|x|, |y|\} \leq \frac{1}{4}$ and $\nu(x, y) = 1$



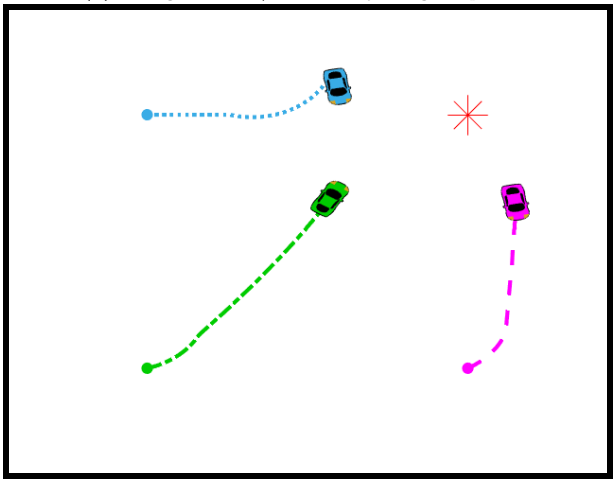
(a) Initial configurations.



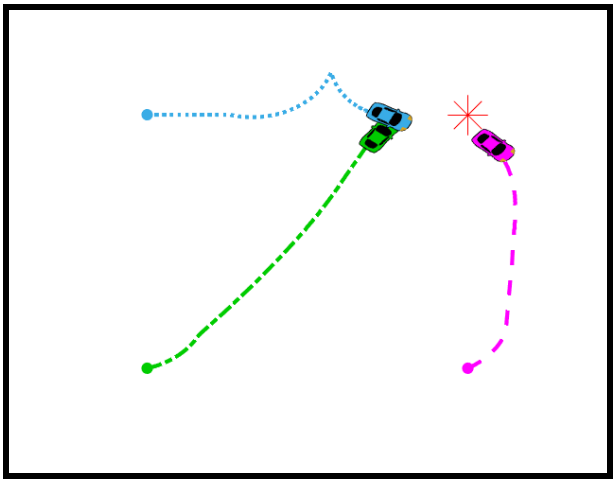
(b) Configurations 1/5 of the way along the paths.



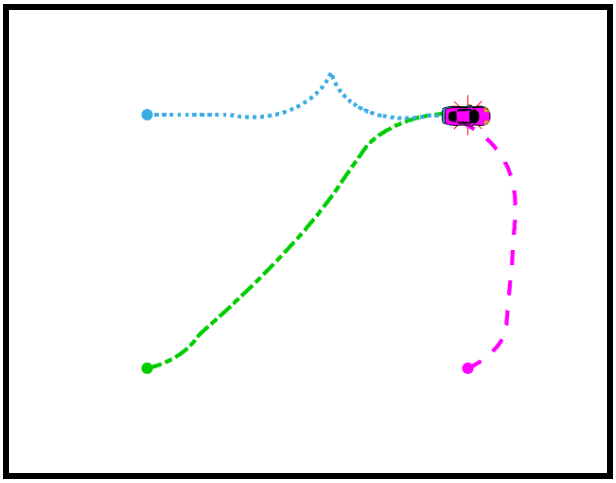
(c) Configurations 2/5 of the way along the paths.



(d) Configurations 3/5 of the way along the paths.



(e) Configurations 4/5 of the way along the paths.



(f) Final configurations.

Figure 6.4: Optimal paths for cars with $(x_0, y_0, \theta_0) = (-\frac{1}{2}, \frac{1}{2}, \pi)$ [blue], $(-\frac{1}{2}, -\frac{1}{2}, 0)$ [green], and $(\frac{1}{2}, -\frac{1}{2}, \frac{5\pi}{4})$ [pink]. The final configuration is $(x_f, y_f, \theta_f) = (\frac{1}{2}, \frac{1}{2}, 0)$ [red star].

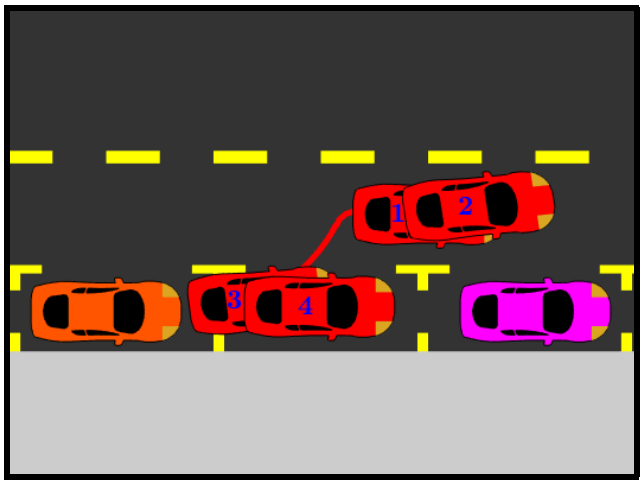


Figure 6.5: A car parallel parking demonstrates an optimal path with two kinks.

otherwise. This creates an obstacle represented by the black square in fig. 6.6. If the starting point (magenta dot) and ending point (red star) are centered horizontally on either side of the obstacle, the optimal path between them is non-unique. Indeed, for any point along the red line, there are multiple optimal trajectories, as depicted by the dotted and dashed lines.

A similar phenomenon arises in path planning for our car. The control variables are uniquely determined by (6.9) wherever ϕ is smooth, but looking at the isocontour in fig. 6.3a,

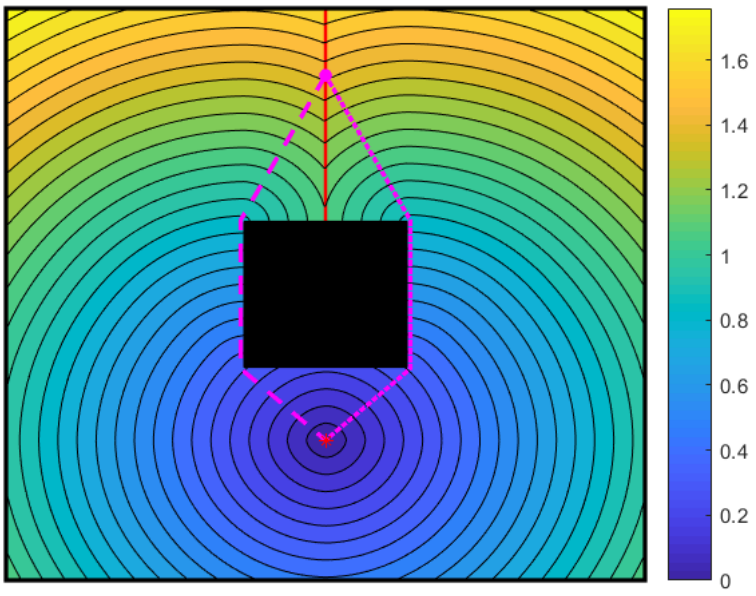
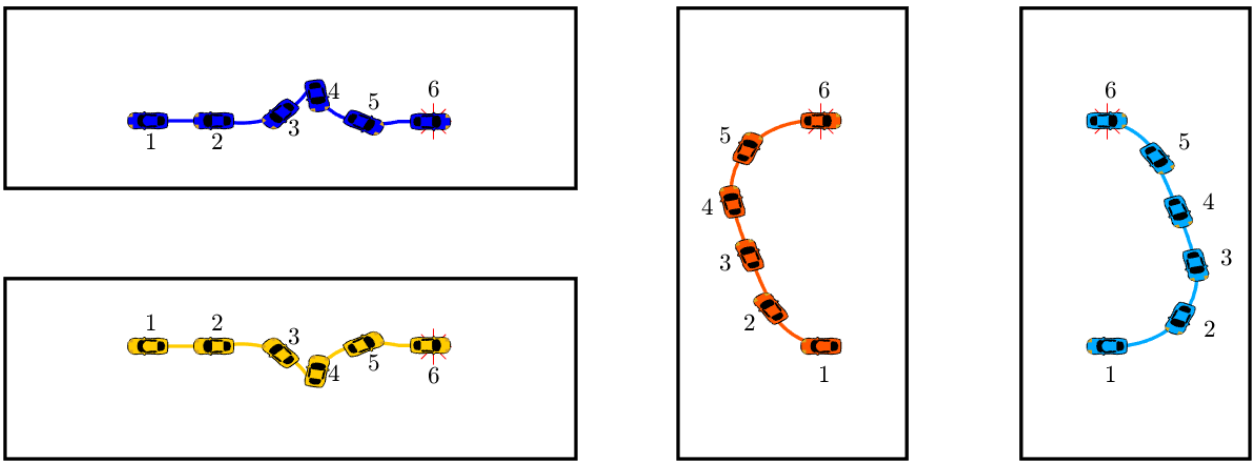


Figure 6.6: Contour map of the solution of ϕ to the Eikonal equation (6.23). Velocity is given by $\nu(x, y) = 0$ in the black square and $\nu(x, y) = 1$ elsewhere. Optimal trajectories are uniquely determined by the normal to level sets wherever the normal is well-defined. Along the red line, the normal cannot be determined and the optimal trajectories are non-unique; the dotted and dashed lines are equally optimal.



(a) Non-uniqueness of optimal trajectory from $(-\frac{1}{2}, \frac{1}{2}, \pi)$ to $(\frac{1}{2}, \frac{1}{2}, 0)$.

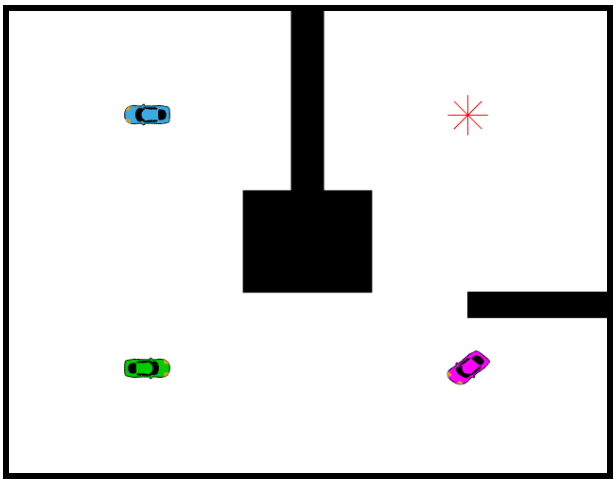
(b) Non-uniqueness of optimal trajectory from $(\frac{1}{2}, -\frac{1}{2}, \pi)$ to $(\frac{1}{2}, \frac{1}{2}, 0)$.

Figure 6.7: One can easily establish non-uniqueness of optimal paths when $\theta_0 = \theta_f + \pi$ by reflection or rotation.

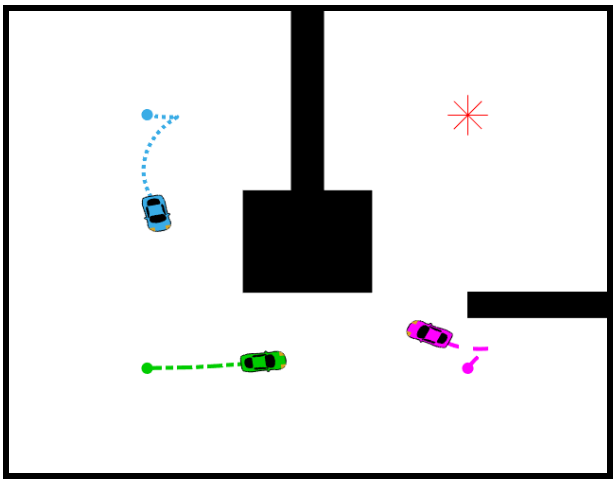
it is somewhat clear there are small sets where ϕ is not smooth and optimal trajectories cannot be determined uniquely. The most obvious points where ϕ will be non-differentiable are the points $(x, y, \theta_f + \pi)$. In this case, it is often easy to intuit multiple optimal paths. Included in fig. 6.7 are two examples of non-unique optimal paths. Figure 6.7a shows the same blue car as in fig. 6.4, and the path reflected across the line $y = \frac{1}{2}$ [gold]. Figure 6.7b shows an optimal path from $(x_0, y_0, \theta_0) = (\frac{1}{2}, -\frac{1}{2}, \pi)$ to $(x_f, y_f, \theta_f) = (\frac{1}{2}, \frac{1}{2}, 0)$ [orange] and the same path rotated by π about the point $(x, y) = (\frac{1}{2}, 0)$ [light blue]. Reflection and rotation are easy methods to establish non-uniqueness of the optimal trajectories, but there are other ways as well. Considering the paths in fig. 6.7a, we could translate the kink in the x-direction to arrive at more optimal trajectories.

Lastly, we introduce obstacles. To reiterate, the algorithm for solving (6.10) is the same, except that the value function is not updated at nodes corresponding to illegal configurations—those that would cause the car to collide with an obstacle. In fig. 6.8, we compute the optimal paths from the same three starting configurations as in fig. 6.4 but now with obstacles [black] hindering the cars' movement.

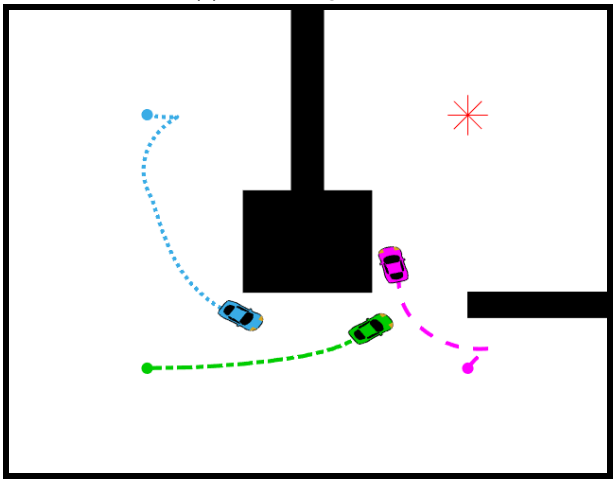
In fig. 6.9, we have a car pulling into a very narrow parking spot. Note that no extra consideration (in the form of local collision avoidance) was necessary to resolve this path. Here the width of the parking spot is only 0.1 and the width of the car is $2R = 0.08$.



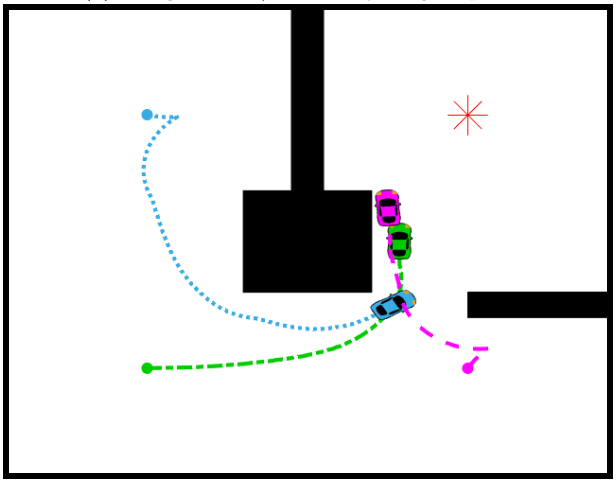
(a) Initial configurations.



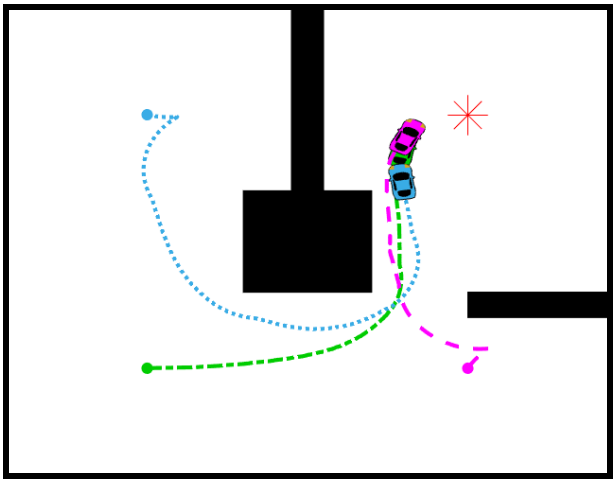
(b) Configurations 1/5 of the way along the paths.



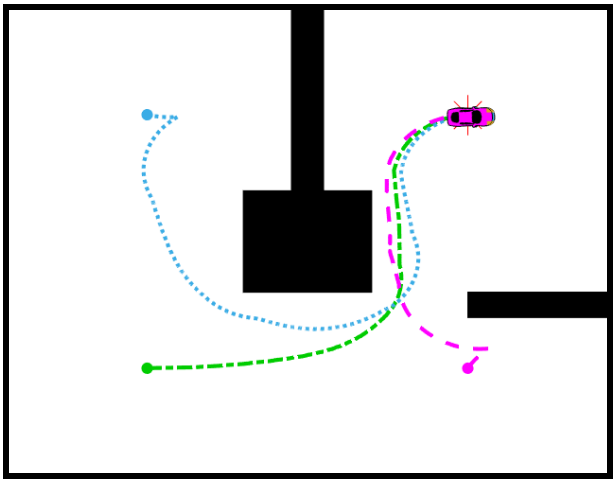
(c) Configurations 2/5 of the way along the paths.



(d) Configurations 3/5 of the way along the paths.



(e) Configurations 4/5 of the way along the paths.



(f) Final configurations.

Figure 6.8: Optimal paths for cars for the same cars as in fig. 6.4 with obstacles [black] occluding their way. The final configuration is $(x_f, y_f, \theta_f) = (\frac{1}{2}, \frac{1}{2}, 0)$ [red star].

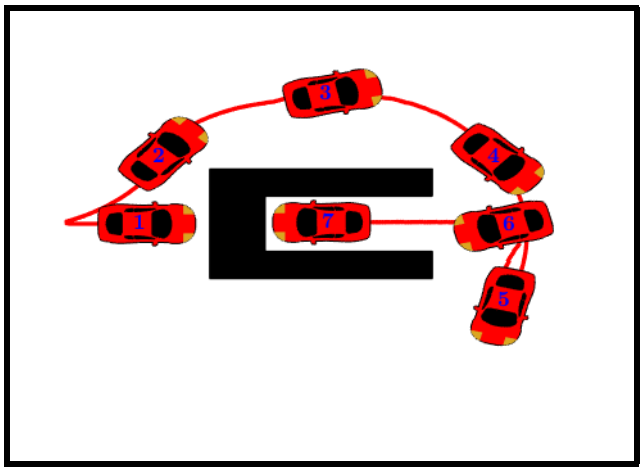


Figure 6.9: A car pulling into a very narrow parking spot.

Thus if we buffered the obstacles, the final configuration would likely by illegal, and if we approximated the car by a point mass, it is likely that it would have taken on some illegal configurations. As an aside, the optimal trajectory into the parking spot has three kinks, showing that the result of [RS90]—stating that only two kinks are sufficient—is not true in the presence of obstacles.

6.5 Conclusions & Future Work

In this chapter, we presented a Hamilton-Jacobi-Bellman formulation for optimal path planning of nonholonomic vehicles, accounting for impassable obstacles and the actual geometry of the vehicle. We developed an upwind sweeping scheme to solve the Hamilton-Jacobi-Bellman equation for the value function, and tested our scheme against the two dimensional Eikonal equation. We validated our model in the presence and absence of obstacles and compared with the classical results for curvature constrained motion. We note that no extra considerations were needed when dealing with obstacles since the geometry of the car is not being neglected.

We propose several avenues for future work on this project. The first would be to improve the realism by accounting for terrain data, modeling more specific problems, including vehicle dynamics, or considering any number of other physical concerns. Along with this, field tests

would be crucial. Terrain-based or indoor path-planning for nonholonomic vehicles like ours is the subject of much recent work [DMC12, DCK18, SLV17, WSY19, WDG19], but to our knowledge, none of this work includes the Hamilton-Jacobi formulation of the problem.

Similarly, one could account for concerns other than time-optimization in this Hamilton-Jacobi formulation. For example, Fierro and Lewis [FL98] compute the marginal energy expenditure along the path

$$E(v, \omega) = \frac{1}{2}mv^2 + \frac{1}{2}I\omega^2 \tag{6.24}$$

where m is the mass of the vehicle and I is the moment of inertia of the vehicle (for the rectangular vehicle considered above, we would have $I = \frac{1}{3}m(R^2 + 4d^2)$). Using this, we could change the cost functional to something like

$$C[\boldsymbol{v}(\cdot), \boldsymbol{\omega}(\cdot)] = \int_0^T [1 + \lambda E(\boldsymbol{v}(t), \boldsymbol{\omega}(t))]dt, \tag{6.25}$$

where $\lambda \geq 0$ is some weight that determines the importance of conserving energy. Time minimization and energy minimization are competing goals. If $\lambda = 0$, we revert to time-optimal path planning. As [DS03] points out, the $\lambda \to +\infty$ limit is not meaningful, since the "optimal" control plan is to stay motionless in that case; thus if energy is the only concern, one must enforce a finite time horizon. This energy term is seemlessly handled by the Hamilton-Jacobi formulation. The new Hamilton-Jacobi-Bellman equation is simply

$$-1 = \inf_{v,\omega} \Big\{ (\phi_x \cos\theta + \phi_y \sin\theta)v + W(-d\phi_x \sin\theta + d\phi_y \cos\theta + \phi_\theta)\omega + \lambda E(v, \omega) \Big\}. \tag{6.26}$$

However, this significantly complicates the numerics. Because of the energy term, there will no longer be a simple, upwind update formula for a sweeping scheme. Also, the equation is no longer a level set equation so single-pass fast marching methods [Tsi95, BS98, AM06, AM09] and level set methods like that presented in chapter 4 are no longer available. Instead, one can recast the problem in a time-dependent manner. Indeed, defining the convex indicator

function of the final configuration

$$\iota_f(x, y, \theta) = \begin{cases} 0, & (x, y, \theta) = (x_f, y_f, \theta_f), \\ +\infty, & \text{otherwise}, \end{cases} \tag{6.27}$$

the equivalent time dependent formulation is

$$\begin{aligned} &\phi_t + \inf_{v,\omega} \left\{ (\phi_x \cos\theta + \phi_y \sin\theta)v + W(-d\phi_x \sin\theta + d\phi_y \cos\theta + \phi_\theta)\omega + \lambda E(v, \omega) \right\} = -1, \\ &\phi(x, y, \theta, T) = \iota_f(x, y, \theta), \\ &\phi(x_f, y_f, \theta_f, t) = 0, \quad \text{for all } t \in [0, T]. \end{aligned} \tag{6.28}$$

Equation (6.28) can be solved numerically using the generic methods for Hamilton-Jacobi equation presented in section 2.4. However, even this is nontrivial. The equation is $3 + 1$ dimensional, and the update rule for the Godunov scheme will be very difficult to resolve. The time horizon T can be taken to be arbitrarily large; given infinite time, the optimal cost function should be static, so if one makes the substitution $\tau = T - t$ and lets $T \to \infty$, then the equation runs forward in time and the solution will reach steady state in finite time. This is a commonality shared with single-pass algorithms. Thus while single-pass algorithms do not apply in their basic form, it is likely possible to adapt them to this situation.

A third avenue for future work is extension of the model to higher dimensions. For example, there are two natural extensions to three dimensions, analogous to the representation of $\mathbb{R}^3$ using either cylindrical or spherical coordinates. In what follows we assume the vehicle is simplified to a point mass. If the curvature constraint in the xy-plane is decoupled from the curvature constraint in the z-direction, the travel time function satisfies the equation

$$1 = |\phi_x \cos\theta + \phi_y \sin\theta| + W_{xy} |\phi_\theta| + W_z |\phi_z| , \tag{6.29}$$

where $W_{xy}, W_z > 0$ are bounds on the angular velocity of planar or vertical motion, respectively. This could model the kinematics of a simple airplane. Alternatively, one could enforce

a total curvature constraint, in which case the travel time function satisfies

$$1 = |\phi_x \cos\theta \cos\varphi + \phi_y \sin\theta \cos\varphi + \phi_z \sin\varphi| + W\sqrt{\phi_\varphi^2 + \frac{\phi_\theta^2}{\cos^2\varphi}}, \tag{6.30}$$

where, as in spherical coordinates, θ represents the xy-planar angle and φ is the angle of inclination [BM10, CZZ17, CS20]. This could perhaps model the kinematics of a simple submarine. The application of these Hamilton-Jacobi formulations to actual airplanes or submarines would likely require additional modeling concerns (for example, an airplane may need to maintain a minimum cruising speed, and a submarine will not have a braking mechanism similar to that of a car), and perhaps even dynamic considerations. However, the simple kinematics have other application. As [RS90, TT13] point out, this could be used by a plumber to plan optimal piping trajectories assuming the pipes cannot be bent too much. There is also application in the medical field: Duan et al. [DPS14] model manueverable bevel-tip needles using curvature constrained paths in 3D, and Duits et al. [DMM18] apply similar methods to model blood vessels in retinal images and brain-connectivity measures in MRI.

Finally, one could modify this model for differential games. Isaacs devised the now famous homicidal chauffeur problem [Isa51, Isa65] wherein a pursuer, who is constrained to paths of bounded curvature, tries to collide with an evader who is slower, but exhibits isotropic motion. While this is a slightly frivolous example, there could be very interesting differential games involving vehicles of this type. For example, classical mathematical models of traffic flow concentrated largely on macroscopic quantities such as density and flow rate, as in the canonical Lighthill-Whitham-Richards model [LW55a, LW55b, Ric56]. In recent years, there has been a push to model traffic flow microscopically, considering each vehicle individually [LCP17, LKY19, WSC19]. One could modify our Hamilton-Jacobi formulation by including interaction between vehicles in order to model traffic jams via mean field games. This could be a significant step bridging the gap between classical macroscopic traffic models and modern microscopic traffic models.

CHAPTER 7

Conclusions

In this dissertation we explored models for human navigation and optimal path planning. Key ingredients of the models included level set methods, optimal control theory and Hamilton-Jacobi equations. We covered the basic theory surrounding these mathematical tools in chapter 2. These methods are very broadly applicable, and could be easily adapted to any number of physical scenarios.

In chapter 3, we introduced a novel, continuous model for deforestation in protected areas. Previous models of this type relied on overly stringent assumptions of symmetry. Using the level set method, we were able to remove any such assumptions, thus providing a model which is applicable to protected areas with realistic geometries. As a proof of concept, we applied our model to Yosemite National Park and Kangaroo Island and pointed out that simple geometric considerations can greatly improve patrol efficacy. We suggested several avenues for future work on this project, including adapting the model for repeated differential games, addressing resourse allocation for real patrol routes, and modifying the model to include real data.

In chapter 4, we designed a level set model for terrain-based optimal path planning. By representing walking direction with a control variable, we derived a Hamilton-Jacobi-Bellman equation which corresponds to optimal travel outward from a starting point. We tested this method in the surroundings of El Capitan and Half Dome, two mountains in Yosemite National Park. We then suggested a method for incorporating uncertainty in the location of the starting point, and discussed clustering all potential paths into a small number of routes.

In chapter 5, we revamped and extended the optimal path planning model from chapter 4

to include stochastic effects. Uncertainty in the equation of motion led to diffusion in the Hamilton-Jacobi-Bellman equation meaning that the level set interpretation of the problem was no longer available. Thus we opted for a control theoretic model. We offered two different notions of "optimal path" when there is uncertainty in the problem, and tested our model against synthetic data and real elevation data, again using El Capitan as an example. We discussed the role of the time horizon in the problem, since this is the major qualitative difference between this model and the level set model. We outlined a few ways this work could be extended. Perhaps the most significant extension would be to adjust the model to be able to answer probabilistic path planning questions. For example, under some uncertainty, could one compute the set which is reachable with %50 likelihood?

In chapter 6, we implemented a model for time-optimal path planning of simple nonholonomic vehicles. Since cars cannot move omnidirectionally, the control variable could not be the direction of motion as in the previous chapters. Rather, we modeled the motion of a car using location and orientation, and controlled these via tangential and angular velocity. The Hamilton-Jacobi formulation averted the need for hierarchical collision avoidance algorithms, and we did not neglect the geometry of the car, meaning no additional consideration was needed near boundaries of obstacles. We suggested additional work on this model to include energy minimization, adapt the model for higher dimensions, or simulate real situations such as traffic flow.

Optimal control theory and Hamilton-Jacobi equations provide a simple framework for countless questions surrounding optimal path planning and human navigation. This dissertation presented several examples of their utility.

BIBLIOGRAPHY

[ABS18] J. Alonso-Mora, P. Beardsley, and R. Siegwart. "Cooperative Collision Avoidance for Nonholonomic Robots." *IEEE Transactions on Robotics*, **34**(2):404–420, April 2018.

[AFJ19] David J Arnold, Dayne Fernandez, Ruizhe Jia, Christian Parkinson, Deborah Tonne, Yotam Yaniv, Andrea L Bertozzi, and Stanley J Osher. "Modeling Environmental Crime in Protected Areas Using the Level Set Method." *SIAM Journal on Applied Mathematics*, **79**(3):802–821, 2019.

[AL12] Cesar O Aguilar and Andrew D Lewis. "Small-time local controllability for a class of homogeneous systems." *SIAM Journal on Control and Optimization*, **50**(3):1502–1517, 2012.

[Alb10] H. J. Albers. "Spatial modeling of extraction and enforcement in developing country protected areas." *Resource and Energy Economics*, **32**:165–179, 2010.

[ALM10] Roumen Anguelov, Jean M.-S. Lubuma, and Froduald Minani. "A monotone scheme for Hamilton–Jacobi equations via the nonstandard finite difference method." *Mathematical Methods in the Applied Sciences*, **33**(1):41–48, 2010.

[AM01] Andreas Aurnhammer and Kurt Marti. "Adaptive Optimal Stochastic Trajectory Planning." In Martin Grötschel, Sven O. Krumke, and Jörg Rambau, editors, *Online Optimization of Large Scale Systems*, pp. 521–543. 2001.

[AM06] K. Alton and I. M. Mitchell. "Optimal path planning under defferent norms in continuous state spaces." In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 866–872, May 2006.

[AM09] Ken Alton and Ian M Mitchell. "Fast marching methods for stationary Hamilton–Jacobi equations with axis-aligned anisotropy." *SIAM Journal on Numerical Analysis*, **47**(1):363–385, 2009.

[AP01] António Pedro Aguiar and Antonio M Pascoal. "Regulation of a nonholonomic autonomous underwater vehicle with parametric modeling uncertainty using Lyapunov functions." In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 5, pp. 4178–4183. IEEE, 2001.

[AS95] David Adalsteinsson and James A Sethian. "A Fast Level Set Method for Propagating Interfaces." *Journal of Computational Physics*, **118**(2):269–277, 1995.

[AS06] M. Aaibid and A. Sayah. "A direct proof of the equivalence between the entropy solutions of conservation laws and viscosity solutions of Hamilton-Jacobi equations in one-space variable." *JIPAM. Journal of Inequalities in Pure & Applied Mathematics [electronic only]*, **7**(2), 2006.

[AW01] Pankaj K. Agarwal and Hongyan Wang. "Approximation Algorithms for Curvature-Constrained Shortest Paths." *SIAM Journal on Computing*, **30**(6):1739–1772, 2001.

[AWS16] Zain Anwar Ali, Daobo Wang, Muhammad Safwan, Wanyue Jiang, and Muhammad Shafiq. "Trajectory tracking of a nonholonomic wheeled mobile robot using hybrid controller." *International Journal of Modeling and Optimization*, **6**(3):136, 2016.

[Bar13] Guy Barles. *An Introduction to the Theory of Viscosity Solutions for First-Order Hamilton–Jacobi Equations and Applications*, pp. 49–109. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[BC08] M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Modern Birkhäuser Classics. Birkhäuser Boston, 2008.

[Bel54] Richard Bellman. "The theory of dynamic programming." Technical report, Rand Corp, Santa Monica, CA, 1954.

[Bel61] Richard Bellman. *Adaptive Control Processes: A Guided Tour*. Karreman Mathematics Research Collection, Princeton Legacy Library. Princeton University Press, 1961.

[Ber00] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2 edition, 2000.

[BFY13] Alain Bensoussan, Jens Frehse, Phillip Yam, et al. *Mean field games and mean field type control theory*, volume 101. Springer, 2013.

[BJ05] Guy. Barles and Espen R. Jakobsen. "Error Bounds for Monotone Approximation Schemes for Hamilton–Jacobi–Bellman Equations." *SIAM Journal on Numerical Analysis*, **43**(2):540–558, 2005.

[BL93] Jérôme Barraquand and Jean-Claude Latombe. "Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles." *Algorithmica*, **10**(2-4):121, 1993.

[BM10] AR Babaei and M Mortazavi. "Three-dimensional curvature-constrained trajectory planning based on in-flight waypoints." *Journal of Aircraft*, **47**(4):1391–1398, 2010.

[BMO07] A. Bayen, I. M. Mitchell, M. Oishi, and C. J. Tomlin. "Aircraft Autolander Safety Analysis Through Optimal Control-Based Reach Set Computation." *Journal of Guidance Control and Dynamics*, **30**:68–77, 01 2007.

[BO91] Martino Bardi and Stanley Osher. "The nonconvex multidimensional Riemann problem for Hamilton–Jacobi equations." *SIAM Journal on Mathematical Analysis*, **22**(2):344–351, 1991.

[Bry96] A. E. Bryson. "Optimal control-1950 to 1985." *IEEE Control Systems Magazine*, **16**(3):26–33, June 1996.

[Bry18] Arthur Earl Bryson. *Applied optimal control: optimization, estimation and control*. Routledge, 2018.

[BS98] Timothy J Barth and James A Sethian. "Numerical schemes for the Hamilton–Jacobi and level set equations on triangulated domains." *Journal of Computational Physics*, **145**(1):1–40, 1998.

[BVM10] Vidhya Balasubramanian, Dmitri V. Kalashnikov, S Mehrotra, and Nalini Venkatasubramanian. "Efficient and Scalable Multi-Geography Route Planning." In *Proceedings of 13th International Conference on Extending Database Technology*, pp. 394–405, 2010.

[CD18] René Carmona and François Delarue. *Probabilistic Theory of Mean Field Games with Applications I.* Springer International Publishing, 1 edition, 2018.

[CDO19] Yat Tin Chow, Jérôme Darbon, Stanley Osher, and Wotao Yin. "Algorithm for overcoming the curse of dimensionality for state-dependent Hamilton-Jacobi equations." *Journal of Computational Physics*, **387**:376–409, 2019.

[CEL84] Michael G Crandall, Lawrence C Evans, and P-L Lions. "Some properties of viscosity solutions of Hamilton-Jacobi equations." *Transactions of the American Mathematical Society*, **282**(2):487–502, 1984.

[CGB15] Leonardo Colombo, Rohit Gupta, Anthony Bloch, and David Martín de Diego. "Variational discretization for optimal control problems of nonholonomic mechanical systems." In *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 4047–4052. IEEE, 2015.

[CGG99] Yun-Gang Chen, Yoshikazu Giga, and Shun'Ichi Goto. *Uniqueness and Existence of Viscosity Solutions of Generalized mean Curvature Flow Equations*, pp. 375–412. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[CL83] Michael G. Crandall and Pierre-Louis Lions. "Viscosity Solutions of Hamilton-Jacobi Equations." *Transactions of the American Mathematical Society*, **277**(1):1–42, 1983.

[CL84] Michael G Crandall and P-L Lions. "Two approximations of solutions of Hamilton-Jacobi equations." *Mathematics of computation*, **43**(167):1–19, 1984.

[CLS19] Elliot Cartee, Lexiao Lei, Qianli Song, and Alexander Vladimirsky. "Time-Dependent Surveillance-Evasion Games." In *2019 IEEE Conference on Decision and Control (CDC)*, pp. 7128–7133. IEEE, 2019.

[CM80] Michael G. Crandall and Andrew J. Majda. "Monotone Difference Approximations for Scalar Conservation Laws." 1980.

[CMK01] Susan Chen, Barry Merriman, Myungjoo Kang, Russel E. Caflisch, Christian Ratsch, Li-Tien Cheng, Mark Gyure, Ronald P. Fedkiw, Christopher Anderson, and Stanley Osher. "A Level Set Method for Thin Film Epitaxial Growth." *Journal of Computational Physics*, **167**(2):475 – 500, 2001.

[Con06] Peter Constantin. "Euler equations, Navier-Stokes equations and turbulence." In *Mathematical foundation of turbulent viscous flows*, pp. 1–43. Springer, 2006.

[CP11] Antonin Chambolle and Thomas Pock. "A first-order primal-dual algorithm for convex problems with applications to imaging." *Journal of mathematical imaging and vision*, **40**(1):120–145, 2011.

[CS03] Gui-Qiang Chen and Bo Su. "Discontinuous Solutions of Hamilton-Jacobi Equations: Existence, Uniqueness, and Regularity." In *Hyperbolic Problems: Theory, Numerics, Applications*, pp. 443–453. Springer, 2003.

[CS20] Reinis Cimurs and Il Hong Suh. "Time-optimized 3D Path Smoothing with Kinematic Constraints." *International Journal of Control, Automation and Systems*, pp. 1–11, 2020.

[CV19] Elliot Cartee and Alexander Vladimirsky. "Control-Theoretic Models of Environmental Crime.", 2019. https://arxiv.org/abs/1906.09289.

[CZZ17] Wenyu Cai, Meiyan Zhang, and Yahong Rosa Zheng. "Task assignment and path planning for multiple autonomous underwater vehicles using 3D dubins curves." *Sensors*, **17**(7):1607, 2017.

[DCK18] Yiqun Dong, Efe Camci, and Erdal Kayacan. "Faster rrt-based nonholonomic path planning in 2d building environments using skeleton-constrained path biasing." *Journal of Intelligent & Robotic Systems*, **89**(3-4):387–401, 2018.

[Dij59] E. W. Dijkstra. "A Note on Two Problems in Connexion with Graphs." *Numer. Math.*, **1**(1):269–271, December 1959.

[DMC12] P. L. J. Drews Jr., D. G. Macharet, and M. F. M. Campos. "A Terrain-Based Path Planning for Mobile Robots with Bounded Curvature." In *2012 Brazilian Robotics Symposium and Latin American Robotics Symposium*, pp. 202–207, 2012.

[DMM18] Remco Duits, Stephan PL Meesters, J-M Mirebeau, and Jorg M Portegies. "Optimal paths for variants of the 2D and 3D Reeds–Shepp car with applications in image analysis." *Journal of mathematical imaging and vision*, **60**(6):816–848, 2018.

[DO16] Jérôme Darbon and Stanley Osher. "Algorithms for overcoming the curse of dimensionality for certain Hamilton–Jacobi equations arising in control theory and elsewhere." *Research in the Mathematical Sciences*, **3**(1):19, 2016.

[DPS14] Yan Duan, Sachin Patil, John Schulman, Ken Goldberg, and Pieter Abbeel. "Planning locally optimal, curvature-constrained trajectories in 3D using sequential convex optimization." In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5889–5895. IEEE, 2014.

[DS03] Ignacy Duleba and Jurek Z. Sasiadek. "Nonholonomic motion planning based on Newton algorithm with energy optimization." *IEEE transactions on control systems technology*, **11**(3):355–363, 2003.

[Dub57] L. E. Dubins. "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents." *American Journal of Mathematics*, **79**(3):497–516, 1957.

[Eva80] Lawrence C. Evans. "On solving certain nonlinear partial differential equations by accretive operator methods." *Israel Journal of Mathematics*, **36**(3-4):225–247, 1980.

[Eva10] L.C. Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 2010.

[EZL12] V. Estellers, D. Zosso, R. Lai, S. Osher, J. P. Thiran, and X. Bresson. "An Efficient Algorithm for Level Set Method Preserving Distance Function." *IEEE Transactions on Image Processing*, **21**(12):4722–4734, Dec 2012.

[FF19] Roberto Ferretti and Adriano Festa. "Optimal route planning for sailing boats: a hybrid formulation." *Journal of Optimization Theory and Applications*, **181**(3):1015–1032, 2019.

[FJT13] F. Fang, A. X. Jiang, and M. Tambe. "Optimal patrol strategy for protecting moving targets with multiple mobile resources." In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, pp. 957–964. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[FL98] Rafael Fierro and Frank L Lewis. "Control of a nonholonomic mobile robot using neural networks." *IEEE transactions on neural networks*, **9**(4):589–600, 1998.

[FNP17] F. Fang, T. H. Nguyen, R. Pickles, W. Y. Lam, G. R. Clements, B. An, A. Singh, B. C. Schwedock, M. Tambe, and A. Lemieux. "PAWS – a Deployed Game-Theoretic Application to Combat Poaching." *AI Magazine*, **38**(1):23–36, March 2017.

[FR75] Wendell H. Fleming and Raymond W. Rishel. *Deterministic and Stochastic Optimal Control*. Springer-Verlag New York, 1 edition, 1975.

[GFC03] Frédéric Gibou, Ronald Fedkiw, Russel Caflisch, and Stanley Osher. "A level set approach for the numerical simulation of dendritic growth." *Journal of Scientific Computing*, **19**(1-3):183–199, 2003.

[GFO18] Frederic Gibou, Ronald Fedkiw, and Stanley Osher. "A review of level-set methods and some recent applications." *Journal of Computational Physics*, **353**:82 – 109, 2018.

[GLL11] Olivier Guéant, Jean-Michel Lasry, and Pierre-Louis Lions. *Mean Field Games and Applications*, pp. 205–266. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[GO09] Tom Goldstein and Stanley Osher. "The split Bregman method for L1-regularized problems." *SIAM journal on imaging sciences*, **2**(2):323–343, 2009.

[God59] Sergei Konstantinovich Godunov. "A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics." *Matematicheskii Sbornik*, **89**(3):271–306, 1959.

[GT01] D. Gilbarg and N.S. Trudinger. *Elliptic Partial Differential Equations of Second Order*. Classics in Mathematics. Springer Berlin Heidelberg, 2001.

[GV] Marc A. Gilles and Alexander Vladimirsky. "Surveillance-Evasion games under uncertainty." *arXiv preprint*. https://arxiv.org/abs/1812.10620.

[GZ18] Victoria Grushkovskaya and Alexander Zuyev. "Obstacle avoidance problem for second degree nonholonomic systems." In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 1500–1505. IEEE, 2018.

[Hac08] O Hachour. "Path Planning of Autonomous Mobile Robots." *International Journal of Systems Applications, Engineering & Development*, **2**(4):178–190, 2008.

[Ham33] Sir William Rowan Hamilton. "On a General Method of Expressing the Paths of Light, & of the Planets, by the Coefficients of a Characteristic Function." *Dublin University Review and Quarterly Magazine*, **1**:795–836, 1833.

[Ham34] William Rowan Hamilton. "On the application to dynamics of a general mathematical method previously applied to optics." *British Association Report*, pp. 513–518, 1834.

[HEO87] Ami Harten, Bjorn Engquist, Stanley Osher, and Sukumar R Chakravarthy. "Uniformly high order accurate essentially non-oscillatory schemes, III." In *Upwind and high-resolution schemes*, pp. 218–290. Springer, 1987.

[HJW18] Jiequn Han, Arnulf Jentzen, and E Weinan. "Solving high-dimensional partial differential equations using deep learning." *Proceedings of the National Academy of Sciences*, **115**(34):8505–8510, 2018.

[HNG15] Christian Hirsch, David Neuhäuser, Catherine Gloaguen, and Volker Schmidt. "Asymptotic properties of Euclidean shortest-path trees in random geometric graphs." *Statistics & Probability Letters*, **107**:122 – 130, 2015.

[IC17] I. J. Irmischer and K. C. Clarke. "Measuring and modeling the speed of human navigation." *Cartograpghy and Geographic Information Science*, **45**(2):177–186, 2017.

[INY11] G. Ishigami, K. Nagatani, and K. Yoshida. "Path Planning and Evaluation for Planetary Rovers Based on Dynamic Mobility Index." In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 601–606, Sept 2011.

[Isa51] Rufus Isaacs. "Games of pursuit." 1951.

[Isa65] Rufus Isaacs. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization.* John Wiley & Sons, New York, NY, 1965.

[Jaf20] Saber Jafarpour. "On Small-Time Local Controllability." *SIAM Journal on Control and Optimization*, **58**(1):425–446, 2020.

[JFT12] M. P. Johnson, F. Fang, and M. Tambe. "Patrol Strategies to Maximize Pristine Forest Area." In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI, 2012.

[JT96] Damir Juric and Grétar Tryggvason. "A Front-Tracking Method for Dendritic Solidification." *Journal of Computational Physics*, **123**(1):127 – 148, 1996.

[JZN12] Xin Jiang, Renjie Zhang, and Shengdong Nie. "Image Segmentation Based on Level Set Method." *Physics Procedia*, **33**:840 – 845, 2012. 2012 International Conference on Medical Physics and Biomedical Engineering (ICMPBE2012).

[Kaw90] Matthias Kawski. *High-order small-time local controllability*, pp. 431–467. CRC Press, 1990.

[KFD15] D. Kar, F. Fang, F. Delle Fave, N. Sintov, and M. Tambe. "A game of thrones: when human behavior models compete in repeated Stackelberg security games." In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1381–1390. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[KFG17] D. Kar, B. Ford, S. Gholami, F. Fang, A. Plumbtree, M. Tambe, M. Driciru, F. Wanyama, and A. Rwetsiba. "Cloudy with a Chance of Poaching: adversary Behavior Modeling and Forecasting with Real-World Poaching Data." In S. Das, E. Durfee, K. Larson, and M. Winikoff, editors, *Proceedings of 16th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '17, pp. 159–167. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[KGF18] Nitin Kamra, Umang Gupta, Fei Fang, Yan Liu, and Milind Tambe. "Policy Learning for Continuous Space Security Games Using Neural Networks." In *AAAI Conference on Artificial Intelligence*, 2018.

[KNH19] Amna Khan, Iram Noreen, and Zulfiqar Habib. "An Energy Efficient Coverage Path Planning Approach for Mobile Robots." In Kohei Arai, Supriya Kapoor, and Rahul Bhatia, editors, *Intelligent Computing*, pp. 387–397, Cham, 2019. Springer International Publishing.

[Kno81] G. Knowles. *An Introduction to Applied Optimal Control*. Mathematics in science and engineering. Academic Press, 1981.

[KOQ04] Chiu Yen Kao, Stanley Osher, and Jianliang Qian. "Lax–Friedrichs sweeping scheme for static Hamilton–Jacobi equations." *Journal of Computational Physics*, **196**(1):367–391, 2004.

[KOT05] Chiu-Yen. Kao, Stanley. Osher, and Yen-Hsi. Tsai. "Fast Sweeping Methods for Static Hamilton–Jacobi Equations." *SIAM Journal on Numerical Analysis*, **42**(6):2612–2632, 2005.

[KP92] Peter E. Kloeden and Eckhard Platen. "Strong Taylor Approximations." In *Numerical Solution of Stochastic Differential Equations*, pp. 339–371. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.

[KQ01] Mikhail Krastanov and Marc Quincampoix. "Local small time controllability and attainability of a set for nonlinear control system." *ESAIM: Control, Optimisation and Calculus of Variations*, **6**:499–516, 2001.

[Kra98] MI Krastanov. "A necessary condition for small time local controllability." *Journal of dynamical and control systems*, **4**(3):425–456, 1998.

[KS95] Gita Krishnaswamy and Anthony Stentz. "Resolution Independent Grid-Based Path Planning." Technical report, Carnegie-Mellon University Robotics Institute, 1995.

[KS98] Ron Kimmel and James A Sethian. "Computing geodesic paths on manifolds." *Proceedings of the National Academy of Sciences*, **95**(15):8431–8435, 1998.

[KT08] Chiu-Yen Kao and Richard Tsai. "Properties of a Level Set Algorithm for the Visibility Problems." *Journal of Scientific Computing*, **35**:170–191, 2008.

[LCO18] Alex Tong Lin, Yat Tin Chow, and Stanley J. Osher. "A splitting method for overcoming the curse of dimensionality in Hamilton–Jacobi equations arising from nonlinear optimal control and differential games with applications to trajectory generation." *Communications in Mathematical Sciences*, **16**(7), 1 2018.

[LCP17] Yongfu Li, Wenbo Chen, Srinivas Peeta, Xiaozheng He, Taixiong Zheng, and Huizong Feng. "An extended microscopic traffic flow model based on the spring-mass system theory." *Modern Physics Letters B*, **31**(09):1750090, 2017.

[LDM14] Curtis Lee, John Dolbow, and Peter J. Mucha. "A narrow-band gradient-augmented level set method for multiphase incompressible flow." *Journal of Computational Physics*, **273**:12 – 37, 2014.

[LDO17] B. Lee, J. Darbon, S. Osher, and M. Kang. "Revisiting the Redistancing Problem Using the Hopf–Lax Formula." *Journal of Computational Physics*, **330**(1):268–281, February 2017.

[LG09] Lanny Lin and Michael A Goodrich. "UAV intelligent path planning for wilderness search and rescue." In *Intelligent robots and systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 709–714. IEEE, 2009.

[LHD11] Chunming Li, Rui Huang, Zhaohua Ding, J Chris Gatenby, Dimitris N Metaxas, and John C Gore. "A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI." *IEEE Transactions on Image Processing*, **20**(7):2007–2016, 2011.

[LJL15] T. Lolla, P.J. Haley Jr., and P.F.J. Lermusiaux. "Path planning in multi-scale ocean flows: Coordination and dynamic obstacles." *Ocean Modelling*, **94**:46 – 66, 2015.

[LJO17] Beom H. Lee, Jae D. Jeon, and Jung H. Oh. "Velocity obstacle based local collision avoidance for a holonomic elliptic robot." *Autonomous Robots*, **41**(6):1347–1363, Aug 2017.

[LKY19] Zhaojian Li, Firas Khasawneh, Xiang Yin, Aoxue Li, and Ziyou Song. "A new microscopic traffic model using a spring-mass-damper-clutch system." *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[LLM14] R. Luna, M. Lahijanian, M. Moll, and L. E. Kavraki. "Fast stochastic motion planning with optimality guarantees using local policy reconfiguration." In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3013–3019, 2014.

[LM93] N. Leader-Williams and E. J. Milner-Gulland. "Policies for the enforcement of wildlife laws: the balance between detection and penalties in Luangwa Valley, Zambia." *Conservation Biology*, **7**(3):611–617, 1993.

[LO95] A. De Luca and G. Oriolo. *Modelling and Control of Nonholonomic Mechanical Systems*, pp. 277–342. Springer Vienna, Vienna, 1995.

[LOC94] Xu-Dong Liu, Stanley Osher, and Tony Chan. "Weighted essentially non-oscillatory schemes." *Journal of Computational Physics*, **115**(1):200–212, 1994.

[LR12] S. Lim and D. Rus. "Stochastic motion planning with path constraints and application to optimal agent, resource, and route planning." In *2012 IEEE International Conference on Robotics and Automation*, pp. 4814–4821, 2012.

[LSC19] Kun Luo, Changxiao Shao, Min Chai, and Jianren Fan. "Level set method for atomization and evaporation simulations." *Progress in Energy and Combustion Science*, **73**:65 – 94, 2019.

[LUY12] Tapovan Lolla, Mattheus P Ueckermann, K Yiğit, Patrick J Haley, and Pierre FJ Lermusiaux. "Path planning in time dependent flow fields using level set meth-

ods." In *2012 IEEE International Conference on Robotics and Automation*, pp. 166–173. IEEE, 2012.

[LW55a] Michael James Lighthill and G Be Whitham. "On kinematic waves I. Flood movement in long rivers." *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, **229**(1178):281–316, 1955.

[LW55b] Michael James Lighthill and Gerald Beresford Whitham. "On kinematic waves II. A theory of traffic flow on long crowded roads." *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, **229**(1178):317–345, 1955.

[Lyg04] John Lygeros. "On reachability and minimum cost optimal control." *Automatica*, **40**(6):917 – 927, 2004.

[Mar99] K. Marti. "Path planning for robots under stochastic uncertainty." *Optimization*, **45**(1-4):163–195, 1999.

[Mar15] Kurt Marti. "Adaptive Optimal Stochastic Trajectory Planning and Control." In *Stochastic Optimization Methods: Applications in Engineering and Operations Research*, pp. 119–194. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[Min04] Chohong Min. "Local level set method in high dimension and codimension." *Journal of Computational Physics*, **200**(1):368 – 382, 2004.

[MMM16] K. R. Memon, S. Memon, B. Memon, A. R. Memon, and S. M. Z. A. Shah. "Real time implementation of path planning algorithm with obstacle avoidance for autonomous vehicle." In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 2048–2053, March 2016.

[MS04] Joseph SB Mitchell and Micha Sharir. "New results on shortest paths in three dimensions." In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pp. 124–133. ACM, 2004.

[MT] L. Martin and R. Tsai. "Equivalent extensions of Hamilton-Jacobi-Bellman equations on hypersurfaces (preprint).".

[MTK16] Sara McCarthy, Milind Tambe, Christopher Kiekintveld, Meredith L. Gore, and Alex Killion. "Preventing Illegal Logging: Preventing Illegal Logging: Simultaneous Optimization of Resource Teams and Tactics for Security." In *AAAI Conference on Artificial Intelligence*, 2016.

[NGK19] Tenavi Nakamura-Zimmerer, Qi Gong, and Wei Kang. "Adaptive Deep Learning for High Dimensional Hamilton-Jacobi-Bellman Equations." *arXiv preprint*, 2019. https://arxiv.org/abs/1907.05317.

[Nis09] Makiko Nisio. *Stochastic Control Theory*. Springer Japan, 2 edition, 2009.

[NME19] D. Nieto-Hernández, C. Méndez-Barrios, J. Escareno, V. Ramírez-Rivera, L. A. Torres, and H. Méndez-Azúa. "Non-holonomic flight Modeling and Control of a Tilt-rotor MAV." In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1947–1952, 2019.

[Obe06] Adam M Oberman. "Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton–Jacobi equations and free boundary problems." *SIAM Journal on Numerical Analysis*, **44**(2):879–895, 2006.

[OF03] Stanley Osher and Ronald P. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer–Verlag, 2003.

[OS88] Stanley Osher and James Sethian. "Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton–Jacobi Formulations." *Journal of Computational Physics*, **79**:12–49, 1988.

[OS91] Stanley Osher and Chi-Wang Shu. "High Order Essentially Non–Oscillatory Schemes for Hamilton–Jacobi Equations." *SIAM Journal of Numerical Analysis*, **28**(4):907–922, August 1991.

[OS01] Stanley J. Osher and Fadil Santosa. "Level Set Methods for Optimization Problems Involving Geometry and Constraints: I. Frequencies of a Two-Density Inhomogeneous Drum." *Journal of Computational Physics*, **171**(1):272 – 288, 2001.

[PAB19] Christian Parkinson, David Arnold, Andrea L Bertozzi, Yat Tin Chow, and Stanley Osher. "Optimal human navigation in steep terrain: a Hamilton–Jacobi–Bellman approach." *Communications in Mathematical Sciences*, **17**(1):227–242, 2019.

[Pap85] Christos H. Papadimitriou. "An algorithm for shortest-path motion in three dimensions." *Information Processing Letters*, **20**(5):259 – 263, 1985.

[Par20] Christian Parkinson. "A Rotating-Grid Upwind Fast Sweeping Scheme for a Class of Hamilton-Jacobi Equations." *arXiv preprint*, 2020. https://arxiv.org/abs/2005.02962.

[PBG62] LS Pontryagin, VG Boltyanskij, RV Gamkrelidze, and EF Mishchenko. "The Mathematical Theory of Optimal Processes." *New York*, 1962.

[Pha09] Huyên Pham. *Continuous-time Stochastic Optimal Control and Optimization with Financial Applications*. Springer-Verlag Berlin Heidelberg, 1 edition, 2009.

[PLM06] Romain Pepy, Alain Lambert, and Hugues Mounier. "Path planning using a dynamic vehicle model." In *2006 2nd International Conference on Information & Communication Technologies*, volume 1, pp. 781–786. IEEE, 2006.

[PMO99] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. "A PDE-based fast local level set method." *Journal of Computational Physics*, **155**(2):410–438, 1999.

[Pon58] Lev Semyonovich Pontryagin. "Optimal Processes of Regulation." In *Proceedings of the International Congress of Mathematicians*, pp. 182–202. Cambridge University Press, 1958.

[PVH17] Marija Popović, Teresa A. Vidal-Calleja, Gregory Hitz, Inkyu Sa, Roland Siegwart, and Juan I. Nieto. "Multiresolution mapping and informative path planning for UAV-based terrain monitoring." *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1382–1388, 2017.

[QWH07] Yingge Qu, Tien-Tsin Wong, and Pheng Ann Heng. *Image Segmentation Using The Level Set Method*, pp. 95–122. Springer New York, New York, NY, 2007.

[Rai18] Maziar Raissi. "Forward-backward stochastic neural networks: Deep learning of high-dimensional partial differential equations." *arXiv preprint*, 2018. https://arxiv.org/abs/1804.07010.

[Ric56] Paul I Richards. "Shock waves on the highway." *Operations research*, **4**(1):42–51, 1956.

[RS90] J. A. Reeds and L. A. Shepp. "Optimal paths for a car that goes both forwards and backwards." *Pacific J. Math.*, **145**(2):367–393, 1990.

[San15] Filippo Santambrogio. "Optimal transport for applied mathematicians." *Birkäuser, NY*, **55**(58-63):94, 2015.

[Set99] J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1999.

[SF99] Mark Sussman and Emad Fatemi. "An Efficient, Interface-Preserving Level Set Redistancing Algorithm and Its Application to Interfacial Incompressible Fluid Flow." *SIAM Journal on Scientific Computing*, **20**(4):1165–1191, 1999.

[SFS98] Mark Sussman, Emad Fatemi, Peter Smereka, and Stanley Osher. "An improved level set method for incompressible two-phase flows." *Computers & Fluids*, **27**(5):663 – 680, 1998.

[Shu98] Chi-Wang Shu. "Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws." In *Advanced numerical approximation of nonlinear hyperbolic equations*, pp. 325–432. Springer, 1998.

[Shu07] Chi-Wang Shu. "High order numerical methods for time dependent Hamilton–Jacobi equations." In *Mathematics and computation in imaging science and information processing*, pp. 47–91. World Scientific, 2007.

[SL16] Deepak N. Subramani and Pierre F.J. Lermusiaux. "Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization." *Ocean Modelling*, **100**:57 – 77, 2016.

[SLV17] Ricardo Samaniego, Joaquin Lopez, and Fernando Vazquez. "Path planning for non-circular, non-holonomic robots in highly cluttered environments." *Sensors*, **17**(8):1876, 2017.

[Sou85] Panagiotis E Souganidis. "Approximation schemes for viscosity solutions of Hamilton-Jacobi equations." *Journal of Differential Equations*, **59**(1):1 – 43, 1985.

[SSO94a] Mark Sussman, Peter Smereka, and Stanley Osher. "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow." *Journal of Computational Physics*, **114**(1):146 – 159, 1994.

[SSO94b] Mark Sussman, Peter Smereka, and Stanley Osher. "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow." *Journal of Computational Physics*, **114**(1):146 – 159, 1994.

[SSW13] Ashwini Shukla, Ekta Singla, Pankaj Wahi, and Bhaskar Dasgupta. "A direct variational method for planning monotonically optimal paths for redundant manipulators in constrained workspaces." *Robotics and Autonomous Systems*, **61**(2):209–220, 2013.

[ST91] Héctor J Sussmann and Guoqing Tang. "Shortest paths for the Reeds-Shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control." *Rutgers Center for Systems and Control Technical Report 10*, pp. 1–71, 1991.

[SU18] T. Slough and J. Urpelainen. "Public Policy Under Limited State Capacity: Evidence from Deforestation Control in the Brazilian Amazon." Technical report, 2018.

[SUA16] C. Saranya, Manju Unnikrishnan, S. Akbar Ali, D.S. Sheela, and Dr. V.R. Lalithambika. "Terrain Based D* Algorithm for Path Planning." *IFAC-PapersOnLine*, **49**(1):178 – 182, 2016. 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2016.

[SV01] James A. Sethian and A. Vladimirsky. "Ordered upwind methods for static Hamilton-Jacobi equations." *Proceedings of the National Academy of Sciences*, **98**(20):11069–11074, 2001.

[SV03] James A. Sethian and A. Vladimirsky. "Ordered Upwind Methods for Static Hamilton-Jacobi Equations: Theory and Algorithms." *SIAM Journal on Numerical Analysis*, **41**(1):325–363, 2003.

[SV15] Z. Shen and A. Vladimirsky. "Piecewise-Deterministic Optimal Path Planning." *arXiv preprint*, 2015. https://arxiv.org/abs/1512.08734.

[SWL18] Deepak N. Subramani, Quantum J. Wei, and Pierre F.J. Lermusiaux. "Stochastic time-optimal path-planning in uncertain, strong, and dynamic flows." *Computer Methods in Applied Mechanics and Engineering*, **333**:218 – 237, 2018.

[TBE01] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. "A Front-Tracking Method for the Computations of Multiphase Flow." *Journal of Computational Physics*, **169**(2):708 – 759, 2001.

[TCO03] Yen-Hsi Richard. Tsai, Li-Tien. Cheng, Stanley. Osher, and Hong-Kai. Zhao. "Fast Sweeping Algorithms for a Class of Hamilton–Jacobi Equations." *SIAM Journal on Numerical Analysis*, **41**(2):673–694, 2003.

[TCO04] Y.-H.R. Tsai, L.-T. Cheng, S. Osher, P. Burchard, and G. Sapiro. "Visibility and its dynamics in a PDE based implicit framework." *Journal of Computational Physics*, **199**(1):260 – 290, 2004.

[Tob93] W. Tobler. "Three presentations on geographical analysis and modeling." Technical Report 93-1, National Center for Geographic Information and Analysis, February 1993.

[Tom05] Paul Tompkins. *Mission-directed path planning for planetary rover exploration.* 2005.

[TR06] Zhichao Tan and Rolf D. Reitz. "An ignition and combustion model based on the level-set method for spark ignition engine multidimensional modeling." *Combustion and Flame*, **145**(1):1 – 15, 2006.

[Tsi95] J. N. Tsitsiklis. "Efficient algorithms for globally optimal trajectories." *IEEE Transactions on Automatic Control*, **40**(9):1528–1538, Sep 1995.

[TT13] Ryo Takei and Richard Tsai. "Optimal Trajectories of Curvature Constrained Motion in the Hamilton-Jacobi Formulation." *Journal of Scientific Computing*, **54**(2):622–644, Feb 2013.

[TTS10] R. Takei, R. Tsai, H. Shen, and Y. Landa. "A practical path-planning algorithm for a simple car: a Hamilton-Jacobi approach." In *Proceedings of the 2010 American Control Conference*, pp. 6175–6180, June 2010.

[TZ06] Lijian Tan and Nicholas Zabaras. "A level set simulation of dendritic solidification with combined features of front-tracking and fixed-domain methods." *Journal of Computational Physics*, **211**(1):36 – 63, 2006.

[VC02] Luminita A Vese and Tony F Chan. "A multiphase level set framework for image segmentation using the Mumford and Shah model." *International Journal of Computer Vision*, **50**(3):271–293, 2002.

[VDS08] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl. "Time-energy optimal path tracking for robots: a numerically efficient optimization approach." In *2008 10th IEEE International Workshop on Advanced Motion Control*, pp. 727–732, March 2008.

[VF18] Valerio Varricchio and Emilio Frazzoli. "Asymptotically optimal pruning for nonholonomic nearest-neighbor search." In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4459–4466. IEEE, 2018.

[WCW00] Weiguo Wu, Huitang Chen, and Peng-Yung Woo. "Time optimal path planning for a wheeled mobile robot." *Journal of Robotic Systems*, **17**(11):585–591, 2000.

[WDG19] Chenming Wu, Chengkai Dai, Xiaoxi Gong, Yong-Jin Liu, Jun Wang, Xianfeng David Gu, and Charlie CL Wang. "Energy-efficient coverage path planning for general terrain surfaces." *IEEE Robotics and Automation Letters*, **4**(3):2584–2591, 2019.

[WSC19] Fangyu Wu, Raphael E. Stern, Shumo Cui, Maria Laura Delle Monache, Rahul Bhadani, Matt Bunting, Miles Churchill, Nathaniel Hamilton, R'mani Haulcy, Benedetto Piccoli, Benjamin Seibold, Jonathan Sprinkle, and Daniel B. Work. "Tracking vehicle trajectories and fuel rates in phantom traffic jams: Methodology and data." *Transportation Research Part C: Emerging Technologies*, **99**:82 – 109, 2019.

[WSY19] Y. Wang, M. Shan, Y. Yue, and D. Wang. "Vision-Based Flexible Leader-Follower Formation Tracking of Multiple Nonholonomic Mobile Robots in Unknown Obstacle Environments." *IEEE Transactions on Control Systems Technology*, pp. 1–9, 2019.

[WWW08] Tianmiao Wang, Bin Wang, Hongxing Wei, Yunan Cao, Meng Wang, and Zili Shao. "Staying-alive and energy-efficient path planning for mobile robots." In *2008 American Control Conference*, pp. 868–873. IEEE, 2008.

[ZDH18] Z. Zhou, J. Ding, H. Huang, R. Takei, and C. Tomlin. "Efficient path planning algorithms in reach-avoid problems." *Automatica*, **89**:28 – 36, 2018.

[Zha05] Hongkai Zhao. "A fast sweeping method for eikonal equations." *Mathematics of computation*, **74**(250):603–627, 2005.