Three-Dimensional Time-Optimal Path Planning in Dynamic and Realistic Environments

by

Chinmay Sameer Kulkarni

B.Tech, Indian Institute of Technology Bombay (2015)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

Author .... Signature redacted ...
Department of Mechanical Engineering
May 12, 2017

Certified by.................. Signature redacted ..
Pierre F.J. Lermusiaux
Professor
Thesis Supervisor

Accepted by .... Signature redacted .
Rohan Abeyratne
Chairman, Department Committee on Graduate Theses

Three-Dimensional Time-Optimal Path Planning in Dynamic and Realistic Environments

by

Chinmay Sameer Kulkarni

Submitted to the Department of Mechanical Engineering
on May 12, 2017, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

Autonomous underwater vehicles (AUVs) are a valuable resource in several oceanic applications such as security, surveillance and data collection for ocean prediction. These vehicles typically travel at speeds comparable to ocean currents, and their movement is significantly affected by these dynamic currents. Further, the speed of currents may vary greatly with depth. Hence, path planning to generate safe and fast vehicle trajectories in such a three-dimensional environment becomes crucial for the successful operation of these vehicles. In addition, many marine vehicles can only move in specific directions and with a speed that is dependent on the direction of travel. Such constraints must be respected in order to plan safe and optimal paths.

Thus, our motivation in this thesis is to study path planning for vehicles with and without motion constraints in three-dimensional dynamic flow-fields. We utilize the time-optimal path planning methodology given by Lolla et al. (2012) for this purpose.

In this thesis, we first review some existing path planning methods (both in two and three-dimensional settings). Then, we discuss the theoretical basis of the rigorous partial differential equation based methodology that is utilized in order to plan safe and optimal paths. This is followed by an elaborate discussion about the application of this methodology to the various types of marine vehicles. We then look at the robust and accurate numerical methods developed in order to solve the governing equations for the path planning methodology with high accuracy in real ocean domains. We illustrate the working and capabilities of our path planning algorithm by means of a number of applications. First we study some benchmark examples with known analytical solutions. Second, we look at more complex flow-fields that analytically model different oceanic flows. Finally, we look at the path planning for different types of marine vehicles in a realistic ocean domain to illustrate the capabilities of the path planning methodology and the developed numerical framework.

Thesis Supervisor: Pierre F.J. Lermusiaux
Title: Professor

Acknowledgments

I would like to thank my advisor Prof. Pierre Lermusiaux for his guidance, support and for allowing me to work on research problems of my interest. His motivation, words of encouragement and constructive criticism have gone a long way in helping me develop a wide research perspective.

I thank Dr. Pat Haley for helping the realistic ocean examples studied in this work. I am also grateful to Dr. Chris Mirabito, Dr. Sudip Jana and Dr. Yulin Pan as well as Marcia and Leslie for their help.

I am grateful to the Office of Naval Research for support under Grants N00014-15-1-2616 (DRI-NASCar) and N00014-14-1-0476 (Science of Autonomy LEARNS) to the Massachusetts Institute of Technology (MIT).

Thanks to my MSEAS family for being a home away from home! Thank you Deepak for always being around and for helping me whenever needed. I will always cherish our numerous late night discussions. Thanks Johnathan for being a great friend and for sharing my enthusiasm for good food. I can never forget our sailing adventures! Thanks to Corbin, Abhinav, Florian and Yukino for being the best labmates anyone could ask for. Thank you Arkopal, Jing and Wael for your support, friendship and the many fun-filled conversations. Special thanks to Tapovan, Sydney, John, Matt and the other senior students for helping me settle into the group.

Life at MIT has been an amazing experience. I would like to thank all my friends at MIT and at IIT Bombay for always being there, looking out for me and being a great company over all these years! Finally, I would like to thank my family, both back home and here in the United States. None of this would have been possible without their constant encouragement and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

The problem of planning feasible, safe and optimal paths in complex, dynamic environments has received a great deal of attention from many branches of science and engineering. In the most general sense, 'path planning' refers to a set of rules provided to an autonomous robot which enables the unit to navigate from one configuration to another, in an 'optimal' fashion. Typically, such optimality is governed by some objective function. Autonomous robotic platforms are becoming ubiquitous day by day, and are used to perform a variety of tasks with different levels of complexity. Such scenarios require the robots to optimize a wide range of objective functions, and hence there does not exist a universal path planning theory that is applicable to all possible situations.

Optimal navigation of autonomous underwater vehicles (AUVs) is crucial for many applications, ranging from security to search and rescue to data collection. Underwater gliders and floats are ideal for such missions due to their high levels of autonomy and long-range endurance. However, these vehicles typically travel at relatively slow speeds, and in many cases the speed of the local ocean currents becomes comparable to this speed (Schmidt et al., 1996; Elisseeff et al., 1999; Yan et al., 2014). Hence, the effect of such ocean currents on the motion of the vehicle cannot be neglected. Further, the environment in which these vehicles operate, i.e.; the oceans, is a highly dynamic and multi-scale system with high spatio-temporal variability. Thus, it is extremely challenging to develop accurate and efficient methodologies to plan paths optimize a desired objective function while accommodating both the environmental constraints and the vehicle limitations.

Robotic path planning is a field of active research. This field has been extensively studied, with many key breakthroughs. However, general robotic path planning and underwater path planning differ at a fundamental level. Most research pertaining to underwater path planning has focused on the direct extension of methodologies for robotic path planning. Underwater path planning focuses on autonomous vehicles such as propelled AUVs but also gliders and floats that execute long-range missions (order of days to months). Consequently, it becomes important to optimize performance parameters such as the energy spent as well as the quality of data collected in addition to generating feasible and safe trajectories. The difficulty in underwater path planning is primarily due to the dynamic environmental fields that directly and indirectly affect the motion of the vehicle as well as the possibility of infinitely many possible movement choices at any time. This makes it a difficult problem to even generate feasible trajectories for oceanic vehicles.

Work has gone into extending approaches such as the A* algorithm (Hart et al., 1968) and the rapidly exploring random trees (RRTs; LaValle (1998); LaValle and Kuffner Jr (2000); Kuffner and LaValle (2000) or other graph search based methods towards underwater path planning (Alvarez et al., 2004; Garau et al., 2005). The A* scheme converts the path planning problem into a graph search problem (Dechter and Pearl, 1985). It first looks at the feasibility of paths between adjacent cells after the entire network is generated, solves the graph search problem by any of the known approaches, such as the depth-first search, breadth-first search, Djikstra's method (Johnson, 1973; Dijkstra, 1959), etc. The A* method makes use of heuristics, which provides the estimate of the cost of the best route that passes through a particular node. The choice of heuristic heavily determines the speed of convergence (Hart et al., 1968). Although this method is widely used by the robotics community, its dependence on heuristics, and inability to account for dynamic environmental fields implies that it is not directly applicable to accurate underwater path planning. Rapidly exploring random trees (RRTs) are a randomized data structure that is widely used in several streams of path planning research (LaValle, 1998). RRTs work by incrementally building a randomized search tree from the start position to the end position. Biasing may be added to RRTs in order to account for directionality in the movement of the vehicle as well as the effects of the background dynamic flow field (Urmson and Simmons, 2003; Huynh et al., 2014; Heo and Chung, 2013). Although RRTs efficiently sample high-dimensional and non-convex search spaces, the paths predicted are often sub-optimal (Bry and Roy, 2011). Further, the effects of dynamic flow-fields cannot be accurately incorporated in this formulation. Both these drawbacks make it impractical for RRTs to be readily adapted for underwater path planning.

Recently, Lolla et al. (2012, 2014a,b) proposed an exact methodology for underwater path planning that accurately accounts for the effects of the dynamic environmental flow-field on the vehicle motion, as well as predicts the globally optimal paths without the requirement of any heuristics. The governing equations for time-optimal path planning are solved forward in time using the level set method to study the evolution of the set of all points that are reachable at a certain instant, also called the reachability set. The optimal path is then given by the trajectory of the point that always traveled on the boundary of this reachability set and crossed the destination in optimal time. This methodology has been shown to be exact (Lolla and Lermusiaux, 2017), computationally efficient (Lolla et al., 2012) and is able to account for unsafe / forbidden regions and stationary or moving obstacles (Lolla et al., 2015).

Subramani et al. (2017b); Edwards et al. (2017); Mirabito et al. (2017) conducted multiple sea-trials to experimentally demonstrate the effectiveness of the said algorithm. The AUV following the optimal path required up to 15% less travel time than the AUV following a straight line path. This conclusively demonstrates the applicability of the said algorithm. Further, this work has also been extended to compute the paths that require the minimum amount of energy (Subramani and Lermusiaux, 2016; Subramani et al., 2017a). It can also be used towards designing paths that maximize the quality of data collected by the autonomous vehicle (Lolla, 2016).

Existing work in the area has only looked at the applications of this methodology two-dimensional or quasi-two-dimensional scenarios. However, oceanic currents often differ greatly with depth (Cushman-Roisin and Beckers, 2011). In order to reach the destination in optimal time, underwater vehicles can utilize the favorable currents and also avoid adverse currents by diving or rising to appropriate depths. This implies that the extension of this methodology to fully three-dimensional real ocean domains is vital. Carrying out simulations over real ocean domains is numerically extremely challenging. Oceans are one of the most complex dynamical systems, with multiple time-scales that span from a fraction of a second to multiple years and length-scales that span from several millimeters to hundreds of kilometers (Cushman-Roisin and Beckers, 2011). Such a multi-scale nature of the setup implies that small disturbances in some scales may cause significant errors in the other scales. This prompts the development and application of robust numerical techniques and highly accurate numerical schemes to obtain the results with high confidence. Further, there exist several different types of marine vehicles with specific degrees of freedom and motion constraints. Typical marine gliders used for sampling data perform a sinusoidal motion (Rudnick et al., 2004; Testor et al., 2010; Javaid et al., 2014) whereas floats that drift with the ocean currents are only able perform vertical motion by adjusting their buoyancy (Roemmich et al., 2009; Kobayashi et al., 2012). In order to plan feasible, safe and optimal trajectories for such vehicles, we need to respect such motion constraints and hence special treatment is required to account for constrained motion in the methodology. The desired attributes listed above are crucial to the application of this exact optimal planning methodology to real ocean scenarios.

Optimal path planning in three-dimensional domains is complex, and hence not many people have studied algorithms for the same beyond two-dimensions. However, utilizing environmental flows and planning feasible paths in three-dimensions has been elaborately studied by several researchers from a wide variety of areas. For example, Kiraly et al. (2004) look at computing feasible three-dimensional trajectories for a bronchoscope in the various airways inside the human body. Similarly, Siddon (1985) looks at planning paths for three-dimensional CT arrays in radiology.

Planning feasible and collision-free trajectories for aerial vehicles is an active field, with many established approaches. Lozano-Pérez and Wesley (1979) look at a network-based approach to plan paths in domains with forbidden regions and obstacles. Mittal and Deb (2007) attempt to plan feasible paths offline using evolutionary algorithms. Wong and Fu (1986) uses a network based approach on two-dimensional sections of the three-dimensional domain for computational efficiency. This idea of planning collision-free paths is also extended to larger aircraft and around congested airports. Most authors utilize optimization techniques similar to integer programming for such cases. For example, Richards and How (2002) uses mixed integer linear programming in order to plan aircraft trajectories with collision avoidance. Frazzoli et al. (2001) approach the same problem using semi-definite programming. Prete and Mitchell (2004); Krozel et al. (2006) look at network based approaches to compute safe aircraft trajectories that minimize travel in time-varying three-dimensional hazardous weather regions.

The various biological systems in nature also serve as a motivation for optimal planning problems. Richardson (2012); Sachs et al. (2013) look at the approaches to mimic the dynamic soaring of the Albatross for autonomous aerial vehicles in order to optimize their energy consumption. These birds utilize the lee eddies located downwind of sharp-crested waves in order to increase their speed. Such a maneuver requires minor adjustments to the trajectory but saves significant amount of energy. Such a dynamic soaring pattern has been used in unmanned aerial vehicles to utilize environmental flows in order to travel at faster speeds (Bonnin, 2016). Other similar dynamic soaring patterns to minimize the energy spent during travel are also studied by Zhao (2004); Zhao and Qi (2004). Some authors have also considered the path planning problem from a continuum viewpoint, eliminating the need for network based models. For example, Connolly et al. (1990); Wang and Chirikjian (2000) look at robotic path planning in three dimensions by using the potential field method. In this method, it is assumed that all obstacles exert a repelling force whereas the target position exerts an attractive force on a potential function. Although this method looks at path planning from a continuous standpoint, heuristics are needed to account for the effect of external flow fields on the computed path.

Due to the strong dynamic flows and vehicle constraints, the above mentioned three-dimensional path planning algorithms cannot be directly applied for underwater path planning. Some advances have been made in order to plan optimal paths in the three-dimensional ocean domains. Garau et al. (2014) look at planning optimal glider paths using the A* algorithm in three dimensions in the western Mediterranean sea. As mentioned before, even though the A* algorithm is efficient, its dependence on heuristics does not guarantee global optimality. Witt and Dunbabin (2008) who build upon the work done by Kruger et al. (2007) to find paths that minimize the energy spent during the travel of an autonomous vehicle in three-dimensional oceanic domains. This is achieved by using the present current to the greatest advantage. This method efficiently chooses an energy optimal path amongst the considered candidate paths. However, as all the possible paths can never be considered, this method also cannot ensure strict global optimality of the paths.

Pereira et al. (2013) provide an algorithm to compute three-dimensional vehicle paths that minimize the risk of collision with ships and land. They use the Regional Ocean Modeling System (ROMS) to obtain oceanic flow-fields and then use a Markov decision process based approach as well as a expectation minimization based approach to decrease the risk of collision. Further, an experimental example of a Slocum glider is demonstrated in order to elaborate of the efficiency of this methodology. Smith et al. (2010) further extend this methodology in order to track a dynamically evolving ocean feature while minimizing the risk of collision with ships and other possible obstacles.Petillot et al. (2001) utilize data from an on-board sonar sensor in order to plan paths that minimize the risk of collisions with objects and obstacles in the ocean. This approach is efficient in that it does not require any offline computation, and the routing is completely performed on-board. However, very few oceanic vehicles are equipped with a sonar (or any other similar sensor), and hence such an algorithm cannot be used for all types of oceanic vehicles.

Another approach towards optimal path planning is to use a myopic optimization, where the vehicle chooses the short term optimal option amongst all the possible options. Although this method cannot guarantee global optimality, its ease of implementation makes it a preferred choice for some researchers. For example, Zamuda and Sosa (2014) look at a glider that gathers data while following a path towards some set destination. A short-term opportunistic sampling algorithm is used to maximize the quality of the data collected.

After a brief review of the existing work on three-dimensional path planning, we now look at the goal of this thesis, which is summarized in the following section.

1.2 Goal of this Work

In this thesis, we aim to *extend the fundamental time-optimal path planning methodology to fully 3 dimensional flow fields and also to vehicles with heading-dependent speeds using high order accurate and consistent numerical schemes that are applicable for complex real ocean domains and dynamics.* Specifically, this work involves: (i) extending the theoretical basis in order to study path planning for different types of marine vehicles; (ii) developing novel high order accurate numerical schemes on suitable non-dimensional forms of the governing equations, and (iii) efficiently implementing the developed theory and schemes in order to compute time-optimal paths in realistic dynamic ocean environments.

1.3 Layout of Thesis

In this thesis, we look at the exact time-optimal path planning methodology for three-dimensional realistic ocean fields, as well as optimal planning strategies for the different types of oceanic vehicles. The second chapter discusses the problem formulation along with the relevant assumptions. It then reviews the differential equations governing forward reachability and time-optimal paths, and the use of the level set method - a tool central for the efficient and accurate solution of the path planning problem. Then, a more general version of the theorem first mentioned in Lolla et al. (2012) that considers motion of vehicles with heading-dependent speeds is stated and proved. Lolla (2016); Lolla and Lermusiaux (2017) look at this problem from the point of view of optimal control to arrive at the same result. This is followed by the application of this theorem for to study the optimal motion of various types of oceanic vehicles with different motion constraints, such as gliders, floats, etc. Finally, we look at the case where the motion of the vehicle is deterministic in certain directions. We show that such a constrained motion of the vehicle reduces the dimensionality of the problem, that is in such cases, a three-dimensional path planning problem is decomposed in to a two-dimensional version with appropriately constructed external velocity field and vehicle speed profile.

The third chapter then looks at the development of novel robust numerical treatments and schemes to accurately solve the path planning problem for real ocean domains. These issues arise primarily because these domains are extremely skewed in length and velocity scales. We first discuss a non-dimensional form of the Hamilton-Jacobi level set equation that is insensitive towards variable scales. This is followed by high order spatial and temporal schemes that allow us to accurately study the evolution of the level set function, governed by the aforementioned non-dimensional equation. Finally, we discuss the forward-backward consistent implicit backward tracing schemes that are in accord with the forward numerics, and accurately backtrack the optimal trajectory as well as compute the corresponding optimal headings.

Finally, chapter 4 looks at the various applications of the developed theoretical and numerical results. First, several benchmarking tests are performed in order to ensure the order of accuracy of the numerical schemes. This is followed by two simple cases and the results are compared with the analytical solutions. We then move onto more complicated three-dimensional analytical flows, such as the double gyre flow and the Arnold-Beltrami-Childress (ABC) flow. Several unique characteristics of our methodology are highlighted through these examples. Finally, we look at examples of the application of our methodologies to fully 3 dimensional realistic oceanic flow fields, and for various types of marine vehicles.

Chapter 5 summarizes the work done, highlights the main contributions of this thesis, and discusses some promising directions for future research in this area.

Chapter 2

Time-Optimal Path Planning: Theory

2.1 Review of the Level Set Method and Hamilton-Jacobi Equations

In this section, we introduce the level set method and discuss its utility in problems involving interface tracking. The governing equations for time-optimal path planning are solved forward in time to track the boundary of the set of all reachable points (the reachability set), for which the level set method is employed (Lolla et al., 2012, 2014a,b). Hence, in this section, we explain the theoretical basis of the level set method that allow the reader to completely understand the methodology presented in section §(2.4) as well as the numerical details discussed in chapter 3. The evolution of this aforementioned reachability set is governed by a Hamilton-Jacobi equation, which is a type of hyperbolic partial differential equations (PDEs). In this section, we also go through a review of Hamilton-Jacobi equations in order to understand their peculiar characteristics and to design efficient numerical schemes. Specifically, we discuss the fundamentals of the level set method, interface tracking, the basics of the Hamilton-Jacobi equations and their connections with viscosity solutions of hyperbolic conservation laws.

2.1.1 Level Set Method

The key motivation behind the level set method is to allow us to represent, track and study the properties of an interface as it evolves in space and time. Developed by Osher and Sethian (1988), the level set method provides us a way to implicitly represent the interface and track its evolution. This is in turn the Eulerian perspective of the interface tracking problem, as opposed to the particle-based (Lagrangian) approach, where hypothetical particles are seeded along the initial interface and their evolution is studied by solving an ordinary differential equation for each of the particles. Although the particle based method is more intuitive, faster to implement and faster in computation due to only localized computations; there are several major pitfalls of this approach, some of which are mentioned below:

- Particle based methods induce distortions in the shape of the interface as it evolves. Especially if the velocity field varies greatly, then particles might be concentrated in some parts of the interface, and some regions may have sparse particle density. This can induce unnatural corners and distortions. A way to alleviate this problem is to re-seed particles along the evolved interface repeatedly after certain duration. Although this approach may decrease the severity of these drawbacks, it cannot completely nullify them. It is also computationally expensive and suffers in accuracy.

- As discussed later in this section, our problem reduces to studying the solution of a hyperbolic conservation law. In general, hyperbolic partial differential equations admit locally non-differentiable solutions, and the interface may be bent, twisted and pinched in such way by the flow that it crosses itself. Particle based approaches lack the detailed treatment of such cases and it becomes difficult to handle the cases where the interface crosses itself.

- It has also been shown that particle based approaches suffer from stability issues when the interface evolves with curvature dependent speed (Persson, 2005))

These issues become of paramount importance while trying to obtain solutions of hyperbolic equations, which prompts us to utilize the level set method. The core of level set methods lies in embedding an iso-contour along the interface of a function with one higher dimension than the interface itself. This 'level set' function is defined on a fixed grid independent of the position of the interface, which allows us to embrace an Eulerian perspective.

The level set of any function $f : \mathbb{R}^n \to \mathbb{R}$ is the set of points along which the function f takes a constant value. Mathematically, this translates to the following set: $\{\mathbf{x} \mid f(\mathbf{x}) = c\}$, where c is a known (or given) constant. This eases the problem of interface tracking in arbitrary dimensions, as instead of explicitly tracking the interface, we can now embed a level set of a function defined over the entire domain and track the evolution of this interface by means of the embedded level set, typically by solving an initial-value partial differential equation for the level set function. That is., given an open domain $\Omega \subseteq \mathbb{R}^n$ and a hypersurface $\partial\Omega \subseteq \mathbb{R}^{n-1}$, we construct a Lipschitz continuous function $\phi : \mathbb{R}^n \to \mathbb{R}$, such that $\partial\Omega$ forms a specific level set of the function ϕ. Typically in literature as well as in this work, we construct ϕ such that $\partial\Omega$ is the zero level set of ϕ (i.e. $\partial\Omega = \{\mathbf{x} \mid \phi(\mathbf{x}) = 0\}$).

Note that, the general form this approach requires information about the dynamics of the entire domain to study interface evolution. It may be the case that the information about such global dynamics is unavailable. As will be discussed in section 3.4, new methods have been developed whereby this drawback is addressed. The benefit of this formulation is that the level set method gives us the ability to represent the interface implicitly, and enables its tracking through solving an evolution equation on a fixed (or at least, known) domain, thus eliminating the particle-related issues of the Lagrangian based approach.

In all front tracking problems, the interface ($\partial\Omega$) divides the domain Ω in 3 parts: the inner region (Ω^-), the outer region (Ω^+) and the interface ($\partial\Omega$) itself. We desire for the level set function to keep a track of whether any point lies 'inside', 'outside' or on the interface. For this purpose, we adopt the following sign convention for our level set function ϕ:

$$\phi(\mathbf{x},t)\begin{cases} >0 & \text{for } \mathbf{x}\in\Omega^+ \ (\mathbf{x} \text{ outside } \partial\Omega) \\ <0 & \text{for } \mathbf{x}\in\Omega^- \ (\mathbf{x} \text{ inside } \partial\Omega) \\ =0 & \text{for } \mathbf{x}\in\partial\Omega \ (\mathbf{x} \text{ on } \partial\Omega) \end{cases} \tag{2.1}$$

Figure 2-1 depicts the signs of the level set field in the different regions, for an arbitrary front in two dimensions. Same concepts extend to three-dimensional fields, for which such a depiction becomes harder.

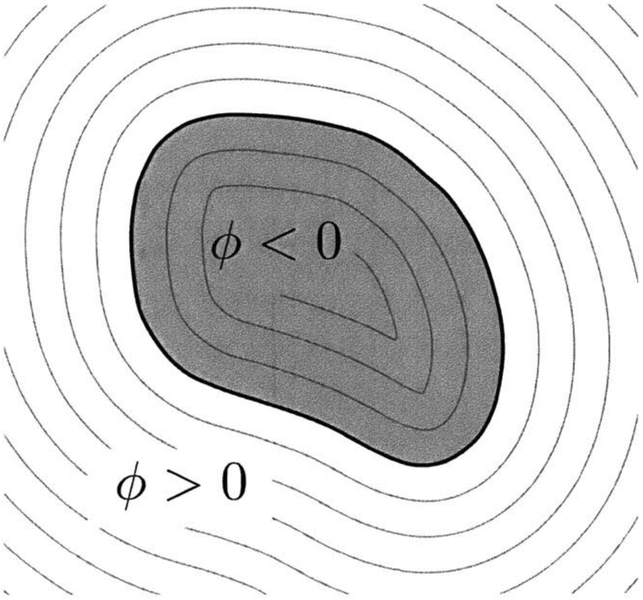


Figure 2-1: Sign convention for $\phi(\mathbf{x},t)$. The shaded gray region lies inside the considered zero level set and hence $\phi < 0$ in this region. The white region is outside the zero level set, which implies $\phi > 0$. The other iso-contours of ϕ are shown for reference.

This convention is followed throughout this work, and allows us to easily determine whether a point lies inside or outside the interface at any time.

In order to study how the front evolves due to some externally imposed velocity field, we resort to the duality between the Lagrangian and the Eulerian viewpoints through the material (or total) derivative. The material derivative of some function

f in a velocity field $\frac{d\mathbf{x}}{dt}$ is defined as:

$$\frac{Df}{Dt} = \frac{\partial f(\mathbf{x},t)}{\partial t} + \frac{d\mathbf{x}}{dt} \cdot \nabla f(\mathbf{x},t) \tag{2.2}$$

The left side of equation (2.2) is the Lagrangian derivative, and right side is its Eulerian representation. But, the function value for any Lagrangian point in the domain does not change in time. Instead, the function value is only advected with the point itself. This implies that $\frac{Df}{Dt} = 0$. Hence, we have:

$$\frac{\partial f(\mathbf{x},t)}{\partial t} + \frac{d\mathbf{x}}{dt} \cdot \nabla f(\mathbf{x},t) = 0 \tag{2.3}$$

Consider any point $\mathbf{x}$ on a level set of the function ϕ, that is $\phi(\mathbf{x},t) = c$. The velocity experienced by this point at that instant is identical to the velocity at the point $\mathbf{x}$, that is $\frac{d\mathbf{x}}{dt}$.

We first consider the case where an interface moves in a direction normal to itself with a constant speed F. This means that every point on the considered level set of the function ϕ experiences a velocity equal to $F\hat{\mathbf{n}}$, where $\hat{\mathbf{n}}$ is the local normal direction. Thus equation (2.3) now becomes:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} + F\hat{\mathbf{n}} \cdot \nabla \phi(\mathbf{x},t) = 0 \tag{2.4}$$

Now we consider the case of an externally imposed velocity field $\mathbf{V}(\mathbf{x},t)$. This simply implies that

$$\frac{d\mathbf{x}}{dt}|_{(\mathbf{x},t)} = \mathbf{V}(\mathbf{x},t) \tag{2.5}$$

Substituting equation (2.5) in equation (2.3), we obtain equation (2.6), which allows us to track the evolution of function ϕ in an externally imposed velocity field.

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} + \mathbf{V}(\mathbf{x},t) \cdot \nabla \phi(\mathbf{x},t) = 0 \tag{2.6}$$

Even though equation (2.4) and equation (2.6) are initial value partial differential equations for the level set function ϕ, one can always recover the interface at any time t by considering the zero level set (iso-contour with $c = 0$) of $\phi(\mathbf{x}, t)$. Note that, if initially the value of ϕ along the interface was different from 0 (some c), then one should consider the corresponding 'c level set' (iso-contour bearing the value c) to retrieve the interface position.

Equation (2.4) and equation (2.6) can be superposed to obtain equation (2.7) in the case where the interface is moving in an imposed velocity field $\mathbf{V}$, and is also evolving with a speed F normal to itself.

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} + (\mathbf{V}(\mathbf{x},t) + F\hat{\mathbf{n}}) \cdot \nabla\phi(\mathbf{x},t) = 0 \tag{2.7}$$

Equation (2.7) can be looked at as the front evolution for the case where the effective total velocity at any point $\mathbf{x}$ on the front is $(\mathbf{V}(\mathbf{x},t) + F\hat{\mathbf{n}})$. This forms the basis of level set method and also the starting point for section §(2.4).

Note that, in some cases the velocity field away from the interface may not be known. In this formulation, even though we solve for the level set function ϕ over the entire domain, the values of ϕ away from the interface does not affect the evolution of the interface. Hence one can use any Lipschitz continuous velocity field to substitute for the regions where the velocity field is not known. It can also be noted that equation (2.7) is an unsteady Hamilton-Jacobi equation (Sethian, 1998). Weak solutions to these equations allow for formation of singularities and shocks, and solutions with mild singularities is typical of these equations, as will be seen later.

In order to study the front evolution, equation (2.7) is solved on a discretized domain with a numerical procedure. The accuracy of the numerical method and parameters such as the discretization size and the time stepping control the accuracy of the solution. In chapter 3 we discuss our development and implementation of high order accurate numerical schemes required to exactly study the front propagation and growth.

We now discuss the choice of the level set function ϕ. As mentioned before, any arbitrary Lipschitz continuous function can be used as the level set function. A preferred choice for the implicit level set function ϕ is the signed distance function $d(\mathbf{x})$, because it is smooth and maintains fixed gradients in the iso-contours (Sethian, 1999a)), especially near the zero level set. Simply, the signed distance function value at any point $\mathbf{x}$ is the minimum distance between $\mathbf{x}$ and the interface $\partial\Omega$. As per our convention discussed in equation (2.1), $d(\mathbf{x})$ is negative for $\mathbf{x}$ inside the front and is positive for $\mathbf{x}$ outside. By definition, $d(\mathbf{x}) = 0$ for $\mathbf{x} \in \partial\Omega$, consistent with the zero level set representation of the front. Equation (2.7) illustrates the definition of the signed distance function.

$$d(\mathbf{x}) \begin{cases} = \min\limits_{\mathbf{x}_0 \in \partial\Omega} |\mathbf{x} - \mathbf{x}_0| \text{ for } \mathbf{x} \in \Omega^+ \\ = -\min\limits_{\mathbf{x}_0 \in \partial\Omega} |\mathbf{x} - \mathbf{x}_0| \text{ for } \mathbf{x} \in \Omega^- \\ = 0 \text{ for } \mathbf{x} \in \partial\Omega \end{cases} \tag{2.8}$$

A defining property of of the signed distance function is that it is the unique viscosity solution of the static Hamilton-Jacobi (Eikonal) equation given by equation (2.9).

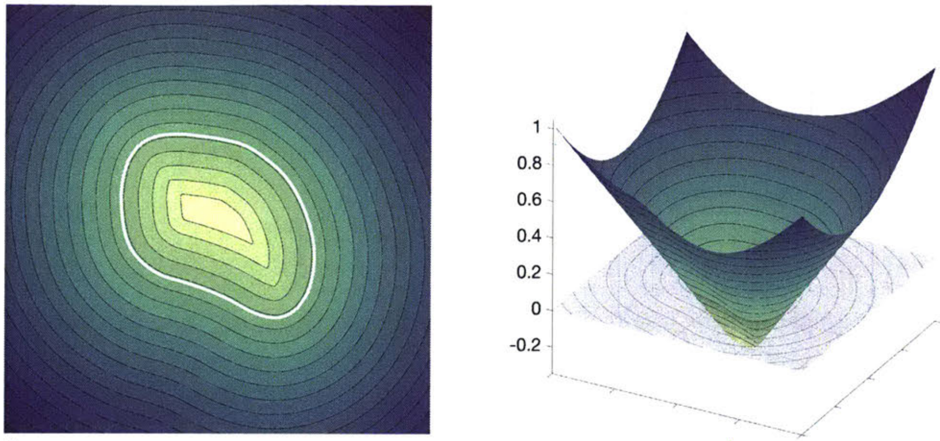
$$|\nabla d(\mathbf{x})| = 1 \tag{2.9}$$

This property has inspired development of fast and efficient schemes for constructing a signed distance field over a given domain (with respect to a known front), some of which are discussed in section §(3.4).

Figure 2-2 shows signed distance function contours in a square Cartesian domain with respect to a closed curve. Note that the 3D representation of ϕ is a cone-like surface, which is a direct implication of equation (2.9). Note that the cross-section of this shape depends on the curve with respect to which the signed distance is computed.

2.1.2 Hamilton-Jacobi Equations

As mentioned in the above discussion, equation (2.7) is a Hamilton-Jacobi equation. In this section, we discuss the main traits of Hamilton-Jacobi equations and the



(a) signed distance function: 2D contours

(b) signed distance function: 3D view

Figure 2-2: Signed distance function with respect to a closed curve in 2 dimensions. The curve is represented in white, and $\phi = 0$ on the curve. If plotted in 3D, ϕ assumes a distorted conical surface as seen from the second figure.

possibility of the formation of weak solutions. The allowance of weak solutions is important in designing numerical schemes, as the scheme should be able to account for locally non-differentiable solutions.

An unsteady Hamilton-Jacobi equation is of the form given by equation (2.10) (Osher and Fedkiw, 2006))

$$\frac{\partial \phi}{\partial t} + H(\mathbf{x}, t, \phi, \nabla\phi) = 0 \tag{2.10}$$

For example, equation (2.6) is an example of Hamilton-Jacobi equation, where $H(\mathbf{x}, t, \phi, \nabla\phi) = \mathbf{V} \cdot \nabla\phi$.

Hamilton-Jacobi equations depend on (at most) the first derivatives of the argument ϕ and are hyperbolic in nature. Further, a direct correspondence can be drawn between Hamilton-Jacobi equations and conservation laws. In particular, the solution u to a conservation law is the derivative of a solution ϕ to a Hamilton-Jacobi equation (Osher and Fedkiw, 2006). Conversely, the solution ϕ to a Hamilton-Jacobi equation is the integral of the solution u of a conservation law. This allows us to point out several important points. We know that the solutions of conservation laws may be discontinuous. For example, Burger's equation with a sinusoidal initial condition develops a discontinuity in the solution in finite time (Smoller, 1994). This implies that solutions of Hamilton-Jacobi equations can develop kinks (discontinuities in the first derivative) even though the initial data is smooth. As solutions of conservation laws may not be unique, we need to impose some entropy condition to select the 'physically relevant' solution amongst the set of solutions of the Hamilton-Jacobi equation.

The numerical solution technique used to solve equation (2.7) should consider the possibility of formation of singularities and kinks, and yield the physically correct non-smooth solution. Thus, the numerical solution to the level set equation (2.7) is based on the viscosity solutions to the related Hamilton-Jacobi equation (Sethian, 1998; Crandall et al., 1984). The idea of viscosity solutions was first introduced by Crandall and Lions (1983), wherein an additional term proportional to the second derivative of the function, with a numerical viscosity ν is added to the equation. This is analogous to the fluid viscosity (or diffusion), which also acts on the second derivative of solution variable. For example, equation (2.10) becomes:

$$\frac{\partial \phi}{\partial t} + H(\mathbf{x}, t, \phi, \nabla \phi) = \nu \nabla^2 \phi \tag{2.11}$$

The role of this additional term is to introduce some diffusion in the problem, thus smoothing out the generated shocks and discontinuities. The solution to equation (2.11) is then computed under the limit $\nu \to 0^+$, yielding the viscosity solution of equation (2.10). It can be proven that the viscosity solution of a Hamilton-Jacobi equation is unique (Crandall and Lions, 1983; Souganidis, 1985).

We now discuss some mathematical background of viscosity solutions to unsteady Hamilton-Jacobi equations. Consider the unsteady Hamilton-Jacobi equation (2.10) on a domain $\Omega \subseteq \mathbb{R}^n$, where $\phi : \Omega \times \mathbb{R}^+ \to \mathbb{R}$.

ϕ^- is a viscosity sub-solution of equation (2.10) if, for all $\mathcal{C}^1$ functions $\psi(\mathbf{x}, t)$, such that $\phi^- - \psi$ has a local maximum at $(\mathbf{x}, t)$, we have:

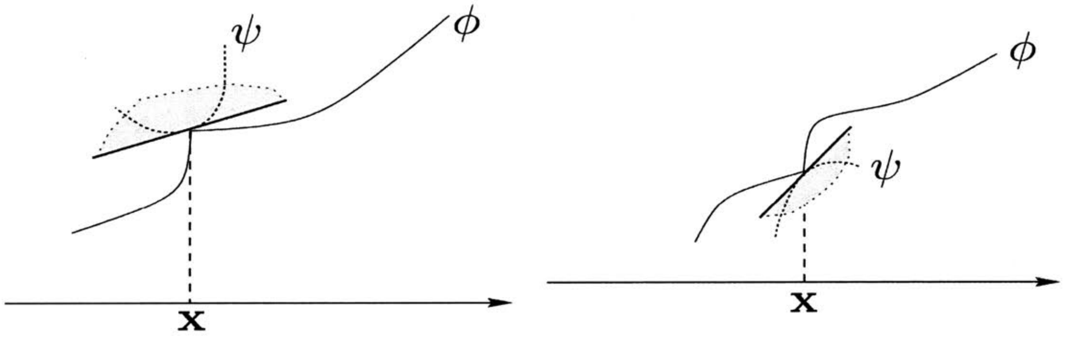
$$\frac{\partial \psi}{\partial t} + H(\mathbf{x}, t, \psi, \nabla \psi) \leq 0 \tag{2.12}$$

Similarly, ϕ^+ is a viscosity super-solution of equation (2.10) if, for all $\mathcal{C}^1$ functions

$\psi(\mathbf{x}, t)$, such that $\phi^+ - \psi$ has a local minimum at $(\mathbf{x}, t)$, we have:

$$\frac{\partial \psi}{\partial t} + H(\mathbf{x}, t, \psi, \nabla \psi) \geq 0 \tag{2.13}$$

In simple words, the viscosity super-solution (ϕ^+) utilizes a hyper-surface that is tangent from above to the graph of ϕ at $\mathbf{x}$ as an approximation for $\nabla\phi$ (which may not exist at $\mathbf{x}$). This is referred to as 'super-differential'. Similarly, the viscosity sub-solution (ϕ^+) approximates $\nabla\phi$ by a hyper-surface tangent to ϕ from below at $\mathbf{x}$, which is called 'sub-differential'. Figure 2-3 depicts super and sub-differentials for a general graph.



(a) Super-differential

(b) Sub-differential

Figure 2-3: Super-differentials and sub-differentials (modified from Bressan (2011). $\nabla\phi$ is approximated by the super-differential $\nabla\phi^+$ (shaded gray plane in the first image) for viscosity-super solution and by the sub-differential $\nabla\phi^-$ (shaded gray plane in the second image) for viscosity sub-solution. Graphs of the corresponding ψ functions are shown for the convenience of understanding.

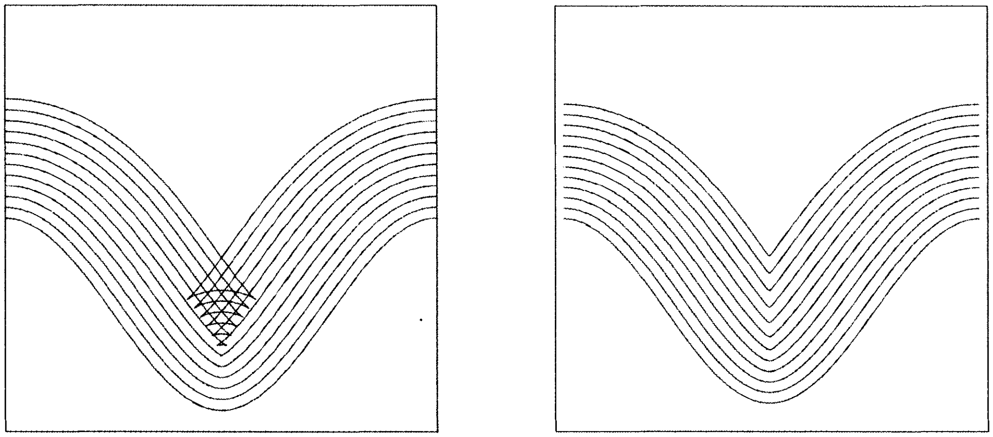
A function ϕ is the viscosity solution of equation (2.10) if it is both a viscosity super-solution and a viscosity sub-solution. Note that nowhere in the definition is ϕ ever differentiated. Only the test functions ψ (which are C^1 continuous) are differentiated. We refer to Bressan (2011) and Crandall and Lions (1983) for a detailed treatment of viscosity solutions of Hamilton-Jacobi equations. We also state some important points regarding viscosity solutions:

- If ϕ is a smooth (differentiable) solution of equation (2.10), then it is also a viscosity solution

- If the viscosity solution ϕ is differentiable at a particular point $\mathbf{x}$ (at any time t), then it exactly satisfies equation (2.10) at $\mathbf{x}$
- Given the initial conditions, the viscosity solution is unique
- It can be shown that viscosity solution is indeed the solution obtained from the vanishing viscosity method (as described above, with equation (2.11). Hence, viscosity solutions are the unique solutions obtained from smoothed Hamilton-Jacobi equations in the limiting condition

In order to numerically solve for viscosity solution of an unsteady Hamilton-Jacobi equation, entropy conditions are used. We use the entropy condition that was given by Sethian (1994) to pick out the unique viscosity solution. Simply stated, the entropy condition is as follows: "If the front is viewed as a burning flame, then once a particle is burnt, it stays burnt". careful adherence to this condition yields physically correct viscosity solutions. This condition simply means that the zero level set can pass through a point at most once, and that once it has passed through a particular point, the value of the level set function assumes some pre-fixed value to denote that the point has been "burnt". Mathematically, this means that at any point, information is reached through the characteristic that reaches this point for the first time and all later characteristic lines that arrive at this point are truncated. This allows us to design numerical schemes consistent with the viscosity solution of the Hamilton-Jacobi equation. A simple way to implement these schemes is to set the value of ϕ in the interior of the front $\partial\Omega^-$ to be some constant negative value (as points inside the front have already been touched by the front). Once any point assumes this value, it can never change. This way a point can never be touched by the front twice (or more), and that prohibits the front from creating twist and self-loops. This in turn yields the correct viscosity solution of the Hamilton-Jacobi equation. Chapter 3 discusses high order numerical schemes for Hamilton-Jacobi equations that hinge upon this viscosity solution. A front can revisit the positions it has visited before if it is exclusively advected (F is zero, but the external field $\mathbf{V}$ is non-zero). That is, the entropy condition is dormant for advecting transport. For propagating fronts, that is fronts moving due to their own speed, the entropy condition ensures that twists and loops are not allowed to form.

Figure 2-4 shows the evolution of a cosine curve normal to itself with constant speed for schemes with and without the entropy condition imposed. It is clear from this example that in the case of propagating fronts, even smooth initial conditions can cause sharp corners or kinks. First case does violates the entropy condition, and hence the evolved front passes through itself to develop a tail-like region. This solution is called the 'swallowtail' solution. The second figure plots the evolution of the front when the entropy condition is satisfied. This ensures that the curve does not cross itself, although the sharp corner created propagates in time without diffusing. This is the correct viscosity solution of this problem. Finally, we refer the reader to Sethian (1999b) for a thorough discussion on convergence and stability of various numerical schemes used to solve the Hamilton-Jacobi and level set equations, and to Osher and Fedkiw (2001) for the applications of level set method to various other fields.



(a) Entropy violating solution

(b) Entropy satisfying solution

Figure 2-4: Entropy condition violating and entropy condition satisfying solutions (adapted from Sethian (1994).Both consider the evolution of a cosine curve, but the entropy violating scheme develops a tail-like region whereas the entropy satisfying scheme produces the physically correct solution

2.2 Problem Formulation

In what follows, we discuss the problem statement and introduce the reader to the notation used. We consider the motion of a vehicle in domain $\Omega \subseteq \mathbb{R}^3$ that experiences a dynamic velocity field $\mathbf{V}(\mathbf{x}, t)$. The *maximum* speed that the vehicle can travel at is denoted by $F(\hat{\mathbf{h}}, \mathbf{x}, t)$ which is a given function of the heading direction $(\hat{\mathbf{h}})$, spatial location $(\mathbf{x})$, and the present time (t). The unit heading of the vehicle (relative to the flow field) is $\hat{\mathbf{h}}$, represented as an ordered pair (θ, φ), where θ and φ are the polar and azimuthal angles respectively. Note that we only have 2 degrees of freedom in the heading direction, as the third degree of freedom (in the radial direction) is accounted for through the speed of the vehicle $(F(\hat{\mathbf{h}}, \mathbf{x}, t))$, that is there are no implicit assumptions to start off with. We consider 3 predominant cases:

- The vehicle can freely move at any heading and the maximum vehicle speed F is independent of the heading $\hat{\mathbf{h}}$; that is, $F(\hat{\mathbf{h}}, \mathbf{x}, t) = F(t)$ only.

- The maximum vehicle speed depends on the direction in which the vehicle is headed, and / or the vehicle can only move in certain directions; that is, $F(\hat{\mathbf{h}}, \mathbf{x}, t) = F(\hat{\mathbf{h}}, t)$.

- Even though the vehicle moves in $\mathbb{R}^3$, the control over vehicle heading is only in one dimension (typically only over θ). This is relevant to many real situations where the vehicle performs a fixed vertical motion (deterministic φ), but can be steered freely in the horizontal (variable θ). Note that although we demonstrate this methodology for known φ, it can readily be extended to the cases where the vehicle motion is known along some parametric curve in the $\theta - \varphi$ space. For this case we have, $F(\hat{\mathbf{h}}(\theta, \varphi), \mathbf{x}, t) = F(\mathbf{x}, \hat{\mathbf{h}}(\theta), t)$.

We assume that the start point $(\mathbf{x}_s)$ and the destination $(\mathbf{x}_f)$ are known a priori. Let the trajectory of the vehicle be given by $\mathbf{X}(t)$. The vehicle moves due to its own speed F_v $(0 \leq F_v \leq F(\hat{\mathbf{h}}, t))$ in the direction specified by the heading $\hat{\mathbf{h}}(t)$, and is also advected by the present velocity field $\mathbf{V}(\mathbf{X}(t), t)$. Hence, the effective velocity experienced by the vehicle is given by equation (2.14). Using this, the evolution equation for the vehicle trajectory can be written as equation (2.15). Figure 2-5 shows the components affecting the motion of the vehicle as it moves from the start point to the target along some trajectory.

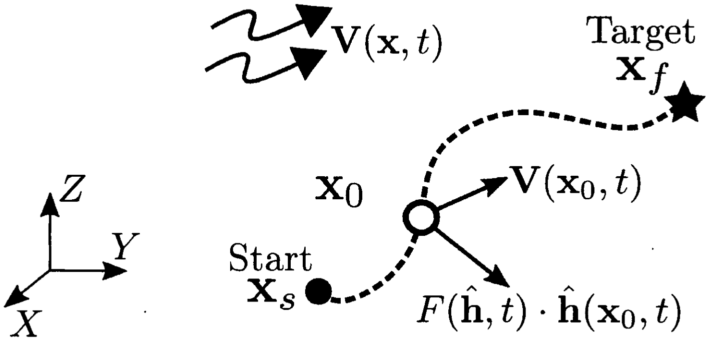

$\mathbf{V}(\mathbf{x}, t)$
Target
$\mathbf{x}_f$
Z
Y
X
$\mathbf{x}_0$
$\mathbf{V}(\mathbf{x}_0, t)$
Start
$\mathbf{x}_s$
$F(\hat{\mathbf{h}}, t) \cdot \hat{\mathbf{h}}(\mathbf{x}_0, t)$

Figure 2-5: Depiction of the problem statement. The dashed line represents a trajectory between the start point $\mathbf{x}_s$ and the target $\mathbf{x}_f$. The vehicle (at position $\mathbf{x}_0$ at some time t) experiences the velocity $\mathbf{V}(\mathbf{x}_0, t)$ and is steers itself along chosen heading $\hat{\mathbf{h}}(\mathbf{x}_0, t)$ with speed $F(\hat{\mathbf{h}}, t)$. The net motion of the vehicle is a vector sum of the observed velocity field and the steering movement of the vehicle.

$$\mathbf{V}_{eff}(\mathbf{X}(t), t) = (F_v \hat{\mathbf{h}} + \mathbf{V}(\mathbf{X}(t), t)) \tag{2.14}$$

$$\frac{d\mathbf{X}(t)}{dt} = \mathbf{V}_{eff}(\mathbf{X}(t), t) = F_v(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) + \mathbf{V}(\mathbf{X}(t), t) \tag{2.15}$$

Further, we denote the 'first arrival time' function by $\mathbf{T}(\mathbf{x} : \mathbf{x}_0, t_0)$, that is the time at which the vehicle reaches any specified point for the first time, given that it started from $\mathbf{x}_0$ at time t_0. Clearly, we have:

$$\mathbf{T}(\mathbf{x}_s : \mathbf{x}_s, 0) = 0 \quad \text{and} \quad \mathbf{X}(\mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0)) = \mathbf{x}_f \tag{2.16}$$

We wish to compute the vehicle heading and its speed as a function of time which will yield minimum travel time for the vehicle between $\mathbf{x}_s$ and $\mathbf{x}_f$, subject to the constraints imposed by equation (2.15) and equation (2.16). As will be discussed later, it can be shown that in order for the vehicle to reach the desired destination in the minimum amount of time, it must travel at the maximum allowable speed (Lolla and Lermusiaux, 2017; Lolla, 2016). Hence, the problem can be formulated as:

$$\hat{\mathbf{h}}(t)_{optimal} = \arg\min_{\hat{\mathbf{h}}} \mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0) \tag{2.17}$$

This optimization will yield the sequence of headings (as a function of time) at which the vehicle should be steered relative to the flow-field for minimum travel time between $\mathbf{x}_s$ and $\mathbf{x}_f$, and the value of this minimum travel time $\mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0)$. In some cases, it is also desirable to obtain way-points along the path to be followed by the vehicle. These way-points are close to each other, and the vehicle is steered in between the way-points through another set of control laws (such as some feedback-type control). In such cases, this framework is used to provide a 'high-level' description of the path to be followed. This sort of structure is particularly useful when information about the velocity at a sub-grid scale is not well-known. Such a sequence of way-points can also be readily generated from the obtained results.

Assumptions and Remarks:

- In this work, we assume that the external velocity field $\mathbf{V}(\mathbf{x}, t)$ is exactly known. Our main focus is on oceanic vehicles, where this external velocity field can be estimated well by various methods, but it still possesses some uncertainty. However, this work can be utilized to predict a sample path for a specific velocity realization or the path corresponding to the averaged velocity field or most probable velocity field in such cases. Further work has also been done by Subramani et al. (2017c); Wei (2015) based on the same framework to study time-optimal paths in stochastic environments. We also assume that the trajectories followed by the vehicle are a continuous function of time. This imposes a constraint that the velocity field experienced by the vehicle cannot be a Dirac delta function. Otherwise, then the trajectory of the vehicle will not be continuous. Note that, for the applications we consider, the velocity field is almost always smooth and slowly varying. Hence, this assumption is well-justified.

- We are primarily looking at oceanic applications. Hence it is assumed that the geometric dimensions of the vehicle are much smaller than the distance traveled by the vehicle, in order to obtain globally optimal long distance trajectories.

- We also assume that the vehicle behaves as a point mass and the interaction between the flow and the vehicle is purely kinematic. We assume that the vehicle speed (F) that we use is the effective vehicle speed; that is, the speed that the vehicle would be able to achieve after considering the effect of drag forces and inertia.

- In this work, we only seek time-optimality. In many cases, optimality in some other objective function, such as energy spent, quality of data collected etc. may be desired. The current framework can be extended inorder to optimize a general objective function that depends on the travel time, followed trajectory and possible other parameters. We refer the reader to Subramani and Lermusiaux (2016) for an example of energy optimization as well as Lolla (2016) for computing optimal sampling locations based on reachability.

2.3 Theoretical Results

In this section, we discuss the level set method based path planning methodology. We first discuss the concept of reachability and its link to Hamilton-Jacobi equations. The basis of the path planning methodology is a theorem first presented in Lolla et al. (2012), and further developed in Lolla and Lermusiaux (2017); Lolla (2016). We present the general form of this theorem and comment about its implications. As seen before, our approach to computing optimal paths in dynamic environments can be segregated into two parts: forward evolution (in time) of the reachability set and backward tracing (in time) of the optimal path. These two parts are discussed in depth in this section. Finally, we present the application of this theorem to the different marine vehicles as described before, namely: vehicles with isotropic speeds, vehicles with anisotropic speeds and vehicles with fixed vertical motion.

We now look at reachability sets. A reachability set (at a fixed time) is the set of *all the points* that can be visited by the vehicle until this fixed time, given the starting point of the vehicle. The boundary of this set is called the reachability front. This means that at any time, all he points that lie on the reachability front have been reached for the first time at the current time (value of the first arrival time function at these points is the current time itself). Thus, the reachability front contains all the present locations of the vehicle that lie on the possible realizable paths and are the farthest away from the start position. In general, the reachable set at time t, starting from $\mathbf{x}_s$, $\mathcal{R}(\mathbf{x}_s, t)$ is defined as:

$$\mathcal{R}(\mathbf{x}_s, t) = \left\{\mathbf{y}_{\mathbf{x}_s} \;\middle|\; \mathbf{y}_{\mathbf{x}_s} \in \mathcal{S}_{[0,\tau]}(\mathbf{x}_s),\ \ \tau \in [0, t]\right\} \tag{2.18}$$

Here, $\mathbf{y}_{\mathbf{x}_s}$ denotes a feasible trajectory (set of points) with starting point $\mathbf{x}_s$ and satisfy equation (2.15) (Falcone and Zidani, 2012). The set $\mathcal{S}_{[0,t]}$ is the set of all feasible trajectories (starting from $\mathbf{x}_s$ at time $t = 0$) till time t.

The fundamental idea behind the path planning methodology used is as follows: If we keep track of the reachability front at all times, we can determine when this front reaches the target. The time at which it crosses the target for the first time is the minimum travel time between the start point and the target. Further, the path(s) traced by the point(s) on the reachability front that reaches the target is / are exactly the optimal path(s) we wish to compute. Figure 2-6 pictorially depicts the reachability front, and the possible heading choices at two different positions.

Our approach is extremely efficient because at each time, it only considers headings that maximally 'expand' the reachability set and hence there is no redundancy. If this was not done, computing an optimal path in dynamic environments would not be easy. The difficulty arises because at any instant, a vehicle has an infinite heading choices to choose from. For each of these headings, there again exist infinite headings at the next instant, and so on. Hence, choosing an optimal sequence of heading

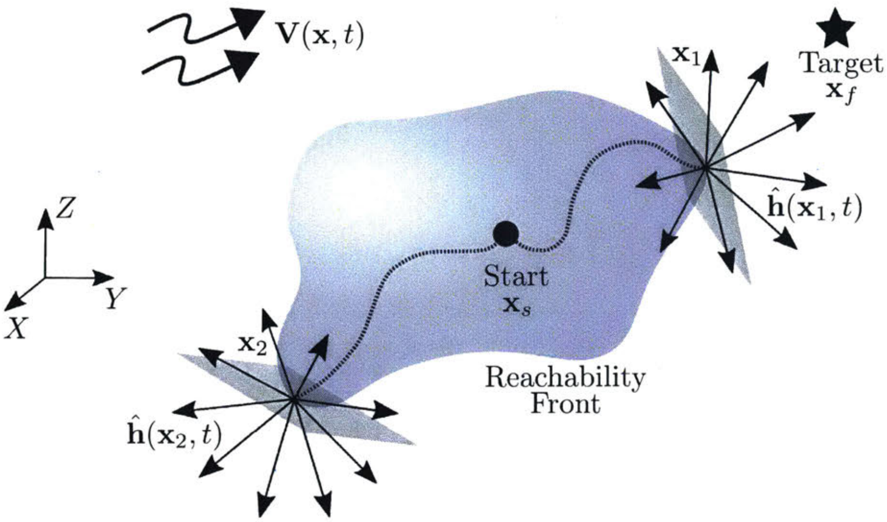

$\mathbf{V}(\mathbf{x},t)$
$\mathbf{x}_1$
Target
$\mathbf{x}_f$
$\hat{\mathbf{h}}(\mathbf{x}_1,t)$
Z
Y
X
Start
$\mathbf{x}_s$
$\mathbf{x}_2$
Reachability
Front
$\hat{\mathbf{h}}(\mathbf{x}_2,t)$

Figure 2-6: Evolution of the reachability front in 3 dimensions. 2 sample positions ($\mathbf{x}_1$ and $\mathbf{x}_2$) and the possible headings at these positions are shown (tangent planes to the reachability front at these positions shown for reference). The dashed lines show the trajectories followed by fictitious particles to reach $\mathbf{x}_1$ and $\mathbf{x}_2$.

would become very extremely challenging with such an approach. A simplification could be made by imposing heuristics as done in some other approximate methods (see chapter 1), but that would not guarantee global optimality. Another way would be to compute all possible paths, and choose the optimal. This is extremely expensive and cannot be applied beyond toy problems.

In what follows, we discuss the evolution of the reachability front and show that its dynamics are governed by a Hamilton-Jacobi equation. The level set method is then used to track the propagation of the reachability front. Finally, we discuss the extraction of the optimal path once the reachability front crosses the target.

In what follows, we discuss a theorem that forms the basis of our methodology and the discuss its applications to the different cases mentioned in section §(2.2).

As can be seen from equation (2.15), we have 2 free control parameters: namely the vehicle speed (F_v) and the vehicle heading ($\hat{\mathbf{h}}$). We would like to predict the optimal values of these control parameters (as a function of time) such that the first arrival time function for the target point $\mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, t_0)$ is minimized. The optimal parameter values will then be used to determine the optimal path. In this regard, we now look at the theorem that allows us to choose optimal values of these control parameters in order to minimize the first arrival time at the target point (given the start time and position).

Theorem. *Let $\boldsymbol{V}(\boldsymbol{x}, t)$ be a Lipschitz continuous velocity field in all its arguments. Assume that the set of permissible heading directions $\mathcal{H}$ is a subset of $\mathbb{R}^n$. Let the vehicle speed $F(\hat{\boldsymbol{h}}, t) : \mathcal{H} \times [0, \infty) \to \mathbb{R}^+$ be Lipschitz continuous in all of its arguments. Let $\boldsymbol{T}(\boldsymbol{x}_f : \boldsymbol{x}_s, 0)$ denote the optimal first arrival time at target $\boldsymbol{x}_f$, given that the vehicle started from start position $\boldsymbol{x}_s$ at time $t = 0$.*
We assume that the possible vehicle trajectories $\boldsymbol{X}(t)$ are governed by equation (2.15), with initial condition $\boldsymbol{X}(0) = \boldsymbol{x}_s$. Then, the evolution of the reachability front is given by the zero level set of the function $\phi : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}$, where $\phi(\boldsymbol{x}, t)$ is the unique viscosity solution of the following equation:

$$\frac{\partial \phi}{\partial t} + \max_{\hat{\boldsymbol{h}}} \left(F(\hat{\boldsymbol{h}}, t)\hat{\boldsymbol{h}}(t) \cdot \nabla \phi + \boldsymbol{V}(\boldsymbol{x}, t) \cdot \nabla \phi \right) = 0 \tag{2.19}$$

For the initial conditions:

$$\phi(\boldsymbol{x}, 0) = |\boldsymbol{x} - \boldsymbol{x}_s| \tag{2.20}$$

That is,

1. *The optimal arrival time $\boldsymbol{T}(\boldsymbol{x}_f : \boldsymbol{x}_s, 0)$ satisfies*

$$\boldsymbol{T}(\boldsymbol{x}_f : \boldsymbol{x}_s, 0) = \inf_{t \geq 0} \{t \mid \phi(\boldsymbol{x}_f, t) = 0\} \tag{2.21}$$

2. *the optimal trajectory (or trajectories) $\boldsymbol{X}(t)$ are given by the characteristic lines of equation (2.19). That is, $\boldsymbol{X}(t)$ satisfies the following equation:*

$$\frac{d\boldsymbol{X}}{dt} = F(\hat{\boldsymbol{h}}, t)\hat{\boldsymbol{h}}(t) + \boldsymbol{V}(\boldsymbol{X}, t) \tag{2.22}$$

Where

$$\hat{h}(t) = \arg\max_{\hat{h}} \left(F(\hat{h}, t)\hat{h}(t) \cdot \nabla\phi \right) \tag{2.23}$$

□

The proof of this theorem for the case when ϕ is differentiable is discussed in appendix A. For a more general version of the proof, we refer the reader to Lolla and Lermusiaux (2017); Lolla (2016).

Figure 2-7 describes the evolution of the reachability front through the motion of an arbitrary fictitious vehicle that resides on the reachability front at time t. The optimal heading is chosen such that $F(\hat{\mathbf{h}}, t) \cdot \hat{\mathbf{h}}(\mathbf{x}_0(t), t)$ has the maximum projection along $\nabla\phi$. If this choice of optimal heading is not made, then the vehicle will end up inside the reachability front at later times, and would not reach the destination in optimal time. Along with this motion, the vehicle is also advected due to the external velocity field $\mathbf{V}(\mathbf{x}_0, t)$. We track the optimal motion of all such fictitious vehicles that lie on the reachability front implicitly through the equation (2.19), which is equivalent to tracking the propagation of the reachability front (and the reachability set).

It should be mentioned that this formulation does not guarantee the uniqueness of the optimal path. Theoretically, all the possible optimal paths can be predicted by this methodology. The only constraint that we impose is to obtain path(s) that take the minimum possible time to reach the destination. As described before, while numerically computing the optimal paths, we seek viscosity solutions of equation (2.19). That is, we permit local non-differentiability. This in turn means that we allow for multiple characteristics to merge and propagate as kinks. As optimal paths are nothing but characteristics of equation (2.19), we have multiple optimal heading choices whenever we encounter a kink while backtracking the path. Each of these choices leads to a different optimal path. The important point to note is that even though the optimal path may not be unique, all such paths take the exact same time to reach the destination, which indeed is the minimum possible travel time between the selected start and target positions.

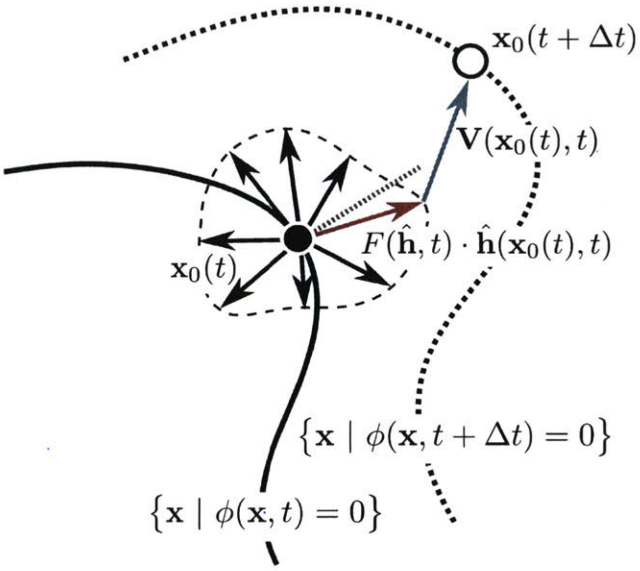

$\mathbf{x}_0(t+\Delta t)$
$\mathbf{V}(\mathbf{x}_0(t),t)$
$F(\hat{\mathbf{h}},t)\cdot\hat{\mathbf{h}}(\mathbf{x}_0(t),t)$
$\mathbf{x}_0(t)$
$\{\mathbf{x} \mid \phi(\mathbf{x},t+\Delta t)=0\}$
$\{\mathbf{x} \mid \phi(\mathbf{x},t)=0\}$

Figure 2-7: Motion of a fictitious vehicle on the reachability front. The black arrows indicate possible steering choices for the vehicle (assuming anisotropic speed). The red arrow is the preferred (optimal) steering direction, as it has the maximum projection along $\nabla\phi$ (dotted gray line). Apart from its own motion, the vehicle is also advected due to the external velocity (cyan arrow) and moves from $\mathbf{x}_0(t)$ to $\mathbf{x}_0(t+\Delta t)$ in one time step. Implicitly tracking the optimal motion of all such vehicles on the current reachability front yields the new evolved reachability front (black dotted curve).

2.4 Forward Evolution of Reachability Set

Section §(2.3) suggests us to look at the path planning problem in two sub-parts: evolution of the reachability set (and reachability front) and back tracing of the optimal path. In this section, we look at the former. We first consider vehicles with isotropic speed. That is, vehicles that can freely move in any direction. Then we look at vehicles whose speed varies with the direction and / or can move only in certain directions. Vehicles such as oceanic floats fall in this category. specific treatment of these types of vehicles is also discussed in chapter 3. Finally, we consider the case of vehicles with fixed vertical motion that can only be steered in the horizontal, and show that such a constraint on the motion of the vehicle reduces the dimensionality of the problem.

2.4.1 Vehicles with Isotropic Speed

In this case, we assume that the vehicle is allowed to travel in all possible directions ($\mathcal{H} = \mathbb{R}^3$) and also that the speed of the vehicle does not depend on the direction of its travel. That means:

$$F(\hat{\mathbf{h}}, t) = F(t) \tag{2.24}$$

Consider equation (2.19) which governs the evolution of the level set function ϕ. Substituting $F(\hat{\mathbf{h}}, t) = F(t)$, we get:

$$\frac{\partial \phi}{\partial t} + \max_{\hat{\mathbf{h}}} \left(F(t)\hat{\mathbf{h}}(t) \cdot \nabla\phi + \mathbf{V}(\mathbf{x}, t) \cdot \nabla\phi \right) = 0 \tag{2.25}$$

$$\tag{2.26}$$

That is,

$$\frac{\partial \phi}{\partial t} + F(t) \max_{\hat{\mathbf{h}}} \left(\hat{\mathbf{h}}(t) \cdot \nabla\phi \right) + \mathbf{V}(\mathbf{x}, t) \cdot \nabla\phi = 0 \tag{2.27}$$

$$\tag{2.28}$$

Now $\hat{\mathbf{h}}(t) \cdot \nabla\phi$ will be maximum only when $\hat{\mathbf{h}}(t) = k\nabla\phi$ for some $k \in \mathbb{R}$. Note that $\hat{\mathbf{h}}(t)$ is also constrained to be a unit vector. This means that $k|\nabla\phi| = 1$. Hence, $k = \frac{1}{|\nabla\phi|}$. Substituting this in equation (2.27), we get:

$$\frac{\partial \phi}{\partial t} + F(t)|\nabla\phi| + \mathbf{V}(\mathbf{x}, t) \cdot \nabla\phi = 0 \tag{2.29}$$

With the optimal headings given by: $\hat{\mathbf{h}}(\mathbf{x}, t) = \frac{\nabla\phi(\mathbf{x},t)}{|\nabla\phi(\mathbf{x},t)|}$.
We refer to this equation as the modified Hamilton-Jacobi level set equation (Lolla et al., 2012; Lolla, 2016), subject to the initial condition given by equation (2.20).

In this equation, the second term on the left hand side refers to contribution due to the vehicle motion. We refer to this as the 'optimal propulsion term'. The last term on the left hand side is referred to as the 'advection term'.

Equation (2.29) is solved forward in time until the zero level set reaches the destination. That is, until $\phi(\mathbf{x}_f, T) = 0$. The time T is the optimal arrival time.

2.4.2 Vehicles with Anisotropic Speed

We now focus on vehicles that have speed dependent on the direction they travel in. This type of vehicles are ubiquitous in oceanic scenarios, and need a more general treatment vehicles with direction-independent speeds. Some examples of such vehicles are marine floats that can only travel in the vertical direction, or sea-gliders that can travel only along certain polar angles. Speeds of various ships and sailboats depend on the direction of the wind, and hence they travel at different speeds in different directions (Lolla and Lermusiaux, 2017; Hessels, 2014).This framework though, is not limited to any specific type of motion and can be used with any restriction on vehicle motion, given that this restriction is known a priori.

In such cases, we need to solve the general form of the modified Hamilton-Jacobi level set equation, that is equation (2.19). As the external velocity field is not a function of the vehicle heading, it can be taken out of the maximization to yield:

$$\frac{\partial \phi}{\partial t} + \max_{\hat{\mathbf{h}}} \left(F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) \cdot \nabla \phi \right) + \mathbf{V}(\mathbf{x}, t) \cdot \nabla \phi = 0 \tag{2.30}$$

In this case, the optimal propulsion term is given by $\max_{\hat{\mathbf{h}}} \left(F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) \cdot \nabla \phi \right)$ and the advection term is still $\mathbf{V}(\mathbf{x}, t) \cdot \nabla \phi$.

Similar to the last case, equation (2.30) is solved until the zero level set crosses the target for the first time, subject to the initial condition given by equation (2.20). Note that, in this case the maximization needs to be performed at each time, and at each spatial location in order to obtain the optimal headings (Lolla and Lermusiaux, 2017; Hessels, 2014). There are several ways to compute this maximum, which are discussed in depth in chapter 3.

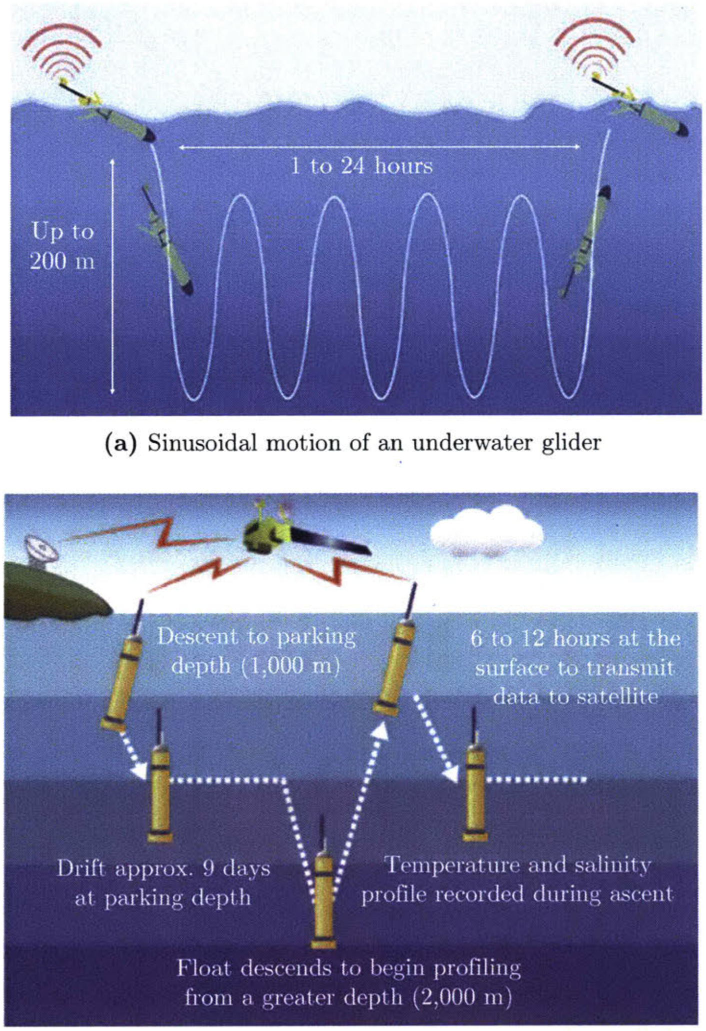
2.4.3 Vehicles with deterministic Constrained Motion

We now present novel path planning results for vehicles whose motion is deterministic along a parametric constraint. The theory presented in the following work is general and may be applied for any parametric constraints on the motion of the vehicle. However, we look at the vehicles whose vertical motion is pre-determined. Many oceanic vehicles travel vertically by adjusting their buoyancy, and in such cases this vertical motion is fixed which typically is a sinusoid, such as a glider performing a yo - yo motion to sample data (Rudnick et al., 2004; Testor et al., 2010; Javaid et al., 2014). Oceanic floats also sometimes are constrained to fixed vertical motion, with one cycle lasting up to 10 days (Roemmich et al., 2009; Kobayashi et al., 2012). Figure 2-8 schematically represents the vertical motion of gliders and floats. For example, for some vehicles, this can imply that the control over the heading direction is only in the horizontal plane. In general, one direct way to plan paths for such cases is to use the framework for vehicles with anisotropic speeds, and to only consider the specific heading directions that are permitted.

The aspect of exactly knowing the vertical motion of the vehicle offers great advantage in terms of computational expense. Specifically, a problem in 3 dimensions with known motion in 1 direction can be reduced to a 2 dimensional problem. This reduces the computational expense significantly.

We now look at a vehicle with known vertical motion and complete freedom over the heading choice in the horizontal plane. We also assume that the horizontal speed of the vehicle is independent of the direction. Although these assumptions simplify the equations, this theory is applicable to cases with extra restrictions on the horizontal movement as well.

Let us assume that the vertical velocity of the vehicle is known, and denoted by U_z. The external velocity field is given by $\mathbf{V} = (V_x, V_y, V_z)$. The vertical position of the vehicle is given by: $z = \int_0^t U_z dt$. The unit vectors in the X and Y directions are $\hat{e}_x$ and $\hat{e}_y$ respectively. Let us look at equation (2.30). Note that, in this case the optimization of the heading $\hat{\mathbf{h}}$ is only performed with respect to θ, as the motion in


1 to 24 hours
Up to 200 m
Descent to parking depth (1,000 m)
6 to 12 hours at the surface to transmit data to satellite
Drift approx. 9 days at parking depth
Temperature and salinity profile recorded during ascent
Float descends to begin profiling from a greater depth (2,000 m)

(a) Sinusoidal motion of an underwater glider

(b) Motion of a profiling float

Figure 2-8: Motion of an oceanic glider and float, modified from Lust and Stevens (2015) and Kobayashi et al. (2012) respectively. The glider performs a sinusoidal motion, diving to a fixed depth. The float on the other hand dives to a particular depth and travels at this fixed depth for a pre-decided duration. It then dives deeper, and collects data while rising from this depth. Ultimately it travels at the ocean surface to transmit data.

φ is known. Expanding the variables for 3 dimensions, we get:

$$\frac{\partial \phi}{\partial t} + \max_{\hat{\mathbf{h}}(\theta)} \left(\Big(F(\hat{\mathbf{h}}(\theta), t)\hat{\mathbf{h}}(\theta) \cdot \hat{e}_x, F(\hat{\mathbf{h}}(\theta), t)\hat{\mathbf{h}}(\theta) \cdot \hat{e}_y, U_z(t) \Big) \cdot \nabla \phi \right) + \mathbf{V} \cdot \nabla \phi = 0 \tag{2.31}$$

Let us denote $F(\hat{\mathbf{h}}(\theta), t)\hat{\mathbf{h}}(\theta) \cdot \hat{e}_x$ by $F_x(\hat{\mathbf{h}}(\theta), t)$ and $F(\hat{\mathbf{h}}(\theta), t)\hat{\mathbf{h}}(\theta) \cdot \hat{e}_y$ by $F_y(\hat{\mathbf{h}}(\theta), t)$.

Hence equation (2.31) becomes:

$$\begin{aligned}
&\frac{\partial \phi}{\partial t} + \max_{\hat{\mathbf{h}}(\theta)} \left(F_x(\hat{\mathbf{h}}(\theta), t) \cdot \phi_x + F_y(\hat{\mathbf{h}}(\theta), t) \cdot \phi_y + U_z(t) \cdot \phi_z \right) + \mathbf{V} \cdot \nabla \phi = 0 \\
&\frac{\partial \phi}{\partial t} + \max_{\hat{\mathbf{h}}(\theta)} \left(F_x(\hat{\mathbf{h}}(\theta), t) \cdot \phi_x + F_y(\hat{\mathbf{h}}(\theta), t) \cdot \phi_y + U_z(t) \cdot \phi_z \right) + \mathbf{V} \cdot \nabla \phi = 0 \\
&\frac{\partial \phi}{\partial t} + \max_{\hat{\mathbf{h}}(\theta)} \left(F_x(\hat{\mathbf{h}}(\theta), t) \cdot \phi_x + F_y(\hat{\mathbf{h}}(\theta), t) \cdot \phi_y \right) + \mathbf{V}_{mod} \cdot \nabla \phi = 0 \qquad (2.32)
\end{aligned}$$

where, $\phi_i = \frac{\partial \phi}{\partial i}$, and $\mathbf{V}_{mod} = (V_x(\mathbf{x}, t), V_y(\mathbf{x}, t), V_z(\mathbf{x}, t) + U_z(t))$. In order to solve efficiently, we split this equation into two parts:

$$\left(\frac{\partial \phi}{\partial t} \right)_1 + \max_{F_x(t), F_y(t)} ((F_x(t), F_y(t)) \cdot (\phi_x, \phi_y)) + (V_x(\mathbf{x}, t), V_y(\mathbf{x}, t)) \cdot (\phi_x, \phi_y) = 0 \qquad (2.33)$$

$$\left(\frac{\partial \phi}{\partial t} \right)_2 + (V_z(\mathbf{x}, t) + U_z(t)) \cdot \phi_z = 0 \qquad (2.34)$$

$$\frac{\partial \phi}{\partial t} = \left(\frac{\partial \phi}{\partial t} \right)_1 + \left(\frac{\partial \phi}{\partial t} \right)_2 \qquad (2.35)$$

Note that equation (2.33) is simply anisotropic path planning in 2 dimensions. Further, for the case of heading independent speeds, it can be reduced similarly to how equation (2.29) was reduced, to yield equation (2.36)

$$\frac{\partial \phi}{\partial t} + F_{2D} |\nabla \phi_{2D}| + \mathbf{V}_{2D}(\mathbf{x}, t) \cdot \nabla \phi_{2D} = 0 \qquad (2.36)$$

where the subscript $2D$ indicates that only the horizontal components are considered. At each time step, we first solve equation (2.36), after which we add the contribution due to equation (2.34).

In real oceanic scenarios, the vertical velocity of the flows is often very small compared to the vehicle speed, that is: $V_z << U_z$ (Pedlosky, 2013). In such cases if we neglect the flow vertical velocity, equation (2.34) reduces to:

$$\frac{\partial \phi}{\partial t} + V_z(\mathbf{x}, t) \cdot \phi_z = 0 \tag{2.37}$$

Computational Remarks

1. *Time-optimal paths with fixed vertical motions in negligible vertical flows.* It can be seen that equation (2.36) corresponds to the 'movement' of the zero level set in the horizontal plane, and equation (2.34) corresponds to its motion in the vertical direction. By splitting the original equation (2.30) into two, we have effectively decoupled the motion in the vertical direction and motion in the horizontal plane. Now, as mentioned above, if we ignore V_z, then the second equation simply states that the zero level set is propagated vertically at the speed of the vehicle. This means that if we started with a planar reachability set, then it will always remain planar, and will be transported vertically as a whole. Note that it will still undergo deformations in the horizontal plane, but none in the vertical. This makes solving the system consisting of equation (2.36) and equation (2.37) even easier as effectively we only need to solve for equation (2.36), knowing that the level zero level set (and hence the vehicle) can only exist at a depth of $z_t = \int_0^t U_z(t)dt$. Note that while solving equation (2.36), one needs to sample the correct velocity field. Specifically, while solving at time t, the velocity field that needs to be considered is given by:

   $$\mathbf{V}_{2D} = (V_x(x, y, z_t), V_y(x, y, z_t)) \tag{2.38}$$

   That is, solving equation (2.30) with the vertical motion constraints is equivalent to solving equation (2.36), but with correctly sampled X and Y velocity field at a depth z_t, as mentioned above.

   This idea can also be

2. *Time-optimal paths with fixed vertical motions in general flows.* The above directly extended to the case where the vertical velocity is not ignored. For

such instances, we first solve the 2D path planning problem similar to the earlier one, but with a modified velocity field as explained ahead. Although, as we are not ignoring the vertical flow velocities, now our zero level set will not remain planar at all times (even though we started with a planar zero level set). To this end, we construct a new 'hybrid' 2D velocity field. This field is constructed by setting the velocity at each point on the zero level set to be the velocity that the location of this corresponding point experiences. This means that such a velocity field is not a real velocity field at any depth, but a manually 'stitched' field which could be used for solving equation (2.36). Once computation of the optimal heading and evolution of the zero level set is complete for the particular time, we again find the actual positions of the points on the zero level set (in 3 dimensions) by adding the vertical velocities experienced by these points at the corresponding locations. This in turn is analogous to solving equation (2.34) in a Lagrangian way. The same procedure is repeated for each time step thereafter.

We now summarize the above results by an algorithm consisting of a few key steps:

1. Obtain the positions of the points that lie on the zero level set at current time. Construct a hybrid velocity field in 2D that uses the velocities experienced by these points (at their current positions).

2. Use this velocity field to solve equation (2.36) to obtain optimal headings and positions of the zero level set points in 2D.

3. Obtain the final positions of the zero level set points by using the positions of the zero level set points in 2D and then compute their correct depth by adding the displacement due to the local total vertical velocity.

Interactions of the zero level set with obstacles and / or bathymetry can also be handled in a similar way. The points on the zero level set that try to pass through these forbidden regions are forced not to move by also setting the vehicle velocity at these points to be equal to zero. For vehicles such as gliders that perform sinusoidal (yo-yo) motion, the direction of the vehicle velocity is inverted once they reach the bottom bathymetry. This can again be easily performed as we individually keep track of the vertical motion of zero level set points.

2.5 Backward Tracing of the Optimal Path

As described in section §(2.3), the second part of the path planning problem involves tracing the optimal path, given that the zero level set has crossed the destination. We now look at the method to obtain this optimal path. The optimal path is the trajectory followed by a particle that always travels on the zero level set and reaches the destination in the shortest amount of time, as seen in section §(2.3). This is a Lagrangian way of looking at the reachability front. This in turn means that the optimal path is a characteristic of equation (2.19) that reaches the target $\mathbf{x}_f$ at time $\mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0)$, given by equation (2.21). Hence, in order to obtain the complete path followed by the vehicle, we need to trace the path backwards, starting from the target position ($\mathbf{x}_f$) and time $t = \mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0)$, and going to the start position $\mathbf{x}_s$, and time $t = 0$. We refer to this as backward tracing or backtracking of the optimal path.

While solving the forward evolution equation of the reachability set (equation (2.29)), we also need to compute the optimal heading direction for each point on the reachability front at that particular time. While solving the backtracking problem, as we are looking for a characteristic of equation (2.19), the partial differential equation (PDE) manifests itself as an ordinary differential equation (ODE) along this path. That is, we only need to know the optimal heading only at the present location of the vehicle (at the corresponding time) in order to backtrack the path. This information is readily available, and no extra computation is required.

It is necessary to mention here that as we are solving for a characteristic curve of a PDE, consistency needs to be maintained between the temporal schemes for the PDE and the ODE. If the temporal schemes are different, then the two are not directly numerically compatible, and may cause significant errors. New results on this topic are included in chapter 3.

We now look at the differential equation followed by the optimal path (trajectory).

This equation is the same as equation (2.22), but with additional constraints.

$$\frac{d\mathbf{X}}{dt} = F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) + \mathbf{V}(\mathbf{X}(t), t) \tag{2.39}$$

where

$$\hat{\mathbf{h}}(t) = \arg\max_{\hat{\mathbf{h}}} \left(F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) \cdot \nabla\phi(\mathbf{X}, t) \right) \tag{2.40}$$

and

$$\mathbf{X}(\mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0)) = \mathbf{x}_f \tag{2.41}$$

We recall that the optimal travel time $\mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0)$ is defined by equation (2.21).Hence, to obtain the obtain headings, we first solve equation (2.39) backward in time, for which equation (2.41) is used as an initial condition. The optimal headings at each time are given by equation (2.40), following the vehicle in a Lagrangian sense.

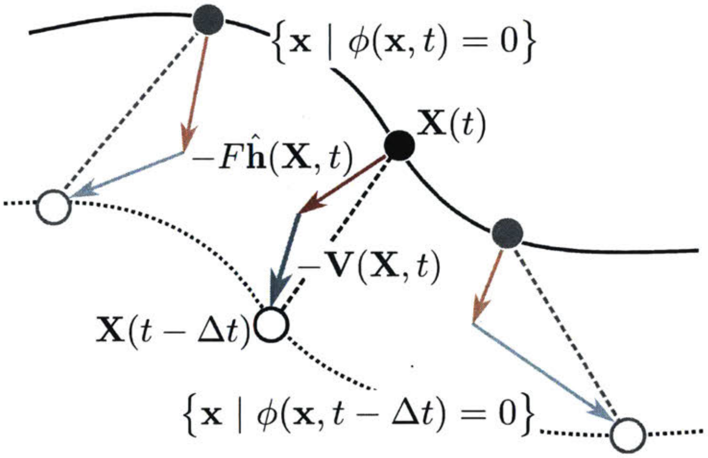

$\{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0\}$
$\mathbf{X}(t)$
$-F\hat{\mathbf{h}}(\mathbf{X}, t)$
$-\mathbf{V}(\mathbf{X}, t)$
$\mathbf{X}(t - \Delta t)$
$\{\mathbf{x} \mid \phi(\mathbf{x}, t - \Delta t) = 0\}$

Figure 2-9: Backtracking the optimal path. The optimal path is given by the trajectory of the point that always moved on the reachability front to reach the destination $\mathbf{x}_f$ at time $\mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0)$. In order to compute the optimal trajectory, we start from $\mathbf{x}_f$ and march backward in time by taking into account the motion due to vehicle steering (red arrow) and advection due to the external velocity field (cyan arrow).

For vehicles with isotropic speeds, equation (2.39) is reduced to equation (2.42), where the optimal heading $\hat{\mathbf{h}}(t)$ has been replaced by its expression $\hat{\mathbf{h}}(t) = \frac{\nabla\phi(\mathbf{X}(t),t)}{|\nabla\phi(\mathbf{X}(t),t)|}$

since in that case the optimal $\hat{\mathbf{h}}(t)$ is exactly the normal to the level set. This equation is also subject to the initial condition given by equation (2.41).

$$\frac{d\mathbf{X}}{dt} = F\frac{\nabla\phi(\mathbf{X},t)}{|\nabla\phi(\mathbf{X},t)|} + \mathbf{V}(\mathbf{X}(t),t) \tag{2.42}$$

For the case of path planning for vehicles with known vertical motion, as described before, optimal headings provide us the travel direction in the horizontal plane. As we know the vertical motion of the vehicle, a vector sum of the horizontal and the vertical travel directions yields the optimal heading direction for such vehicles.

Chapter 3

Numerical Challenges, Schemes and Implementation

We reviewed he theoretical foundation behind our path planning methodology and extended it to vehicles with anisotropic and constrained motions in the previous chapter. Another equally important aspect is its applicability in realistic conditions and the accuracy of the results. Specifically, we describe the partial differential equation governing the evolution of the reachability set (equation (2.19)) and the ordinary differential equation governing the corresponding time-optimal path (equation (2.22). In this chapter, novel approaches towards the accurate, consistent and efficient numerical implementation of these equations are introduced.

To compute and predict accurate reachability sets and time-optimal paths in realistic 3D conditions, several critical computational, numerical, and implementation questions arise. They include: (i) how to obtain accurate solutions of level-set contours in dynamic fields with sharply skewed and anisotropic length and temporal scales; (ii) how to solve for the reachability front evolution with highly accurate numerical schemes that minimize numerical dissipation and other errors; (iii) numerically, what are the best approaches to handle constrained and anisotropic vehicle motions in 3 dimensions (iv) how to ensure that the forward-backward numerical consistency is maintained in all the computations; (v) what are some approaches to efficiently re-construct ϕ to be a signed distance function, as the level set function is distorted and the signed distance property is lost during advection (vi) what are the efficient implementation techniques for execution in complex domains. The results presented in this section provide solutions to each of these key questions.

First, we present the various challenges that arise while applying our methodology to realistic 3D scenarios followed by the solutions to surmount them. We then discuss the non-dimensional form of the Hamilton-Jacobi level set equation, which is crucial in solving the path planning problem for real ocean domains. After this, we explain the various high order spatial and temporal numerical schemes used to solve for the evolution of the zero level set function. We then look at the methods to perform maximization of the Hamiltonian from equation (2.19), and mention the relative merits of various approaches. This is followed by the discussion about reinitialization of the signed distance function, which is necessary to maintain regularity of the level set function ϕ. Finally, we go over the schemes for backtracking the optimal path, and the necessity of consistency between temporal schemes for forward evolution and backward tracing.

3.1 Challenges in Implementation for 3D Realistic Ocean Flows

3.1.1 Challenges in Forward Evolution

In this section, we take a look at the various challenges that we face in the forward evolution of the reachability set. First, we discuss the issues pertaining to skewed length scales, and then look at the other issues that may affect the accuracy of evolution of the zero level set.

Skewed Parameter Scales:

Most environmental domains and flows such as ocean and atmospheric flows are extremely skewed in terms of length and/or velocity scales. That is, the length of the domain and/or the velocity in one or more directions is much larger than those in the other directions. Even in biological flows such as blood flow in the vascular system, extreme skewness in length-scales is often observed (Wiedeman, 1963). A similar disparity between length and velocity scales is also observed in several artificial flows. For example, some micro-fluidic devices take the advantage of the different scales to efficiently separate the various components of a fluid mixture (Sackmann et al., 2014). Although we study the specifics of ocean flows, the considerations in the following work are general and the conclusions hold for many natural and man-made engineering systems.

As is well known, the depth of oceans is typically much smaller than their longitudinal or latitudinal dimensions. For example, let us look at the deepest point in all of the earth's oceans: the Mariana Trench, which is close to 11,000 meters (11 kilometers) deep. This depth is about 10 times smaller than 1° latitude, which is about 110 kilometers. Oceans typically spans several degrees of latitude and longitude (for example, length of the Pacific ocean is over 15,500 kilometers).

Such skewed length scales significantly affect any simulation based 3D study of oceanic domains. For example, as discussed in chapter 4, we will consider a domain off the coast of New Jersey, in the Atlantic ocean. The length and breadth of the domain is off the order of 400 km, whereas we only consider the top 100 meters as the operable region for our AUVs. The domain is divided in 128 cells along latitudinal and longitudinal directions respectively. This implies that the length of an individual cell in the longitudinal and latitudinal (X and Y) directions ($\sim \mathcal{O}(3 \text{ km})$) is much larger than the depth of the domain (Z direction). This implies that our 3 dimensional grid cells are extremely thin, and hence even extremely small deviations from the exact solutions in the horizontal directions may cause a major change in the vertical direction.

Similar to length scales, the velocity magnitudes in different directions are also extremely different. The vertical velocity of ocean flows may be orders of magnitude smaller than the latitudinal or longitudinal velocity. Typically, the vertical velocity is obtained as a diagnostic variable especially in hydrostatic ocean models, by enforcing the divergence-free condition and accounting for the free surface motion (Haley and

Lermusiaux, 2010)), which is typically much smaller in magnitude than the other velocity components. This implies that, while computing the advective fluxes, the vertical fluxes may be severely altered even by small relative errors in the fluxes in other directions, thus impacting the accuracy of the zero level set evolution in 3D.

Accuracy in Level Set Evolution:

In the forward evolution of the level set function ϕ, what we are really interested in is the movement of the zero level set front. Particularly, the evolution of the zero level set needs to be as close to the truth as possible, because, small deviations from the truth may not change the arrival times much, but could severely alter the computation of the optimal path. As mentioned in chapter 2, we utilize the entropy condition specified by Sethian (Sethian, 1994)) in order to solve for the viscosity solution of equation (2.19). Through this condition, the value of the level set function inside the reachability set is set to some constant different from the other permissible values of the level set function (typically this value is set to be some negative constant). The level set function behaves is a signed distance function with respect to the reachability front outside the reachability set. This implies that there is a discontinuity in the level set function at the reachability front. As is well known, finite difference and finite volume numerical schemes are diffusive of shocks and discontinuities, and these errors are amplified with time (mainly due to compounding of errors). Hence high order schemes are required for the spatial as well as temporal solves for the evolution of the reachability set.

For the case of anisotropic vehicle speeds, there is a need for maximization over a term concerning the gradient of the level set function. For vehicles whose speed strongly depends on the heading, slight errors in the level set function computation can easily be amplified. Further, for vehicles with only a few permissible heading directions, slight deviations from the truth may cause the computed heading direction to be vastly different from the optimal one, this effect aggravating over time. Hence, due to the multiple reasons mentioned above, high accuracy in the forward evolution of the level set function is required.

3.1.2 Challenges in Backward Tracing

We now focus our attention on the issues faced in the backward tracing of the optimal path, once the forward evolution of the reachability set is complete. Specifically, we look at the requirement of numerical consistency in temporal schemes between forward and backward evolutions, and the possibility of amplification of errors.

Consistency in Temporal Schemes:

An important consistency in the computation of the optimal path by solving equation (2.22) is the consistency of temporal schemes between the forward evolution and the backtracking. As we solve equation (2.22), we are solving for a characteristic of equation (2.19). That is, we are trying to track the position of a hypothetical vehicle that always remained on the zero level set of ϕ as it traveled and its position coincided with the destination $\mathbf{x}_f$ at $\mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0)$. In the limiting case of $\Delta t \to 0$, the path of this vehicle will be traced in continuous time. But, as we can only solve numerically for a finitely small Δt, care needs to be taken to consistently follow the path of this hypothetical vehicle.

The forward reachability evolution is governed by a PDE, and hence it represents the motion of the hypothetical 'optimal' point that reaches the destination through an implicit functional representation of the reachability front in space. Note that, even though the above-mentioned PDE implicitly represents the reachability front, it is solved using explicit numerical time integration. Backward tracing of the optimal involves an ODE, which means that we explicitly (in space) trace the motion of the particular optimal point in time. For minimal numerical errors, we would like to match the implicit functional representation of the point (in space) in an exact fashion while backtracking the path, which in turn, is the position of the optimal point as a function of time. Specifically, we need to maintain two numerical consistency properties: forward/backward explicit/implicit consistency for the numerical time integration. and type of time-integration consistency. First, if the forward temporal evolution is explicit, then the backward temporal evolution needs to be implicit, and vice versa. Second, the exact nature of the forward evolution scheme also needs to be 'mimicked' by the back tracing scheme. For example, if the forward evolution uses the forward Euler time marching, then the backward tracing must to use the backward Euler scheme (marching backward in time). If such consistency is not maintained, then backward tracking will not guarantee the exact position tracing of the optimal vehicle in time, and hence the path constructed will not be accurate and numerically consistent.

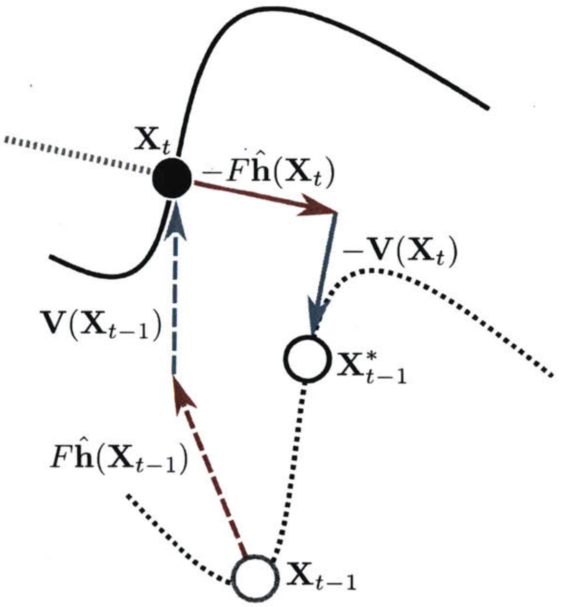

$\mathbf{X}_t$
$-F\hat{\mathbf{h}}(\mathbf{X}_t)$
$-\mathbf{V}(\mathbf{X}_t)$
$\mathbf{V}(\mathbf{X}_{t-1})$
$\mathbf{X}^*_{t-1}$
$F\hat{\mathbf{h}}(\mathbf{X}_{t-1})$
$\mathbf{X}_{t-1}$

Figure 3-1: Error in backward tracing due to numerical temporal inconsistency. The external velocity (cyan arrow) and optimal heading direction (red arrow) are chosen to be those at point $\mathbf{X}_t$ (normal to the zero level set at this point shown by dotted gray line), which leads the vehicle to $\mathbf{X}^*_t$. However, the forward evolution was explicit; that is the external velocity and optimal heading values point at $\mathbf{X}_{t-1}$ were used in order to move from $\mathbf{X}_{t-1}$ to $\mathbf{X}_t$; shown with dotted cyan and red lines respectively. This inconsistency implies that the computed position of the vehicle , $\mathbf{X}^*_{t-1}$ is different from the correct position $\mathbf{X}_{t-1}$. In addition, further errors are caused if compatible forward and backward time integration schemes are not used for the PDE and the ODE respectively. For example, if one uses the forward Euler scheme for the PDE evolution and a Runge-Kutta (RK) scheme for the backward evolution, then the exact nature of the schemes is different. This causes additional numerical errors. Alleviation of such errors is crucial, and will be discussed later in the thesis.

Figure 3-1 schematically represents the discrepancy arising if the correct numerical temporal consistency is not maintained (we consider the isotropic speed case here, but the same issues hold true for the anisotropic speed cases as well). The actual path that was followed by the fictitious vehicle during forward evolution is shown in dotted lines. This path uses the normal direction and velocity of point $\mathbf{X}_{t-1}$, assuming explicit forward evolution. Ideally, our backtracking scheme should take the vehicle from $\mathbf{X}_t$ to reach $\mathbf{X}_{t-1}$. However, if an explicit backtracking scheme is also used, then, the normal direction and the velocity values at point $\mathbf{X}_t$ are used. In this backtracking, the resulting point reached is then $\mathbf{X}^*_{t-1}$, which may be far off from $\mathbf{X}_{t-1}$. Such an error is typically worse if the normal direction to the zero level set changes rapidly in the vicinity of $\mathbf{X}_t$ and is amplified through time marching.

Amplification of Errors During Time-Integration:

We now discuss the time-compounded effects of numerical errors in the forward and backward solves on the optimal paths that are obtained. All of the issues discussed above initially introduce local errors, that are compounded as time marching is carried out. Such growing errors may cause the computed optimal paths to be far off from the real optimal paths, and if not addressed, would lead to a low confidence in the results. Specifically two scenarios may emerge:

1. *Backward accumulation of errors.* While solving ODE (2.22) backward in time, we start from the target position and utilize the forward PDE computed optimal headings and the time dependent velocity field at trajectory points to evolve the optimal path. The introduced errors may imply that the vehicle that starts from the target position and numerically backtracks the computed optimal path will never reach the start point.

2. *Effect of accumulation of errors on the predicted forward vehicle motions.* Solving ODE (2.22) backward not only provides us with the optimal path, but also gives us the time series of optimal headings at which the vehicle should be steered in order to reach the target in the shortest time. As a verification, one can then check to see if the vehicle steered with the computed optimal headings actually reaches to the target point. Similarly to the last case, errors in the optimal headings solution may imply that if we steer our numerical vehicle

forward from the start point towards to the target point along the computed optimal headings (forward ODE time-integration), the vehicle will never reach the target position.

Either of the cases described above is unacceptable, and all computed optimal paths should be tested with these two checks. Thus, it is clear that even small numerical errors or inconsistencies may end up causing large deviations from the exact solutions, and hence we need to make sure to be computationally accurate and consistent throughout the solve.

3.1.3 Solutions

As seen above, the major numerical challenges we face while solving path planning problems for realistic complex oceanic or environmental flows and domains are that: (i) the length and velocity scales are extremely skewed; (ii) slight errors in the forward evolution caused by low-order numerical schemes and the compounding numerical errors lead to large differences in the computed optimal headings (iii) diverging paths are computed if correct numerical consistency is not maintained throughout the computation, and (iv) the computed optimal headings may not actually lead to feasible paths.

Next, we address the approaches to alleviate these difficulties. First, we non-dimensionalize the Hamilton-Jacobi level set equation governing the forward evolution, that is equation (2.19). This implies that the various length and velocity scales now do not affect the accuracy of computation, as each of the non-dimensional terms is of the same order of magnitude.

We then solve this non-dimensionalized equation with highly accurate numerical schemes to capture exactly the evolution of the zero level set front. We provide schemes and illustrations up to 5^{th} order in space and 3^{rd} order in time. This implies that the computed optimal headings are close to exact, which is necessary for a near-exact optimal path. Note that, one can go to even higher order schemes than the ones mentioned, but the gain in accuracy is minuscule as compared to the increased computational expenses.

Lastly, as mentioned before, we make sure that numerical temporal consistency is maintained throughout. As it is much easier to solve a PDE through explicit time marching, we use implicit time marching for the backward tracing of optimal path. That is the headings and velocities are implicitly represented in which case an iterative solver is necessary to arrive at the correct answer. Further details regarding implicit backtracking are discussed in section §(3.5).

3.2 Non-Dimensionalization of the Hamilton-Jacobi Equation

In this section, we discuss the non-dimensionalization of the Hamilton-Jacobi equation governing the evolution of the level set function. Specifically, we look at equation (2.30), restated here for convenience.

$$\frac{\partial \phi}{\partial t} + \max_{\hat{\mathbf{h}}} \left(F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) \cdot \nabla \phi \right) + \mathbf{V}(\mathbf{x}, t) \cdot \nabla \phi = 0 \tag{3.1}$$

We now look at the non-dimensionalization of each term on the left hand side. For this purpose, a reference scale needs to be set up for each of the variables. The choice of the reference scales for each of the spatial dimensions, as well as each of the velocities utilizes information about the domain length and the approximate velocity magnitudes in the respective dimensions. The other variables that remain are the level set function ϕ and the time t. We propose the following non-dimensionalizing scales for each of the variables, summarized in table 3.1.

Note that the scales chosen for the spatial variables require minimal computation. The characteristic scales for the velocities consider an average over the entire spatial and temporal domain of interest, and are required to be computed once per simulation. Note that another suitable scale for velocities could have also been the maximum velocity in that direction over the entire temporal domain of interest. Although an acceptable choice, the maximum velocity might be much larger than the

Table 3.1: Scales for non-dimensionalization

Variable	Scale
x	$x^c = L_x$ (Characteristic length in the X direction)
y	$y^c = L_y$ (Characteristic length in the Y direction)
z	$z^c = L_z$ (Characteristic length in the Z direction)
V_x	$V_x^c = \text{mean}_{\mathbf{x},t}(V_x(\mathbf{x}, t))$
V_y	$V_y^c = \text{mean}_{\mathbf{x},t}(V_y(\mathbf{x}, t))$
V_z	$V_z^c = \text{mean}_{\mathbf{x},t}(V_z(\mathbf{x}, t))$
t	$t^c = \max\left(\frac{\lvert x_f - x_s\rvert}{\max(V_x^c, F_x^{\max})}, \frac{\lvert y_f - y_s\rvert}{\max(V_y^c, F_y^{\max})}, \frac{\lvert z_f - z_s\rvert}{\max(V_z^c, F_z^{\max})}\right)$

typical expected velocity magnitudes, and hence the non-dimensionalization is not very effective in such cases. If the velocities in different regions vary greatly, then the domain can be divided into appropriate sub-domains and a separate non-dimensional scale for the velocities can also be chosen in each of the domains. The characteristic time scale proposed is an estimate for the travel time assuming that the vehicle either travels only at its own speed or is only advected with the flow, whichever is the faster choice. Specifically, we look at the maximum of the travel times along each of the basis directions (that is, along the X, Y and Y directions) when the vehicle optimally chooses the to either travel with the flow, or travel on its own, but not both. It must be stated that these choices are far from unique, and appropriate characteristic scales should be adapted to the problem.

A peculiar point to notice is that we have not defined any scale for the level set function ϕ in table 3.1. This is because, when the terms in the equation are replaced by the corresponding non-dimensional quantities, the characteristic scale of ϕ is linearly involved with each term, and hence it can be factored out. Hence, ϕ does not require any scale of its own. Note that, if there were additional terms involved in the equation due to various effects, such that ϕ cannot be factored out, then one can choose an appropriate scale for it by using its interpretation as the signed distance function from the reachability front. For example, a characteristic scale that we propose for ϕ is the maximum distance between any point in the domain and the start position of the vehicle. That is,

$$\phi^c = \max_{\mathbf{x}\in\Omega} (|\mathbf{x} - \mathbf{x}_s|) \tag{3.2}$$

Dividing all the variables with the concerned characteristic scales, we obtain a new form of equation (2.30), which is numerically well conditioned. This non-dimensional form is given by equation (3.3).

$$\begin{aligned}\frac{1}{t^c}\frac{\partial\phi^*}{\partial t^*} + \max_{\hat{h}_x,\hat{h}_y,\hat{h}_z}\left(F(\hat{\mathbf{h}},t)\left(\frac{\hat{h}_x(t)}{x^c}\frac{\partial\phi^*}{\partial x^*} + \frac{\hat{h}_y(t)}{y^c}\frac{\partial\phi^*}{\partial y^*} + \frac{\hat{h}_z(t)}{z^c}\frac{\partial\phi^*}{\partial z^*}\right)\right) \\ + \left(\frac{V_x^c}{x^c}V_x^*\frac{\partial\phi^*}{\partial x^*} + \frac{V_y^c}{y^c}V_y^*\frac{\partial\phi^*}{\partial y^*} + \frac{V_z^c}{z^c}V_z^*\frac{\partial\phi^*}{\partial z^*}\right) = 0\end{aligned} \tag{3.3}$$

Where the superscript ($\bullet^*$) refers to the non-dimensional quantity corresponding to ($\bullet$). For the case of isotropic speeds, this equation reduces to equation (3.4).

$$\begin{aligned}\frac{1}{t^c}\frac{\partial\phi^*}{\partial t^*} + F\sqrt{\frac{1}{(x^c)^2}\left(\frac{\partial\phi^*}{\partial x^*}\right)^2 + \frac{1}{(y^c)^2}\left(\frac{\partial\phi^*}{\partial y^*}\right)^2 + \frac{1}{(z^c)^2}\left(\frac{\partial\phi^*}{\partial z^*}\right)^2} \\ + \left(\frac{V_x^c}{x^c}V_x^*\frac{\partial\phi^*}{\partial x^*} + \frac{V_y^c}{y^c}V_y^*\frac{\partial\phi^*}{\partial y^*} + \frac{V_z^c}{z^c}V_z^*\frac{\partial\phi^*}{\partial z^*}\right) = 0\end{aligned} \tag{3.4}$$

From these equations, it can be seen that the advective fluxes in each direction scale with the ratio $\frac{V}{L}$ in that corresponding direction. As stated before, even though both V_z and L_z are much smaller that either of V_x and L_x or V_y and L_y, the ratio $\frac{V_z}{L_z}$ can easily be comparable to $\frac{V_x}{L_x}$ and $\frac{V_y}{L_y}$. This means that even though the length and the velocity in the Z direction is small, the advection contribution in this direction is not negligible. This arises from the incompressible continuity equation.

Focusing our attention on the optimal propulsion term (second term on the left hand side), it is clear that the contribution due to each of the directional terms scales as $\frac{1}{L}$. This again implies that even though length of the domain in the Z direction is much smaller, the contribution to the optimal propulsion term in this direction can be substantial. Hence, ignoring the vertical direction may cause large errors, as its contributions to both the optimal propulsion and advection are comparable to the other existing terms.

3.3 Forward Evolution

We now discuss the numerical schemes and the general implementation for solving the forward evolution of ϕ by using either equation (3.3) or equation (3.4). First, we take a look at the solver and the grid details as well as the imposed initial and boundary conditions. This is followed by a discussion on advection and optimal propulsion schemes. Lastly, we look at the various high order time marching schemes utilized in this work.

We use a 3D structured grid with rectangular prismatic elements in a cuboid domain. All the equations are solved using the finite volume method as it allows for an easier implementation of high order schemes. In our case, the velocity field is deterministic and is known a priori, hence we do not have any coupling of variables. Further, we solve for the optimal propulsion and the advection terms independently of each other. If the domain is irregular in shape, then the extraneous regions are masked, and the external velocity as well as the vehicle speed in these regions is set to be zero. This ensures that the values of the level set function never changes in these regions.

We use the initial conditions provided by equation (2.20). Although this is an ideal initial condition, computationally, this introduces a singularity at the start point. Hence, we modify this initial condition as:

$$\phi(\mathbf{x}, 0) = |\mathbf{x} - \mathbf{x}_s| - r_0 \tag{3.5}$$

Where r_0 is a small quantity (comparable to the grid size). This creates a small zero level set sphere around the start point of radius r_0. The interior of this sphere is then set to a constant negative value due to the entropy condition. This eliminates the singularity and provides a smooth initial condition. This implies that while backtracking, the optimal path will only guide us to the surface of this sphere. As r_0 is small enough, we can then claim that the vehicle can travel from this surface point to the actual start position in a straight line.

The evolution of the reachability front does not depend on the function values away from the front. Hence the boundary conditions do not have a direct impact on the evolution of the zero level set. This only restriction on the boundary conditions is that they should not introduce any new reachable points in the domain. To this end, we use radiation boundary conditions for ϕ at all boundaries. That is, the imposed boundary condition is $\frac{\partial^2 \phi}{\partial n^2} = 0$, where n is the local surface normal. While solving numerically, this derivative is computed such that the order of discretization matches the order of advection and optimal propulsion schemes.

We now discuss the numerical schemes applied and implemented to solve for the advection and the optimal propulsion terms and the time marching schemes.

3.3.1 Advection Term

The numerical schemes implemented for solving the advection term utilize the given velocity field explicitly, and find suitable approximations for $\nabla\phi$ by using the sign of this known velocity field. We implement schemes with order of accuracy up to 5.

We use the finite volume method to solve the PDE at hand. As is well known, the finite volume method is a reformulation of the control volume approach applied to individual computational elements. This means that we require estimates of ϕ values on the faces of the individual grid cells.

A simple first order scheme uses upwinding to approximate ϕ values on the cell faces. Looking in the X direction, the east face value is either set to be the value of the east cell or the value of the present cell depending on the sign of the velocity V_x.

Similar idea also applies for the Y and the Z directions. Specifically,

$$\phi_e = \begin{cases} \phi_E \text{ if } V_x < 0 \\ \phi_P \text{ if } V_x > 0 \end{cases} \tag{3.6}$$

where the subscript e specifies the east face, and the subscripts E and P denote the east and the present cells respectively. Similar equations are considered for the west, the north, the south, the front and the back faces. Once the face values are approximated, the net flux contribution in all the directions can be computed, and their addition (after multiplying by the appropriate dimensional constants) yields the net advective flux contribution for each cell.

Alternatively, the advective fluxes can also be computed by using the total variation diminishing (TVD) schemes. These schemes ensure that the total variation, given by $TV = \int \left|\frac{\partial u}{\partial x}\right|$ does not grow in time. The second order TVD scheme with various flux limiters is implemented in Ueckermann and Lermusiaux (2012), which is able to solve the optimal path planning problem in 2 dimensions. An issue with TVD schemes involves their their extension to higher order of accuracy (Sweby, 1984; LeVeque, 2002). This prompts us to resort back to upwinding in order to use higher order numerical schemes.

The current work utilizes the essentially non-oscillatory (ENO) schemes and its extension to the weighted essentially non-oscillatory (WENO) scheme for Hamilton-Jacobi equations (Shu and Osher, 1988; Jiang and Shu, 1996). The general unsteady Hamilton-Jacobi equation, given by equation (2.10), only contains at most the first derivatives of the unknown function ϕ. ENO and WENO schemes find highly accurate estimates for this first derivative, and hence they can be used for any Hamilton-Jacobi equation. We combine upwinding with these schemes to obtain up to 5^{th} order accuracy in the advective flux estimates. As these schemes provide us with the estimates for the derivatives of ϕ (that is, of $\nabla\phi$), they can also be used in the computation of the optimal propulsion term with equal accuracy, as will be seen later. These schemes are presented in detail in appendix §$(B.1)$ and §$(B.2)$.

3.3.2 Optimal Propulsion Term

The second contribution to the rate of change of the level set function ϕ is through the optimal movement of the vehicle due to its own speed. We now look at the numerical schemes used to compute the contribution of this term, such that the overall order of accuracy is maintained. We first look at the case of isotropic speed, for which equation (2.19) reduces to equation (2.25). We then look at the methods to compute the optimal propulsion term for vehicles with anisotropic speed.

Isotropic Speed

For the case of heading independent speed, the optimal propulsion term is given by:

$$F|\nabla\phi| = F\left(\left(\frac{\partial\phi}{\partial x}\right)^2 + \left(\frac{\partial\phi}{\partial y}\right)^2 + \left(\frac{\partial\phi}{\partial z}\right)^2\right)^{\frac{1}{2}} \tag{3.7}$$

We use the Godunov scheme to compute $F|\nabla\phi|$ efficiently by using the estimates for ϕ_x, ϕ_y and ϕ_z that were computed previously (Osher and Fedkiw, 2006). The Godunov scheme can be expressed in a concise manner by equation (3.8).

$$\begin{aligned}
\phi_x^2 &= \max\left(\max\left(\phi_x^-, 0\right)^2, \min\left(\phi_x^+, 0\right)^2\right) \\
\phi_y^2 &= \max\left(\max\left(\phi_y^-, 0\right)^2, \min\left(\phi_y^+, 0\right)^2\right) \\
\phi_z^2 &= \max\left(\max\left(\phi_z^-, 0\right)^2, \min\left(\phi_z^+, 0\right)^2\right)
\end{aligned} \tag{3.8}$$

where ϕ_x, ϕ_y and ϕ_z represent the x, y and z derivatives of ϕ respectively. Further, ϕ_i^+ represents the forward difference approximation to ϕ_i and ϕ_i^- represents the backward difference approximation to ϕ_i. That is:

$$\begin{aligned}
\phi_i^+ &= \frac{\phi_{i+1} - \phi_i}{\Delta i} \\
\phi_i^- &= \frac{\phi_i - \phi_{i-1}}{\Delta i}
\end{aligned} \tag{3.9}$$

As we are computing $|\nabla\phi|$, we only require ϕ_x^2, ϕ_y^2 and ϕ_z^2, and not the actual gradient values. Further, we have represented the entire scheme in terms of ϕ_x^- and ϕ_x^+ only.

This makes it convenient and efficient to use appropriate estimates for ϕ_x^- and ϕ_x^+ borrowed from the advection computation. It also ensures that the order of accuracy maintained, and increases the computational efficiency as the estimates for ϕ_x^- and ϕ_x^+ have to be computed only once.

The numerical details behind the Godunov scheme and equation (3.8) are discussed in section §(B.3).

Anisotropic Speed

If the speed of the vehicle depends on the direction of travel, the optimal propulsion term is given by:

$$\max_{\hat{\mathbf{h}}} \left(F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) \cdot \nabla\phi \right) \tag{3.10}$$

In the most general case, this cannot be reduced further. In order to compute the contribution of this term, a maximization is required to be performed at each point in the domain, at all times. We discuss some approaches to solve this maximization problem, and discuss the advantages/disadvantages of each method.

In some cases, if the analytical expression for vehicle speed F in terms of the heading $\hat{\mathbf{h}}$ (and the time t) is known, then the maximization may be performed analytically. As an example, we consider the case of an oceanic float. These vehicles can only travel in the vertical direction by adjusting their buoyancy, and are advected by the flow in the horizontal plane. We model this by enforcing the following:

$$\hat{\mathbf{h}} = \{\hat{n_z}, -\hat{n_z}, 0\} \tag{3.11}$$

where $\hat{n_z} = (0, 0, 1)$ is the unit vector along the Z (vertical) direction. This implies that the only directions that the vehicle can be steered in are vertically up or vertically down (or the vehicle can stay at the same place). Using this in the maximization term, we get equation (3.12).

$$\begin{aligned}\max_{\hat{\mathbf{h}}}\left(F(\hat{\mathbf{h}},t)\hat{\mathbf{h}}(t)\cdot\nabla\phi\right) &= F\max\left((0,0,\pm1)\cdot(\phi_x,\phi_y,\phi_z)\right)\\ &= \max\left(\pm\phi_z\right) \qquad (3.12)\\ &= \left|\frac{\partial\phi}{\partial z}\right|\end{aligned}$$

Hence, the final Hamilton-Jacobi equation to be solved for the motion of a float is given by equation (3.13)

$$\frac{\partial\phi}{\partial t}+\mathbf{V}\cdot\nabla\phi+F\left|\frac{\partial\phi}{\partial z}\right|=0 \qquad (3.13)$$

with the optimal heading given by:

$$\hat{\mathbf{h}}(t)=sgn(\frac{\partial\phi}{\partial z}) \qquad (3.14)$$

where *sgn* is the signum function.

Similarly, if the vehicle is permitted to only travel along certain directions then the maximum is calculated over all the directional derivatives along the permitted directions.

Another way to solve the maximization is to construct a look-up table prior to the forward evolution (Lolla, 2016; Lolla and Lermusiaux, 2017). This table holds the optimal heading directions for all possible level set gradient directions. That is, given $\frac{\nabla\phi(\mathbf{x},t)}{|\nabla\phi(\mathbf{x},t)|}$ (unit vector in the direction of $\nabla\phi(\mathbf{x},t)$), the table returns optimal $\hat{\mathbf{h}}(\mathbf{x},t)$ such that the propulsion term is optimized. This table is constructed in a brute-force sense, where the value of the optimal propulsion term is computed for all possible headings, the maximum one is chosen. This is referred to as 'offline maximization'. This maximization is only performed once, at the start of the simulation, in the case when the vehicle speed does not depend on time.

Such a maximization can also be performed after every time step while the forward evolution of ϕ is being carried out. This is called 'online maximization'. In this case, the optimal heading direction is computed at each point and at each time by testing all permissible heading directions and choosing the maximum.

Often, offline maximization is computationally favorable (especially when the speed does not vary with time) as the propulsion term is only required to be computed once per heading direction. If the maximum speed varies with time, then the online maximization may prove to be more efficient, as in such a case the offline table has to be separately constructed for each time. Care needs to be taken in making sure that the discretization of the polar domain for the computation of the optimal heading is comparable to the grid discretization, otherwise unfavorable errors may be introduced. As of the isotropic speed case, the formulation for anisotropic speed vehicles is also in terms of $\nabla\phi$ or its components. The specific value of $\nabla\phi$ is borrowed from the advection term computation as it maintains the order of accuracy, and also avoids repetitive computation.

3.3.3 Time Marching

We now discuss the time marching schemes used in the forward evolution computation. All the temporal schemes used are explicit and are up to 3^{rd} order in accuracy.

Often, it is desirable to have a higher order of accuracy for the temporal error as well. For higher accuracy, we resort to the Runge-Kutta (RK) schemes (Gottlieb and Shu, 1998; Osher and Fedkiw, 2006). Specifically, we use the total variation diminishing (TVD) Runge-Kutta schemes. While there are numerous RK schemes, these TVD RK schemes guarantee that no spurious oscillations are produced as a consequence of the higher-order accurate temporal discretization, as long as no spurious oscillations in time are produced with the forward Euler scheme, which is the building block of these schemes. The first order accurate TVD RK scheme is just the standard forward Euler time marching scheme. We assume that the forward Euler method is TVD in conjunction with the spatial discretization of the PDE. Then higher-order accurate methods are obtained by sequentially taking Euler steps and combining the results with the initial data using a convex combination. Since the Euler steps are assumed TVD and the convex combination operation is TVD as long as the coefficients are positive, the resulting higher-order accurate method is TVD. Unfortunately, ENO and WENO schemes are not TVD when used in conjunction with upwinding. However, it can be shown that ENO and WENO schemes always result in a total variation value that is finite, and hence are considered to be total variation bounded (TVB) scheme (Osher and Fedkiw, 2006). As TVD is a stricter condition than TVB, the numerical method using TVD RK scheme for time marching along with ENO/WENO schemes for spatial discretization is total variation bounded. Numerical details of the 2^{nd} and 3^{rd} order TVD RK schemes are in appendix $\S(B.4)$.

While fourth-order accurate (and higher) TVD RK schemes exist (Gottlieb and Shu, 1998; Gottlieb et al., 2001)), this improved temporal accuracy does not make a significant difference, especially since the spatial schemes also contribute to the net numerical errors. Also, the fourth-order accurate (and higher) TVD RK methods require both the upwind and the downwind differencing approximations, doubling the computational cost of evaluating the spatial operators (Gottlieb et al., 2001). Thus, they are not considered in this work.

3.3.4 Data Storage

During the forward time integration, the zero level set of ϕ represents the reachability front. In order to be able to retrieve the optimal path, we store this front at every time step. The front is extracted from $\phi(\mathbf{x}, t)$ using the surface extraction algorithm in $MATLAB^{®}$. As the reachability front is always 1 dimension lower than the domain (it is a curve in 2D, and a surface in 3D), the amount of storage required at each time step for only storing the reachability front is $\mathcal{O}(n^2)$, where n is the number of grid points along a dimension. Thus, only saving the zero level set surface is much cheaper than storage of the entire ϕ field at each time step, which requires $\mathcal{O}(n^3)$ space.

3.4 Reinitialization

This section looks at the reinitialization of the signed distance field (Keck, 1998; Min, 2010; Russo and Smereka, 2000). We first discuss the need of reinitialization, followed by the conditions to be satisfied by the reinitialization methods. Finally,we discuss an efficient reinitialization algorithm that is implemented in this work that utilizes the various numerical schemes described in the prior sections.

As stated by equation (3.5), we initialize the level set function ϕ as a signed distance function with respect to a small spherical surface centered at the start position. As the level set function evolves with time, the zero level set surface grows, propagates and distorts. In the general case, the signed distance property of ϕ is gradually lost due to the presence of the advective velocity field, similar to the work from Lolla (2012). Theoretically, the optimal propulsion term maintains the signed distance property of ϕ through time, but it is also subject to numerical errors.

We now show that the signed distance property of ϕ is not maintained if its evolution is governed by equation (2.29) (Lolla, 2012). It can also be shown that equation (2.30) also does not maintain this property in general, but as equation (2.29) is a special case of equation (2.30) (for vehicles with isotropic speeds), showing that even this special case does not satisfy the condition is sufficient to claim that equation (2.30) does not maintain the signed distance field in general.

Assume that ϕ is a signed distance field at some time t. That is, ϕ is the unique viscosity solution of the following Eikonal equation:

$$|\nabla\phi| = 1 \tag{3.15}$$

Consider the evolution of ϕ, given by equation (2.29). By substituting $|\nabla\phi| = 1$, we obtain the following equation:

$$\frac{\partial\phi}{\partial t} + F(t) + \mathbf{V}(\mathbf{x}, t) \cdot \nabla\phi = 0 \tag{3.16}$$

and taking x, y and z derivatives of equation (3.16), we get equation (3.17).

$$\begin{aligned}
\frac{\partial \phi_x}{\partial t} + \frac{\partial \mathbf{V}}{\partial x} \cdot \nabla\phi + \mathbf{V} \cdot \frac{\partial(\nabla\phi)}{\partial x} &= 0 \\
\frac{\partial \phi_y}{\partial t} + \frac{\partial \mathbf{V}}{\partial y} \cdot \nabla\phi + \mathbf{V} \cdot \frac{\partial(\nabla\phi)}{\partial y} &= 0 \\
\frac{\partial \phi_z}{\partial t} + \frac{\partial \mathbf{V}}{\partial z} \cdot \nabla\phi + \mathbf{V} \cdot \frac{\partial(\nabla\phi)}{\partial z} &= 0
\end{aligned} \tag{3.17}$$

We intend to compute $\frac{\partial|\nabla\phi|}{\partial t}$, which can be written as:

$$\frac{\partial|\nabla\phi|}{\partial t} = \frac{1}{|\nabla\phi|}\left(\phi_x \frac{\partial \phi_x}{\partial t} + \phi_x \frac{\partial \phi_y}{\partial t} + \phi_x \frac{\partial \phi_z}{\partial t}\right) \tag{3.18}$$

Substituting the above quantities,

$$\frac{\partial|\nabla\phi|}{\partial t} = \begin{array}{l} \frac{-1}{|\nabla\phi|}\nabla\phi \cdot \left(\phi_x \frac{\partial \mathbf{V}}{\partial x} + \phi_y \frac{\partial \mathbf{V}}{\partial y} + \phi_z \frac{\partial \mathbf{V}}{\partial z}\right) + \\ \frac{-1}{|\nabla\phi|}\mathbf{V} \cdot \left(\phi_x \frac{\partial(\nabla\phi)}{\partial x} + \phi_y \frac{\partial(\nabla\phi)}{\partial y} + \phi_z \frac{\partial(\nabla\phi)}{\partial z}\right) \end{array} \tag{3.19}$$

This can be simplified as:

$$\frac{\partial|\nabla\phi|}{\partial t} = \frac{-1}{|\nabla\phi|}\left(\nabla\phi \cdot (\nabla\mathbf{V} \cdot \nabla\phi) + \frac{1}{2}\mathbf{V} \cdot \nabla\left(\nabla\phi \cdot \nabla\phi\right)\right) \tag{3.20}$$

Equation (3.20) implies that:

$$\frac{\partial|\nabla\phi|}{\partial t} = 0 \iff 2\nabla\phi \cdot (\nabla\mathbf{V} \cdot \nabla\phi) = -\mathbf{V} \cdot \nabla\left(\nabla\phi \cdot \nabla\phi\right) \tag{3.21}$$

This is an extra imposed condition between the velocity field $\mathbf{V}$ and the level set function ϕ, which may not be true. Hence, equation (3.21) implies that in general,

$$\frac{\partial|\nabla\phi|}{\partial t} \neq 0 \tag{3.22}$$

This in turn means that $|\nabla\phi| \neq 1$ for later times, and hence the signed distance property of ϕ is gradually lost. Further, if $\mathbf{V} = 0$, then $\frac{\partial|\nabla\phi|}{\partial t} = 0$ at all times. This means that the optimal propulsion term maintains the signed distance property of the function ϕ. Note that, the loss of signed distance property is not at all a numerical issue. It is the correct property of the exact viscosity solution of equation (2.29). Hence, even with accurate numerical schemes and even with exact schemes, ϕ will deviate from a signed distance function as it is evolved through equation (2.29).

This may cause the field $\phi(\mathbf{x}, t)$ to develop steep gradients at some places and shallow gradients at other places. Analytically, this is not an issue, but computationally, this can lead to large numerical errors in the evolution of the reachability front and in the evaluation of its geometrical properties such as the directions of the normals and the curvature. This problem, in general, cannot be alleviated by using a higher order scheme to approximate the spatial gradients, or for the time integration as shown by Mulder et al. (1992). Reinitialization is defined as a process in which $\phi(\mathbf{x}, t)$ is reset to a new scalar signed-distance field with the zero iso-contour being unchanged. We require the reinitialization method to:

- Modify $\phi(\mathbf{x}, t)$ such that it now satisfies the signed distance property with respect to the current zero level set of $\phi(\mathbf{x}, t)$

- Keep the current zero level set of $\phi(\mathbf{x}, t)$ unchanged

An intuitive scheme to reinitialize ϕ is to compute the distance of each point in the domain with respect to all the points on the zero level set, and then choose the minimum amongst them. Although extremely easy to implement, this procedure is intensive on both memory and computation and hence it is seldom used. The computational cost of such a method is $\mathcal{O}(n^3)$ and if reinitialization is required often, then this step becomes the bottleneck of using level set method.

We now look at a PDE based scheme implemented in the present work for efficient reinitialization of the function ϕ, first given by Sussman et al. (1994a), and later worked on by multiple researchers (Sussman et al., 1998; Sethian and Smereka, 2003; Sun et al., 2010). Consider the following equation:

$$\frac{\partial \phi}{\partial t} + sgn(\phi_0)\left(|\nabla \phi| - 1\right) = 0 \tag{3.23}$$

with the initial condition $\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x})$, and ϕ_0 implicitly defines the zero level set at a particular time step through its zero iso-contour. Equation (3.23) is a Hamilton-Jacobi equation, and is solved in pseudo-time to steady state. Solution of equation (3.23) implies that $|\nabla\phi| = 1$, which is the signed distance with respect to the initial zero iso-contour provided through ϕ_0. This equation can be solved using the spatial and temporal schemes discussed before for high accuracy. Assuming smooth fields, the distorted ϕ (which is to be reinitialized) is not far from a signed distance function, and in such cases this equation quickly converges to steady state. A known drawback of this method is that it may shrink the volume enclosed by the zero level set surface, displacing it inward. In order to fix this issue, we use the idea of subcell fix, as given by Russo and Smereka (2000). This method still solves equation (3.23), but now the gradients within one grid cell of the zero level set surface are computed by utilizing the information that $\phi = 0$ on the surface. Gradients farther than one grid cell are computed as before. This drastically reduces the displacement of the zero level set during reinitialization. This scheme for reinitialization is implemented in three dimensions in the current work. More details and further modifications to this method may be found in Sun et al. (2010); Sussman et al. (1994b); Zhao et al. (2001).

Another important aspect is the frequency of reinitialization. Reinitializing the level set field after every time step may marginally increase the accuracy, but it is extremely expensive. Hence, we reinitialize ϕ after a fixed number of time steps or after the Fréchet distance between the zero level sets exceeds some threshold, whichever is lower. This choice is somewhat empirical, and may vary with the problem.

It must be noted that there exist many more reinitialization techniques, and this is an area of active research. Min (2010) details some of the commonly used techniques. A useful and efficient approach to reinitialization is to solve equation (2.9) by using fast marching method (Sethian, 1996, 1999a)) the computational cost of which is $\mathcal{O}(n \cdot log(n))$, a significant improvement over the $\mathcal{O}(n^3)$ operations required by the brute-force method. Chopp (2009) and Adalsteinsson and Sethian (1999) present an approach using extension velocities that maintains the signed distance property of ϕ. Li et al. (2005) consider a cost function which ensures that ϕ always remains close to a signed distance field, whereas Ovsyannikov et al. (2012); Sabelnikov et al. (2014) add a source term to equation (2.19) such that ϕ remains a signed distance field.

As mentioned several times before, even though we solve for $\phi(\mathbf{x}, t)$ at all points in the domain, the only quantity we are interested is the position of the zero level set surface (as a function of time). This is unaffected by the evolution of the level set function away from the zero level set at all times. Hence, an efficient approach is to only solve equation (2.25) in the vicinity of the zero level set surface at each time instant. This approach is called the 'narrow band level set method' and is an efficient alternative to the regular level set method. First introduced by Adalsteinsson and Sethian (1995), this approach only computes ϕ values within a set distance – also called the 'active band' near the zero level set. Reinitialization is required at each time step in this approach as the new grid cells where ϕ is undefined enter the active band as it moves with the zero level set. Efficient reinitialization schemes given by Adalsteinsson and Sethian (1995); Peng et al. (1999) and other approaches based on the fast marching method provide an efficient way to achieve this. Note that, even though narrow band level set method requires extra computations (for reinitialization) at every time step, the savings due to not solving for ϕ away from the zero level set are large, and the net computational cost of the narrow band level set method is often much cheaper than the regular level set computation.

3.5 Implicit Backward Tracing

In this section we look at the second part of the path planning problem, namely the backward tracing of the optimal path once the destination lies on the zero level set of ϕ. Optimal path from the start position $\mathbf{x}_s$ to the destination $\mathbf{x}_f$ is a characteristic line of the Hamilton-Jacobi equation governing the evolution of ϕ, that passes through the destination.

The optimal path is obtained by solving the ODE given by equation (3.24) backward in time, with the initial condition $\mathbf{X}(t = \mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0)) = \mathbf{x}_f$.

$$\frac{d\mathbf{X}}{dt} = \mathbf{V}(\mathbf{X}, t) + F\hat{\mathbf{h}}(\mathbf{X}, t) \tag{3.24}$$

While solving for a characteristic, the consistency of temporal schemes must be maintained. As all the time marching schemes that we utilize for the Hamilton-Jacobi equation are explicit in nature (i.e., explicit time integration is used while solving the PDE (2.29)), we find that the corresponding schemes for the optimal path computation must be implicit. Further, the scheme used for characteristic computation must mimic the temporal scheme from the forward evolution computation. These are new results that we present next and that were implemented for 3D domains and flows.

3.5.1 Computation of Normals

At each step during the forward evolution, only the position of the zero level set surface is stored. In order to do this, the surface represented by multiple triangular surfaces, and the vertices of all these triangles along with the connectivity matrix of these vertices is stored.

We first study the case of the vehicles with isotropic speed. For this case, the optimal headings $\hat{\mathbf{h}}$ are the normals to the zero level set surface. The normal to the zero level set surface at any point lying on its surface is computed in the following way:

1. The triangle to which the given point belongs is found out.

2. The normal direction to the plane of this triangle is computed by calculating the cross product of the edge vectors. This normal is placed at the centroid of the triangle.

3. Normal directions at the vertices of this triangle are computed. Normal at each of the vertex is computed by first computing normals to all the triangles that have the said point as a vertex. As shown in figure 3-2, the normal direction at this vertex is then given by a weighted combination of the constituent triangle

normal directions, where the weights are proportional to the angle subtended by the corresponding triangle and the concerned vertex.

4. Once the normals at each of the vertex and the centroid are computed, the normal at the given point is computed by a weighted average of these 4 normals, where the weights are proportional to the distance between the given point and the location of the normal.

5. It is important to ensure that the chosen normal direction points outwards (with respect to the reachability set). The is checked by considering the dot product of the normal vector with $\nabla\phi$. It is safe to assume that ϕ does not deviate much with respect to the signed distance function. Hence, the orientation of the normal that has positive projection along $\nabla\phi$ is chosen as the outward pointing normal.

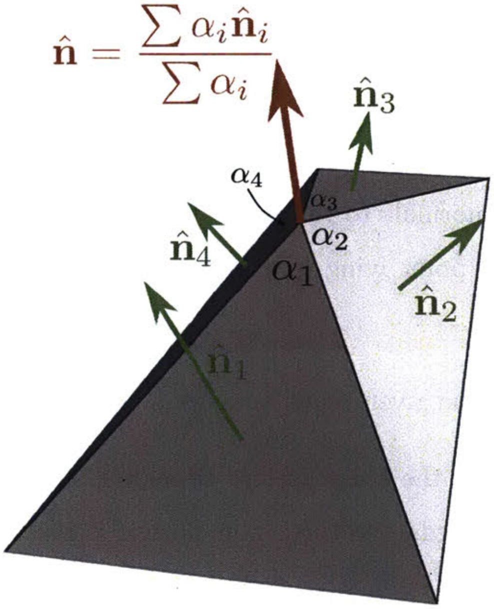


Figure 3-2: Computation of normals at the vertices. In order to compute the normal direction at the vertex($\hat{\mathbf{n}}$), we consider the weighted average of the normals of all the triangles that have the considered point as a vertex ($\hat{\mathbf{n}}_i$), where the weighting factor is the angle subtended by each of the triangle at this vertex (α_i). Once the normals at all the vertices are computed, these normals along with the surface normal (situated at the centroid) are used to compute normal at the required point.

In the case of vehicles with anisotropic speed, the normal direction at a given point is computed in the same fashion, and the optimal heading is obtained by querying it in the offline maximization table.

3.5.2 Backtracking Schemes

We now obtain various backtracking schemes that are numerically consistent with the corresponding forward evolution schemes. We only discuss the schemes for vehicles with isotropic speeds. The extension to vehicles with anisotropic speeds is straightforward through the use of the maximization table, as mentioned above. As the forward evolution is explicit, all these backtracking schemes are implicit, and iterative solves are required at each time step. Our naming convention is such that the references to the backtracking schemes are made by the corresponding forward evolution schemes. For example, the backtracking scheme corresponding to the forward Euler temporal evolution of the PDE is referred to as the forward Euler backtracking scheme. Hence, even though the name of the backtracking scheme suggests explicit nature, they indeed are implicit.

Figure 3-3 schematically represents the methodology behind our implicit backtracking schemes. The idea is to iteratively obtain better and better guesses for the optimal heading and external velocity values until the solution converges.

Following notation is adopted for all schemes for the ease of understanding: superscript $< \bullet >$ represents iteration number at a specific temporal instant. The time is indicated in index form (indexed with an interval of Δt) and denoted as a subscript.

Stopping Criterion

As with almost all other implicit iterative schemes, there is no optimal stopping criterion. There are multiple stopping criteria that are based on either the change between the successive solutions or on the difference between the initial solution and

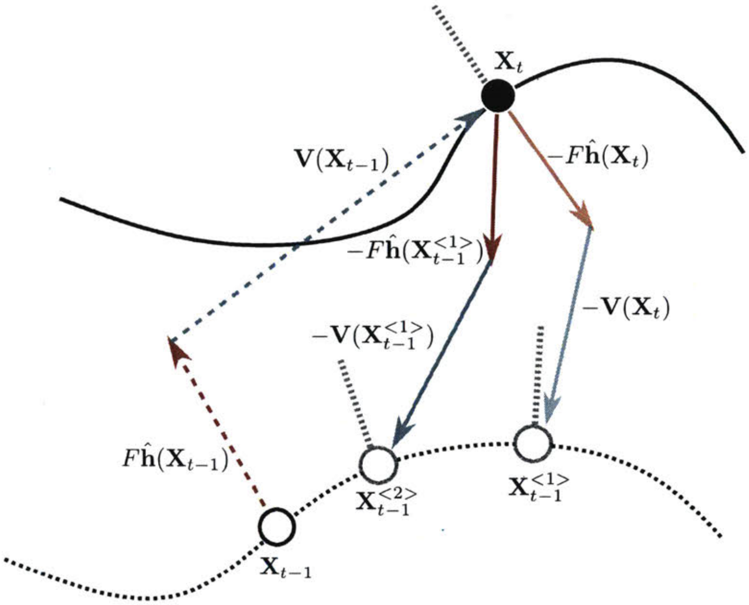

$\mathbf{X}_t$
$\mathbf{V}(\mathbf{X}_{t-1})$
$-F\hat{\mathbf{h}}(\mathbf{X}_t)$
$-F\hat{\mathbf{h}}(\mathbf{X}_{t-1}^{<1>})$
$-\mathbf{V}(\mathbf{X}_t)$
$-\mathbf{V}(\mathbf{X}_{t-1}^{<1>})$
$F\hat{\mathbf{h}}(\mathbf{X}_{t-1})$
$\mathbf{X}_{t-1}^{<2>}$
$\mathbf{X}_{t-1}^{<1>}$
$\mathbf{X}_{t-1}$

Figure 3-3: Implicit backtracking scheme. The forward Euler backtracking scheme is discussed as an example. First, the optimal heading and the external velocity at point $\mathbf{X}_t$ and at time t (faint red and cyan arrows respectively) are used to reach the point $\mathbf{X}_{t-1}^{<1>}$. The normal to the zero level set at $\mathbf{X}_{t-1}^{<1>}$ at time $t-1$ is then computed (dotted gray line at $\mathbf{X}_{t-1}^{<1>}$), and this is used as the new optimal heading to move from $\mathbf{X}_t$ to $\mathbf{X}_{t-1}^{<2>}$. This process is repeated until the required convergence criterion is met.

the current solution (Ferziger et al., 1997). We use the following stopping criterion:

$$\text{Stop if} \quad |\mathbf{X}^{<k+1>} - \mathbf{X}^{<k>}| \leq \varepsilon \tag{3.25}$$

where ε is a small number, typically of the order of the smallest cell size.

Forward Euler Backtracking

The first scheme we obtain is first order accurate, and mimics the forward Euler scheme for the evolution of the level set function. It also serves as a building block towards the higher order schemes, similar to the results in §(3.3). The scheme is given by:

$$\mathbf{X}_p^{<k+1>} = \mathbf{X}_{p+1} - \Delta t \cdot \left(F\hat{\mathbf{h}}(\mathbf{X}_p^{<k>}, p\Delta t) + V(\mathbf{X}_p^{<k>}, p\Delta t) \right) \tag{3.26}$$

with $\mathbf{X}_p^{<0>} = \mathbf{X}_{p+1}$, and the stopping criterion given by equation (3.25).

As we are solving backwards in time, the value of $\mathbf{X}_{p+1}$ is known a priori. The normal $\hat{\mathbf{h}}(\mathbf{X}_p^{<k>}, t)$ is computed by the method described above. Note that, in general $\mathbf{X}_p^{<k>}$ may not lie on the zero level set of $\phi(\mathbf{x}, p\Delta t)$. In such cases, the normal is computed by taking the projection of this point onto the closest constituent triangle of the zero level set surface.

For convenience in the forthcoming parts, we rewrite equation (3.26) in terms of an operator $\mathcal{B}$ as:

$$\mathbf{X}_p^{<k+1>} = \mathbf{X}_{p+1} - \Delta t \cdot \mathcal{B}(\mathbf{X}_p^{<k>}, p\Delta t) \tag{3.27}$$

where,

$$\mathcal{B}(\mathbf{X}_p^{<k>}, p\Delta t) = \left(F\hat{\mathbf{h}}(\mathbf{X}_p^{<k>}, p\Delta t) + V(\mathbf{X}_p^{<k>}, p\Delta t)\right) \tag{3.28}$$

TVD RK2 Backtracking

Equation (3.29) describes the new implicit backtracking scheme mimicking the TVD RK2 scheme for forward evolution.

$$\mathbf{X}_p^{<k+1>} = \mathbf{X}_{p+1} - \Delta t \cdot \mathcal{B}\left(\mathbf{X}_p^{<k>} + \frac{1}{2}\Delta t \cdot \mathcal{B}\left(\mathbf{X}_p^{<k>}, p\Delta t\right), \left(p + \frac{1}{2}\right)\Delta t\right) \tag{3.29}$$

This scheme may be initiated similarly to the forward Euler backtracking scheme with $\mathbf{X}_p^{<0>} = \mathbf{X}_{p+1}$, along with the stopping criterion given by equation (3.25). Note that $\mathcal{B}$ is the Euler building block, and two computations of this function are required per iteration, thus doubling the computational cost per iteration as compared to forward Euler backtracking, that is equation (3.26).

TVD RK3 Backtracking

Finally, we obtain the backward tracing scheme corresponding the TVD RK3 forward evolution scheme. Equation (3.31) and equation (3.33) describe the iterative equations for the backward tracing of the optimal path using TVD RK3 scheme.

Referring to to the discussion on time marching in section §(B.4), it can be seen that $\phi(\mathbf{x}, t + \Delta t)$ is computed using equation (B.40). Hence, the optimal trajectory backtracking equation corresponding to this step becomes:

$$\mathbf{X}_{p+1} = \frac{1}{3}\mathbf{X}_p + \frac{2}{3}\mathbf{X}_{p+\frac{3}{2}} \tag{3.30}$$

Rearranging to compute the unknown ($\mathbf{X}_p$), we get:

$$\mathbf{X}_p = 3\mathbf{X}_{p+1} - 2\mathbf{X}_{p+\frac{3}{2}} \tag{3.31}$$

Note that as both the quantities on the right hand side are after time $p\Delta t$, they are known. Hence, using equation (3.31), we can compute $\mathbf{X}_p$.

Note that, even though we obtain $\mathbf{X}_p$ without an iterative solve, we require the value of $\mathbf{X}_p$ at intermediate times (for example, we require $\mathbf{X}_{p+\frac{3}{2}}$ to compute $\mathbf{X}_p$). Hence, along with $\mathbf{X}_p$, we also need to compute the value of $\mathbf{X}_{p+\frac{1}{2}}$ at every step. This requires an iterative solve.

Equation (B.39) implies that $\mathbf{X}_{p+\frac{1}{2}}$ can be written as:

$$\frac{\mathbf{X}_{p+\frac{3}{2}} - \mathbf{X}_{p+\frac{1}{2}}}{\Delta t} = \left(F\hat{\mathbf{h}}\left(\mathbf{X}_{p+\frac{1}{2}}, \left(p + \frac{1}{2}\right)\Delta t\right) + V\left(\mathbf{X}_{p+\frac{1}{2}}, \left(p + \frac{1}{2}\right)\Delta t\right)\right) \tag{3.32}$$

Invoking the notation from equation (3.28), this can be written as:

$$\mathbf{X}_{p+\frac{1}{2}}^{<k+1>} = \mathbf{X}_{p+\frac{3}{2}} - \Delta t \cdot \mathcal{B}\left(\mathbf{X}_{p+\frac{1}{2}}^{<k>}, \left(p + \frac{1}{2}\right)\Delta t\right) \tag{3.33}$$

Equation (3.31) and equation (3.33) complete our backtracking scheme corresponding to the TVD RK3 forward evolution. The stopping criterion used is the same as the schemes above, and is described in equation (3.25). In order to initialize this scheme, we set $\mathbf{X}_{p+\frac{1}{2}}^{<0>} = \mathbf{X}_{p+\frac{3}{2}}$, similar to the earlier schemes. Equation (3.31) is not iterative, and hence does not require any initialization.

3.5.3 Backtracking for Vehicles with Deterministic Constrained Motion

As discussed in chapter 2, the movement of the vehicles with deterministic motion along some constraint can be decomposed into two components – the motion along this constrained direction and the motion orthogonal to this constraint direction. While backtracking the optimal path, the contribution to the vehicle movement due to these two components is computed independently, and their vector sum yields the resulting trajectory. Although this is applicable to any general deterministic constrained motion of the vehicle, we focus on the vehicles whose vertical motion is known. In this case, we can split the forward evolution in two parts, given by equation (2.33) and equation (2.34). Similarly to this, the optimal trajectory of the vehicle can be also decomposed in two parts: the horizontal motion and the vertical motion.

The computation of the horizontal motion can be done by the algorithm mentioned above, in a consistent way to the forward time-marching used for the level set. Note that this computation will be in 2 dimensions, and the trajectory point obtained is actually the projection of the exact point onto the XY plane. The vertical position is then computed by adding the vertical displacement due to the vehicle velocity and the local flow velocity.

The effective vertical velocity can be computed explicitly, given by:

$$\frac{\mathbf{X}_t - \mathbf{X}_t^*}{\Delta t} = -\left(V_z(\mathbf{X}_t^*) + U_z(t)\right) \tag{3.34}$$

where $\mathbf{X}_t^*$ is the auxiliary position at time t when only the horizontal motion is accounted for. For numerical consistency, this can also be solved implicitly, as given by equation (3.35)

$$\frac{\mathbf{X}_t - \mathbf{X}_t^*}{\Delta t} = -\left(V_z(\mathbf{X}_t) + U_z(t)\right) \tag{3.35}$$

Similar to the earlier implicit schemes, this equation also needs to be solved iteratively, with fist guess as $\mathbf{X}_t^{<0>} = \mathbf{X}_t^*$, and a stopping criterion based on the relative error between the successive iterated values. Hence, by first computing the horizontal motion followed by the vertical motion, the optimal trajectory for vehicles with known vertical velocities can be computed.

3.5.4 Checking of Optimal Paths

Once the optimal paths are computed, then they are tested for accuracy. We do this in two ways, in order to ensure that the paths are accurate and feasible, as described previously in section §(3.1):

1. While computing the optimal trajectory backwards, ensure that the end point of the backward tracing is either at the start position or within some allowable error tolerance from the start position

2. Use the computed optimal headings to then forward evolve the optimal trajectory. Ensure that the end point of this trajectory is at the target position or within some allowable error tolerance from the target position

The paths that satisfy the above two checks are accepted, and if a path does not satisfy either of the criteria, then it is re-computed with higher accuracy.

Chapter 4

Dynamic Three-Dimensional Applications

This chapter completes the various applications of the exact time-optimal path planning equations and their novel numerical integration and implementation that were derived earlier. Results are showcased in both analytical and realistic three-dimensional flow-fields. We categorize the considered applications 3 types:

1. Applications to benchmark the numerical schemes and the individual terms contributing to the level set function evolution.

2. Applications that employ analytically known steady and unsteady flow-fields to model the various oceanic phenomenon.

3. Applications to the real oceanic flow-fields.

Each of these sets highlights specific features of the path planning algorithm and the novel numerical schemes that were obtained in chapters 2 and 3.

Benchmarking examples allow us to separately study, quantify and check the effects of the advection and optimal propulsion terms and compare them with the theoretical outcomes. We also look at the convergence rates of the various schemes used to approximate spatial gradients against the corresponding expected values.

This is then followed by some 3D analytical flow-fields that try to model some typical environmental velocity fields. Specifically, we study path planning in the following two analytical flows:

1. Unsteady periodic double gyre flow that models a wind driven double gyre in the real ocean

2. Steady Arnold-Beltrami-Childress (ABC) flow. This field is notable as a simple example of a fluid flow that can have chaotic trajectories.

These applications effectively depict the full 3 dimensional nature of our path planning methodology as well as the ability to account for velocity fields that are spatially and temporally variable.

The last set of examples looks at path planning for different autonomous vehicles in realistic ocean domains. The region that we consider is in the North Atlantic ocean, off the coast of New Jersey, USA. We discuss path planning in this domain for vehicles with isotropic speed, oceanic floats with constrained motion as well as gliders performing sinusoidal vertical motion. This final set of examples utilizes the ocean flow-fields produced by the MSEAS primitive equation model (Haley and Lermusiaux, 2010; Leslie et al., 2010; Haley et al., 2015) and demonstrates the unique features of our path planning algorithm and numerical schemes.

4.1 Benchmarking and Comparisons

This section deals with the benchmarking of the developed numerical schemes as well as understanding the individual effects of the advection and the optimal propulsion. We also look at 2 simple examples and compare the obtained results for the optimal travel time with their theoretical counterparts.

4.1.1 Convergence Tests

To study and exemplify the convergence of the spatial and temporal numerical schemes that were discussed in chapter 3. We first examine the performance of numerical schemes for the passive advection of a tracer by a divergence-free, deformational flow in a unit square basin (Lolla, 2016; Durran, 1999). The initial tracer concentration is a bell curve centered at $\mathbf{c} = (0.25, 0.25)$. The tracer field is advected forward for a fixed time duration, and then backward for the same duration. Ideally, in the absence of any numerical errors, the resulting tracer field should be exactly same as the one that we started with. Hence, the advected field is compared with the initial tracer field to quantify the effect of numerical errors. Grid resolution is varied in order to compute the order of accuracy of the spatial schemes. In all the simulations, TVD RK3 time marching is used with a small time step size in order to ensure that the temporal errors are much smaller than the spatial errors and that they minimally affect the order of accuracy computations.

The tracer ϕ is advected according to equation (4.1)

$$\frac{\partial \phi}{\partial t} + \mathbf{V} \cdot \nabla \phi = 0 \tag{4.1}$$

with initial condition:

$$\phi(\mathbf{x}, 0) = e^{-100(\mathbf{x}-\mathbf{c})^T(\mathbf{x}-\mathbf{c})} = e^{-100\left((x-0.25)^2+(y-0.25)^2\right)} \tag{4.2}$$

The velocity field $\mathbf{V}$ in equation (4.1) is analytically known, and is given by equation (4.3).

$$\mathbf{V}(x, y) = \left(\sin^2(\pi x) \cdot \sin(2\pi y), -\sin^2(\pi y) \cdot \sin(2\pi x)\right) \tag{4.3}$$

where $(x, y) \in (0, 1)^2$. It is clear that the velocity field is divergence-free and vanishes at the boundaries. This implies that the tracer always remains confined in the interior of the domain and hence the boundary conditions do not play any role.

Table 4.1 describes the various simulation parameters. We test the donor-cell (1^{st} order), the ENO (3^{rd} order) and the WENO (5^{th} order) schemes.

Figure 4-1 displays the initial tracer field, the tracer field after the forward advection is complete, then the tracer field after the backward advection is complete. All

Table 4.1: Swirl flow parameters

Parameter	**Value**
Time of forward / backward advection	1
Time step size	5×10^{-4}
Considered grid resolutions	128×128
	256×256
	512×512
	1024×1024

these demonstrative results are for a 128×128 grid. It is quite clear that the 1^{st} order donor-cell scheme introduces heavy diffusion, whereas the 5^{th} order WENO scheme is able to completely preserve the initial tracer field.

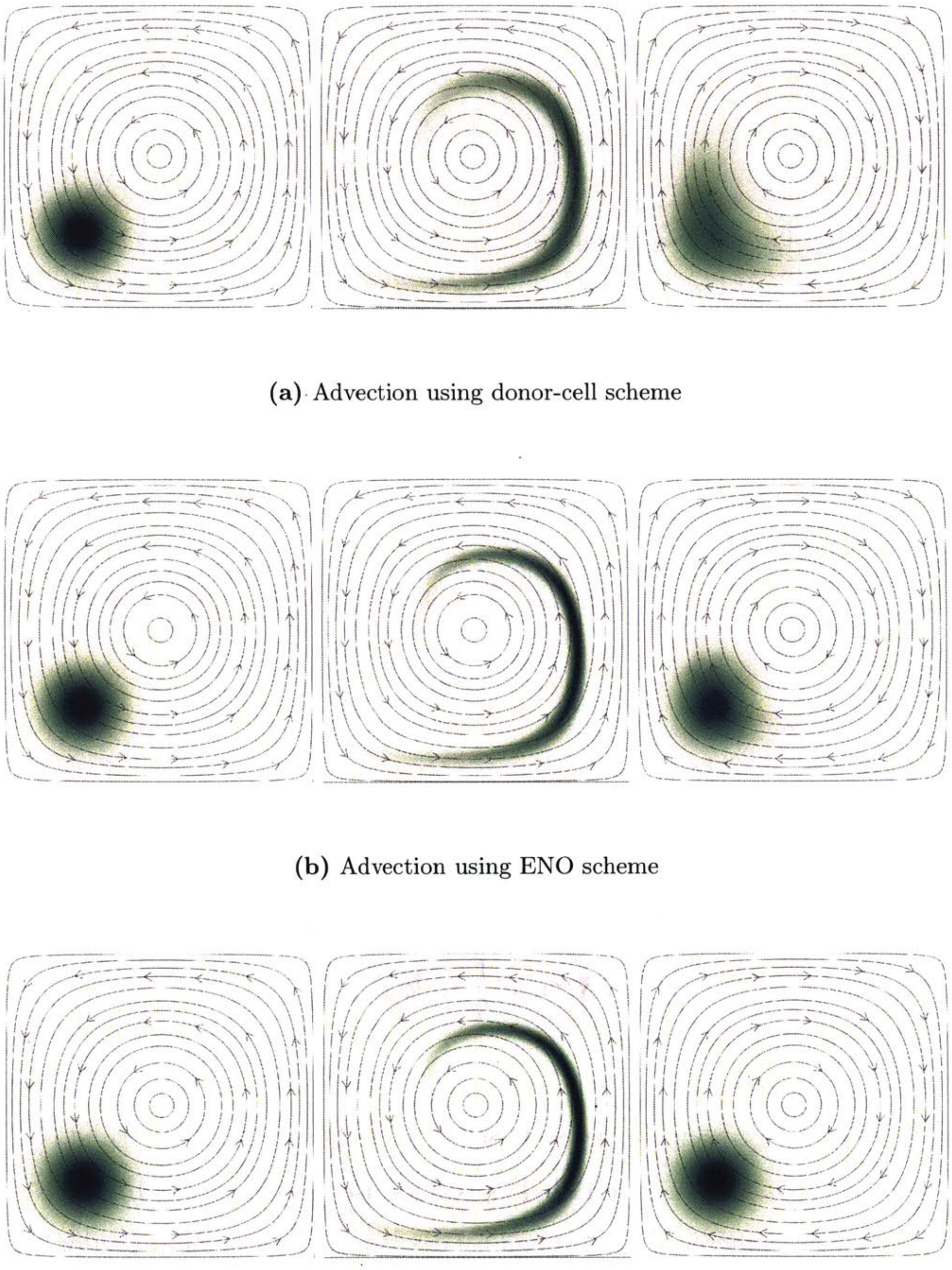
Figure 4-2 shows the relative error versus the grid resolution on a log-log scale, for all the 3 advection schemes. Lines with slope 1, 3 and 5 are also plotted for reference. From this plot, it is quite clear that the donor-cell scheme converges at 1^{st} order, the ENO scheme at 3^{rd} order and the WENO scheme at 5^{th} order. The exact orders of accuracy for these schemes (as computed from the error values) are 0.9979, 2.9649 and 4.9543 respectively. This serves as a validation for our advection schemes and verification of their implementations..

In order to test our time marching schemes, we consider the following ODE in **X**:

$$\frac{d\mathbf{X}}{dt} = \cos(2\pi t) \tag{4.4}$$

Subject to the initial condition:

$$\mathbf{X}(0) = 0 \tag{4.5}$$



(a) Advection using donor-cell scheme

(b) Advection using ENO scheme

(c) Advection using WENO scheme

Figure 4-1: Swirl flow test case for advection schemes. In the absence of numerical errors, the final field should be the same as the initial field. Upwind scheme introduces significant diffusion. ENO scheme performs well, however some errors are still observed. WENO scheme almost perfectly conserves the field and minimal errors are introduced.

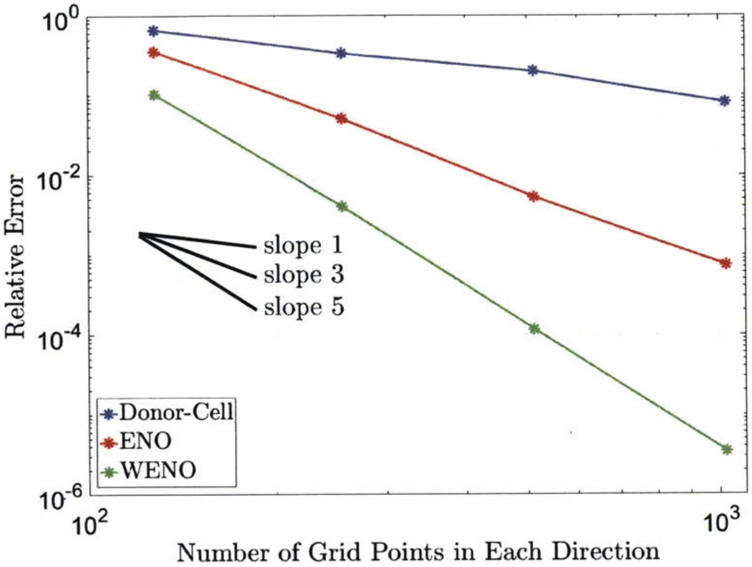


Figure 4-2: Order of accuracy for spatial schemes. It is seen that the donor-cell scheme converges at 1^{st} order, the ENO scheme converges at 3^{rd} order and the WENO scheme converges at 5^{th} order.

The exact solution for this ODE is given by equation (4.6)

$$\mathbf{X}(t) = \frac{1}{2\pi}\sin(2\pi t) \tag{4.6}$$

We now solve equation (4.4) using forward Euler, TVD RK2 and TVD RK3 methods to study the order of accuracy of each of these methods. The start time is $t_s = 0$ and the end time is $t_f = 1$. The time step values used are $10^{-2}, 5 \times 10^{-3}, 2.5 \times 10^{-3}$ and 10^{-3}. We plot the error norms against the time step size on a log-log plot, as shown in figure 4-3. Reference lines with slope 1, 2 and 3 are also plotted. The orders of accuracy of the forward Euler, the TVD RK2 and the TVD RK3 schemes as computed from the plotted data are 0.9964, 1.9923 and 2.9988 respectively. That is, the forward Euler scheme converges at 1^{st} order, the TVD RK2 at 2^{nd} order and the TVD RK3 at 3^{rd} order, which serves as validation of the time-marching schemes.

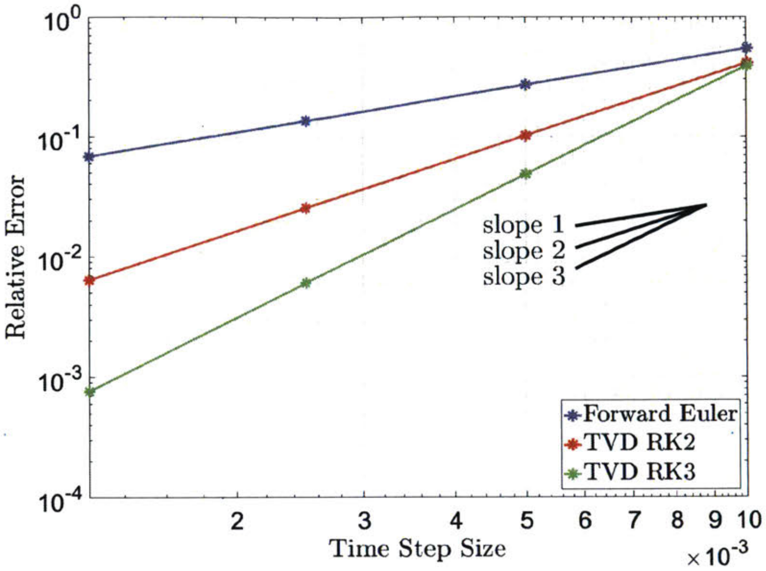


Figure 4-3: Order of accuracy of temporal schemes. It can clearly be seen that forward Euler scheme converges at first order, TVD RK2 converges at second order and TVD RK3 converges at third order.

4.1.2 No Flow

We now look at path planning when the advective velocity field is absent. That is, $\mathbf{V}(\mathbf{x},t) = 0$. In such cases, the Hamilton-Jacobi level set equation (assuming isotropic speed vehicle) becomes:

$$\frac{\partial \phi}{\partial t} + F|\nabla \phi| = 0 \tag{4.7}$$

With the initial condition $\phi(\mathbf{x},0) = |\mathbf{x} - \mathbf{x}_s| - r_0$ (signed distance with respect to a small sphere of radius r_0 centered at $\mathbf{x}_s$). As discussed in section §(3.4), in the absence of an external velocity field the signed distance property of ϕ is maintained by the Hamilton-Jacobi equation (2.29). That is, equation (4.7) ensures that $|\nabla \phi(\mathbf{x},t)| = 1$ at all times, given that $|\nabla \phi(\mathbf{x},0)| = 1$. Substituting this in equation (4.7), we have:

$$\frac{\partial \phi}{\partial t} + F = 0 \tag{4.8}$$

That is,

$$\phi(\mathbf{x}, t) = \phi(\mathbf{x}, 0) - F \cdot t \tag{4.9}$$

$$\phi(\mathbf{x}, t) = |\mathbf{x} - \mathbf{x}_s| - (r_0 + F \cdot t) \tag{4.10}$$

This means that the zero level set of ϕ at time t will form a sphere of radius $r = r_0 + F \cdot t$, centered at $\mathbf{x}_s$. The travel time to the destination position $\mathbf{x}_f$ can be computed as:

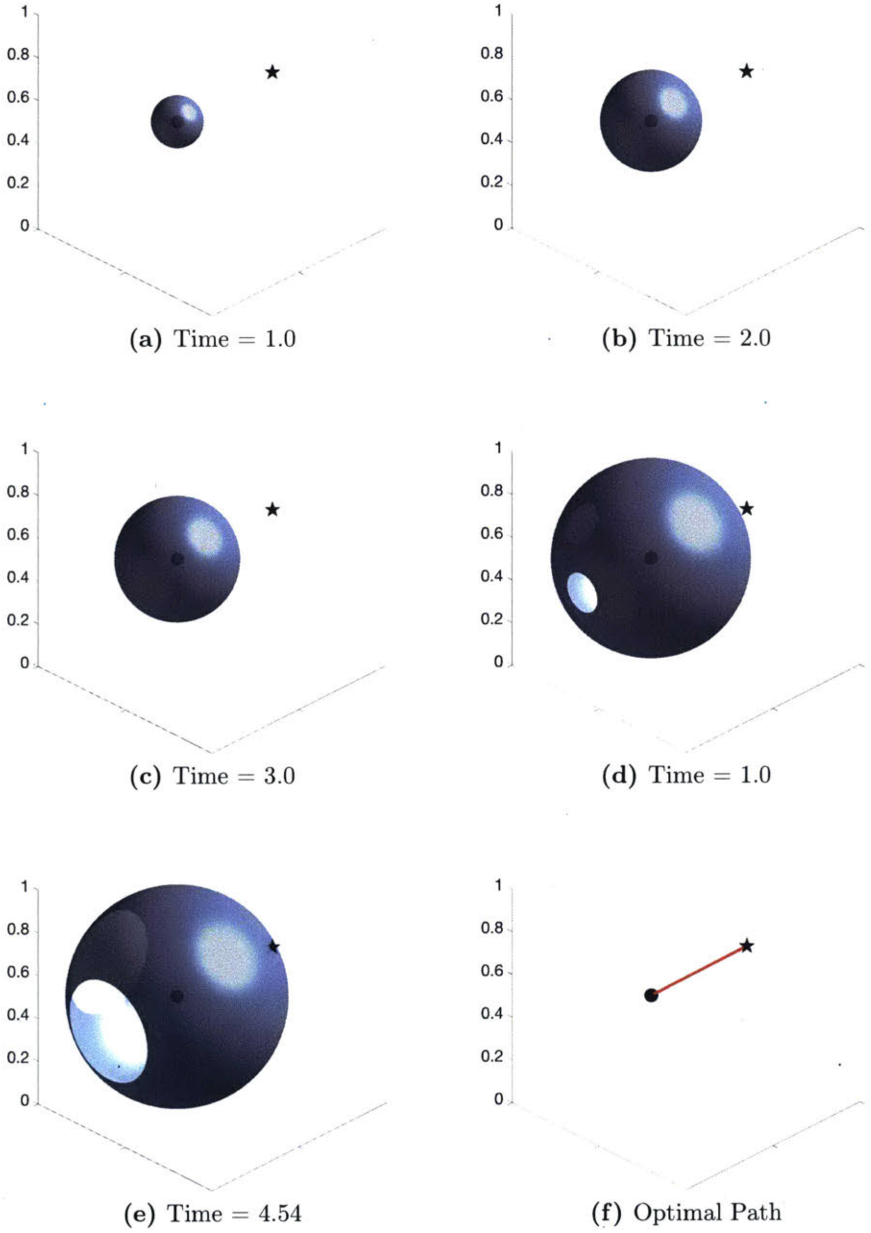
$$\phi(\mathbf{x}_f, \mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0)) = 0 \implies |\mathbf{x}_f - \mathbf{x}_s| = r_0 + F \cdot \mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0) \tag{4.11}$$

Hence,

$$\mathbf{T}(\mathbf{x}_f : \mathbf{x}_s, 0) = \frac{|\mathbf{x}_f - \mathbf{x}_s| - r_0}{F} \tag{4.12}$$

Equation (4.10) implies that the zero level set evolves as a sphere and the travel time between the start and the target positions is given by equation (4.12). This analytical expression is used to compare the obtained arrival time from the simulation. It can be easily shown that the optimal path is a straight line joining the start and the end positions, as the normal direction always points in the radial direction (although, this optimal path starts only from the surface of the initial zero level set sphere, a a small distance r_0 away from the start position).

Table 4.2 discusses the simulation parameters and the other details. We use the WENO scheme for spatial terms and TVD RK3 scheme for time marching, along with the radiation boundary conditions at all boundaries. Figure 4-4 shows the evolution of the zero level set at intermediate times and also the optimal path. For the parameter values considered, the theoretical optimal travel time is 4.537. From the simulation results, the computed travel time is 4.540. That is the error in the computed time is 0.041%, which is minimal and can be attributed to the time step size.



(a) Time = 1.0

(b) Time = 2.0

(c) Time = 3.0

(d) Time = 1.0

(e) Time = 4.54

(f) Optimal Path

Figure 4-4: Zero level set evolution and optimal path for when the advective velocity field is absent. The zero level set is represented by the shaded blue surface, the start position is represented by a black circle (always inside the zero level set) and the target position is denoted by a black star. The level set evolution stops when the zero level set just crosses the destination position. Optimal path between the start and the end positions is shown in red, and is the straight line joining these two points.

Table 4.2: Simulation parameters for path planning in the absence of advective field

Parameter	**Value**
Domain size	$(0,1)^3$
Start position ($\mathbf{x}_s$)	$(0.4, 0.4, 0.5)$
Target position ($\mathbf{x}_f$)	$(0.7, 0.65, 0.75)$
Vehicle speed (F)	0.1
Grid resolution	$100 \times 100 \times 100$
Time step size	5×10^{-3}
Initial zero level set radius (r_0)	0.01
Theoretical optimal travel time	4.537

4.1.3 Uniform Velocity Field

We now look at the case when the advective velocity field is uniform and constant. That is, when $\mathbf{V}(\mathbf{x}, t) = \mathbf{V}_0 = (u_0, v_0, w_0)$ for all $\mathbf{x}$ and t. In this case as well, a theoretical formulation for the optimal time can be obtained, as described in the following work. We compare this theoretical optimal time with the simulation result in order to study the accuracy of the path planning PDE simulation.

From equation (3.21), it can be seen that even in the case of constant and uniform velocity, ϕ remains a signed distance field, given that it was a signed distance field at time $t = 0$. Let us denote the theoretical optimal time by T. In order to compute the optimal travel time, we consider the inertial reference frame moving with velocity $\mathbf{V}_{frame} = -\mathbf{V}_0$. Value of the advective velocity field ($\mathbf{V}_0$ in the stationary frame) in this frame will be $\mathbf{V}^* = \mathbf{V} + \mathbf{V}_{frame} = 0$. That is, the advective velocity field is absent in this frame. However, effective target position will now be $\mathbf{x}_f^* = \mathbf{x}_f - T \cdot \mathbf{V}_0$. That is, the vehicle motion alone will need to account for the transport of the vehicle between the points $\mathbf{x}_s$ and $\mathbf{x}_f^*$. Hence, this reduces to the case discussed previously, with no advective velocity field. We know that the optimal travel time between $\mathbf{x}_s$ and $\mathbf{x}_f$ in such situations is given by equation (4.12). Substituting the specific values, we have:

$$T = \frac{|\mathbf{x}_f^* - \mathbf{x}_s| - r_0}{F} = \frac{|\mathbf{x}_f - \mathbf{x}_s - T \cdot \mathbf{V}_0| - r_0}{F} \tag{4.13}$$

$$(FT + r_0)^2 = |\mathbf{x}_f - \mathbf{x}_s - T \cdot \mathbf{V}_0|^2 \tag{4.14}$$

Let $\mathbf{x}_f - \mathbf{x}_s = \boldsymbol{\delta} = (\delta_x, \delta_y, \delta_z)$. Substituting this, we get the following quadratic equation, which can be solved to obtain the value of T.

$$(FT + r_0)^2 = (\delta_x - Tu_0)^2 + (\delta_y - Tv_0)^2 + (\delta_z - Tw_0)^2 \tag{4.15}$$

$$\left(F^2 - \left(u_0^2 + v_0^2 + w_0^2\right)\right) T^2 + 2\left(r_0F + \delta_x u_0 + \delta_y v_0 + \delta_z w_0\right) T = \left(\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right) - r_0^2\right) \tag{4.16}$$

This is concisely written as:

$$\left(F^2 - |\mathbf{V}_0|^2\right) T^2 + 2\left(r_0F + \boldsymbol{\delta} \cdot \mathbf{V}_0\right) T - \left(|\boldsymbol{\delta}|^2 - r_0^2\right) = 0 \tag{4.17}$$

The non-negative solution of this quadratic equation (which indeed is the optimal travel time) is given by:

$$T = \frac{-\left(r_0F + \boldsymbol{\delta} \cdot \mathbf{V}_0\right) + \sqrt{\left(r_0F + \boldsymbol{\delta} \cdot \mathbf{V}_0\right)^2 + \left(F^2 - |\mathbf{V}_0|^2\right)\left(|\boldsymbol{\delta}|^2 - r_0^2\right)}}{\left(F^2 - |\mathbf{V}_0|^2\right)} \tag{4.18}$$

When $F = |\mathbf{V}_0|$, equation (4.17) reduces to a linear equation, and in such cases,

$$T = \frac{|\boldsymbol{\delta}|^2 - r_0^2}{2\left(r_0|\mathbf{V}_0| + \boldsymbol{\delta} \cdot \mathbf{V}_0\right)} \tag{4.19}$$

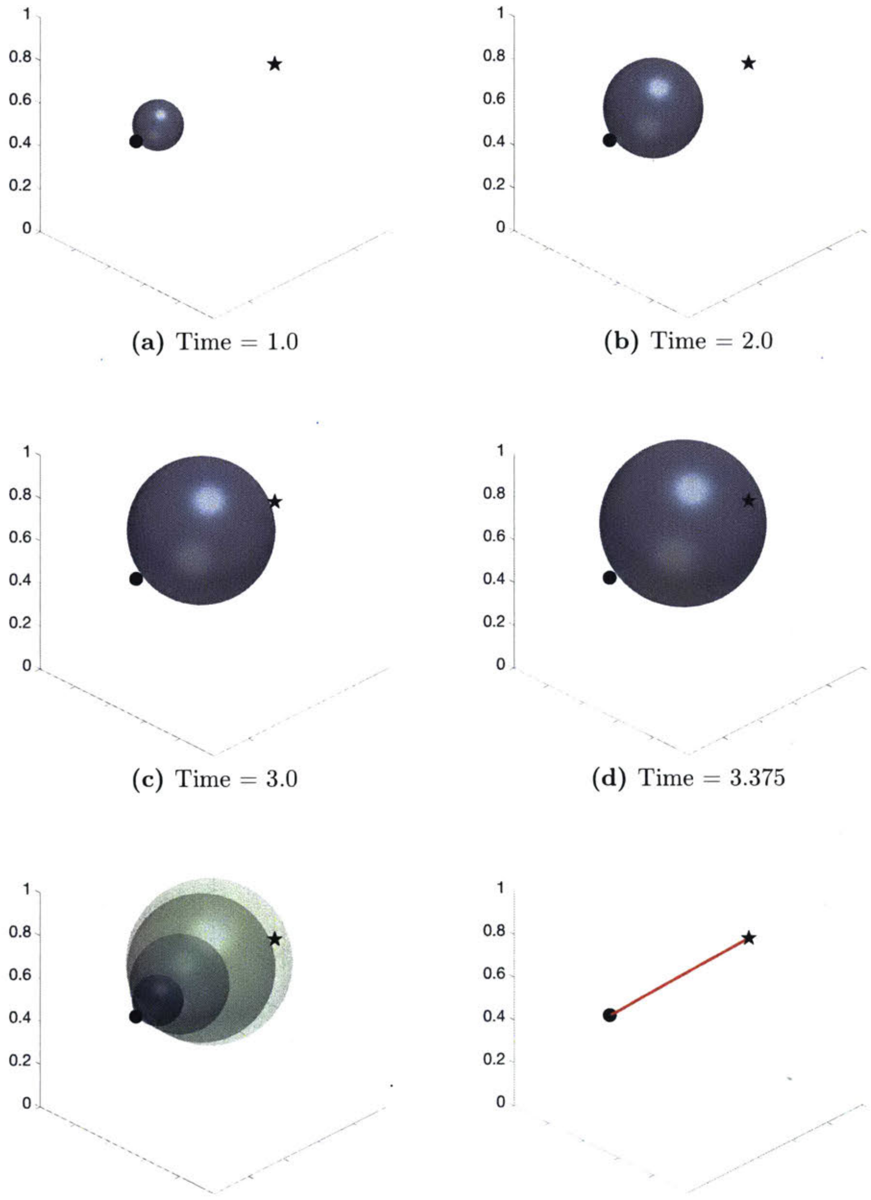
Equation (4.17) has a non-negative solution as long as $F \geq |\mathbf{V}_0|$. But when $F < |\mathbf{V}_0|$, the discriminant of equation (4.17) may become negative, and in such cases it is inferred that the destination position is not reachable from the start position for the given flow and vehicle parameters. Note that if $F \geq |\mathbf{V}_0|$, then all the positions in the domain are reachable from any start position. It can be shown that in such cases equation (2.29) reduces to a modified Eikonal equation (Lolla, 2016), and hence this property follows immediately.

This value of the theoretical optimal travel time can readily be computed as all the involved parameters are known. For the purposes of the simulation, we consider a similar setup as the previous case. The external velocity field is given by: $\mathbf{V}_0 = (0.050, 0.075, 0.065)$. Hence, $|\mathbf{V}_0| = 0.11 > F$, and the equation is not Eikonal). The start and the target positions are $\mathbf{x}_s = (0.25, 0.30, 0.40)$ and $\mathbf{x}_f = (0.70, 0.65, 0.80)$. Apart from these, all the other quantities bear the same values as given in table 4.2. The theoretical optimal time, computed using equation (4.18) for this setup is 3.377. The computed optimal time is 3.375, which is off from the true value by 0.029 %. This amount is extremely small, and can again be attributed to the size of the time step and other computational inaccuracies.

For uniform advective velocities, the zero level set grows as a sphere with radius $r = r_0 + F \cdot t$, and is also advected along the velocity direction. That is, the center of this sphere lies at $\mathbf{x}_c = t \cdot \mathbf{V}_0$. This is clearly seen from figure 4-5, where the zero level set evolves as a sphere, and moves along the direction of the velocity. It can be shown that the optimal path is a straight line joining $\mathbf{x}_s$ and $\mathbf{x}_f$ in this case as well.

4.2 Three-Dimensional Analytical Flow Fields

This section deals with the path planning for vehicles with isotropic motion in three-dimensional analytically known velocity fields. The two 3D flow fields that we consider are the analytical double gyre and the steady ABC flow. These examples are chosen as they are a first approximation of more complex oceanic flows. Both these examples effectively illustrate how the vehicle utilizes the advective velocity to its advantage in order to minimize travel time.



(a) Time = 1.0

(b) Time = 2.0

(c) Time = 3.0

(d) Time = 3.375

(e) Evolution of the zero level set

(f) Optimal Path

Figure 4-5: Zero level set evolution and optimal path for constant and uniform advective velocity field. The zero level set is represented by the shaded blue surface, the start position is represented by a black circle and the target position is denoted by a black star. The penultimate figure collectively plots the zero level sets at the considered times to help understand how the zero level set is affected by the external velocity field. Optimal path between the start and the end positions is shown in red, and is the straight line joining these two points.

4.2.1 Three-Dimensional Analytical Double Gyre – Shear Flow

We now look at the analytical double gyre flow, which is a periodic, unsteady and divergence-free velocity field given by Shadden et al. (2005); Froyland and Padberg (2009). Note that, this model is not a real fluid flow, in that it is not obtained as a solution to the Navier-Stokes' equations, but is a simplification of the double gyre pattern that frequently occurs in geophysical flows (Coulliette and Wiggins, 2001; Lolla et al., 2014b; Subramani et al., 2017c). We consider this flow in the domain $(0,2) \times (0,1) \times (0,1)$. The velocity field is analytically described by equation (4.20).

$$v_x = -\pi A \sin\left(\pi f(x)\right) \cos(\pi y) \qquad v_y = \pi A \cos\left(\pi f(x)\right) \sin(\pi y) \frac{df}{dx} \tag{4.20}$$

where,

$$f(x) = a(t)x^2 + b(t)x \tag{4.21}$$

$$a(t) = \epsilon \sin(\omega t) \tag{4.22}$$

$$b(t) = 1 - 2\epsilon \sin(\omega t) \tag{4.23}$$

The flow is completely determined by the parameters A, ϵ and ω. A determines the magnitude of the velocities, ω is the angular frequency of oscillations and ϵ approximately determines how far the line separating the gyres moves to the left or right, that is, the amplitude of motion of the separation line.

The double gyre is a 2 dimensional velocity field. In addition to the velocities given by equation (4.20), we consider a parabolic velocity profile for the Z direction that bears positive values in the region of the first gyre and negative values within the other gyre. Specifically, the Z velocity is given by:

$$v_z = xy(2-x)(1-y)\left(1 + \epsilon \sin(\omega t) - x\right) \tag{4.24}$$

Note that this z velocity still keeps the velocity field divergence free. The velocity field is qualitatively depicted in figure 4-6 and the simulation parameters are specified in table 4.3.

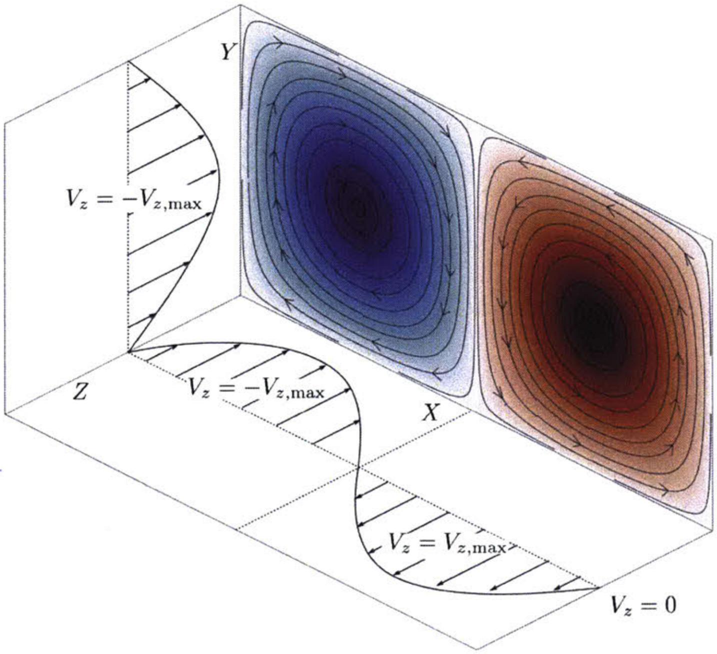
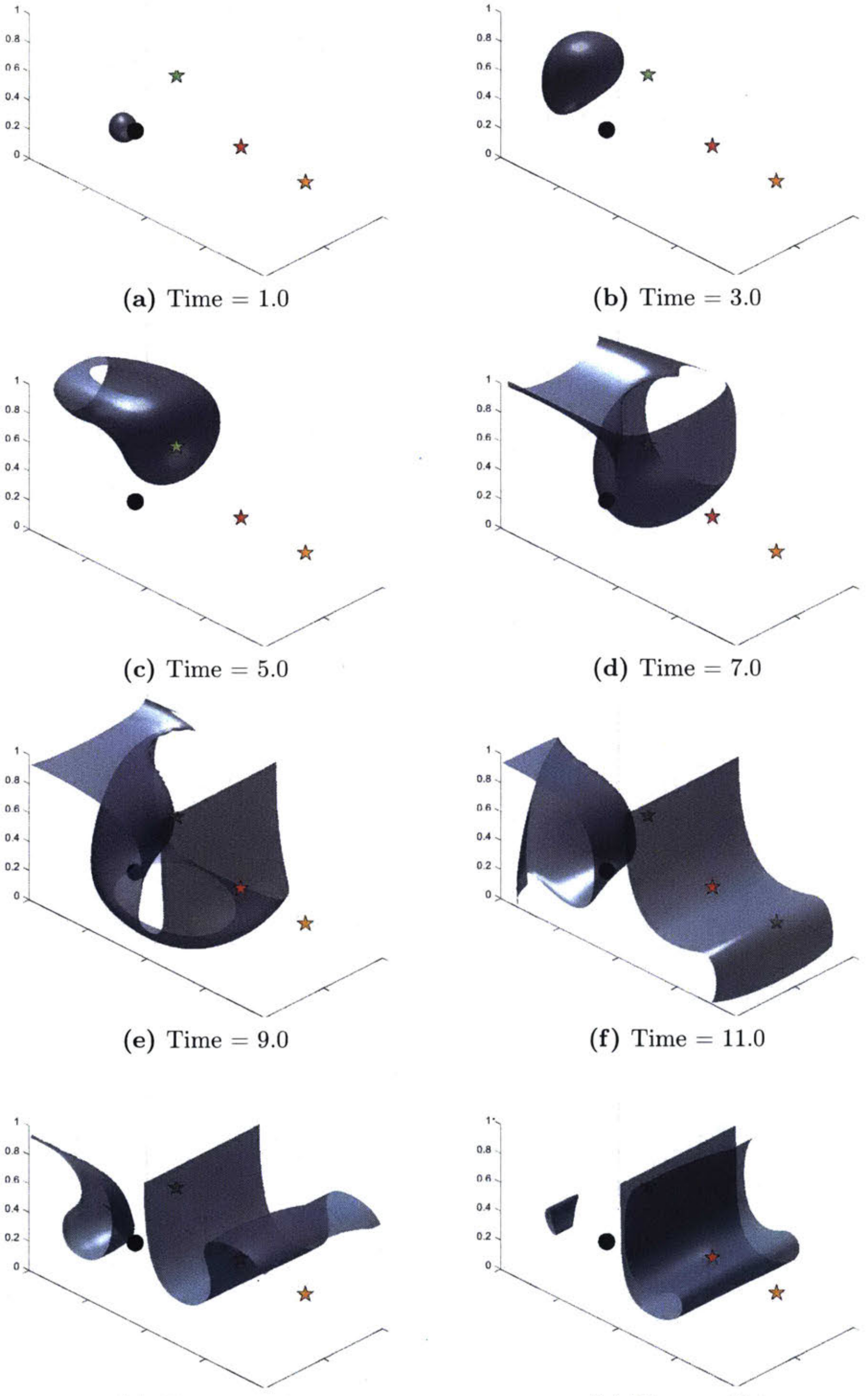


Figure 4-6: Three-dimensional analytical double gyre – shear flow velocity field. The analytical double gyre flow exists in the $X - Y$ plane, and its streamlines are overlaid on a vorticity plot for the same flow field. The Z direction has a double- parabolic velocity profile, whose projection is shown over the $X - Z$ and the $Y - Z$ planes.

Figure 4-7 shows the evolution of the zero level set from the start position until it crosses the considered target locations. This example demonstrates a one-to-all broadcast, where multiple vehicles leave from the same start point in order to travel to different target locations. Even though we have multiple vehicles, such a broadcast only requires one level set evolution per start point, and hence it is computationally advantageous. The zero level set is evolved from the start point until all the destination positions lie on or inside it. While the zero level set evolves, the time at which it crosses a particular destination position is noted, and in order to compute the optimal path, we backtrack from this stored time until the start time. Hence, for the present scenario, increasing the number of target positions only increases the cost of the backtracking ODE computation, which is much lesser than the computational expense of solving the forward evolution PDE.

Figure 4-8 shows the optimal paths to be followed by the vehicles in order to reach



(a) Time = 1.0

(b) Time = 3.0

(c) Time = 5.0

(d) Time = 7.0

(e) Time = 9.0

(f) Time = 11.0

(g) Time = 13.0

(h) Time = 15.0

Figure 4-7: Evolution of the zero level set in analytical double gyre – shear flow. The start position is marked in black, and the 3 colored stars represent the target positions. Radiation boundary conditions allow the level set to freely exit the domain. It crosses the green target at $t = 2.07$, the orange target at $t = 10.38$ and the red target at $t = 15$.

Table 4.3: Simulation parameters for path planning in analytical double gyre – shear flow

Parameter	**Value**
Domain size	$(0,2) \times (0,1) \times (0,1)$
Start position ($\mathbf{x}_s$)	$(0.4, 0.4, 0.5)$
Target position ($\mathbf{x}_f$)	$(0.7, 0.65, 0.75)$
A	0.1
ϵ	0.1
ω	$\pi/5$
Vehicle speed (F)	0.05
Grid resolution	$200 \times 100 \times 100$
Time step size	5×10^{-3}
Initial zero level set radius (r_0)	0.01

the respective destinations in the shortest time. It can clearly be seen that the vehicle utilizes the background flow field to its advantage, and even though it travels a much higher distance (as compared to a straight line path), it does so at a higher effective speed, thus reaching in the optimal time.

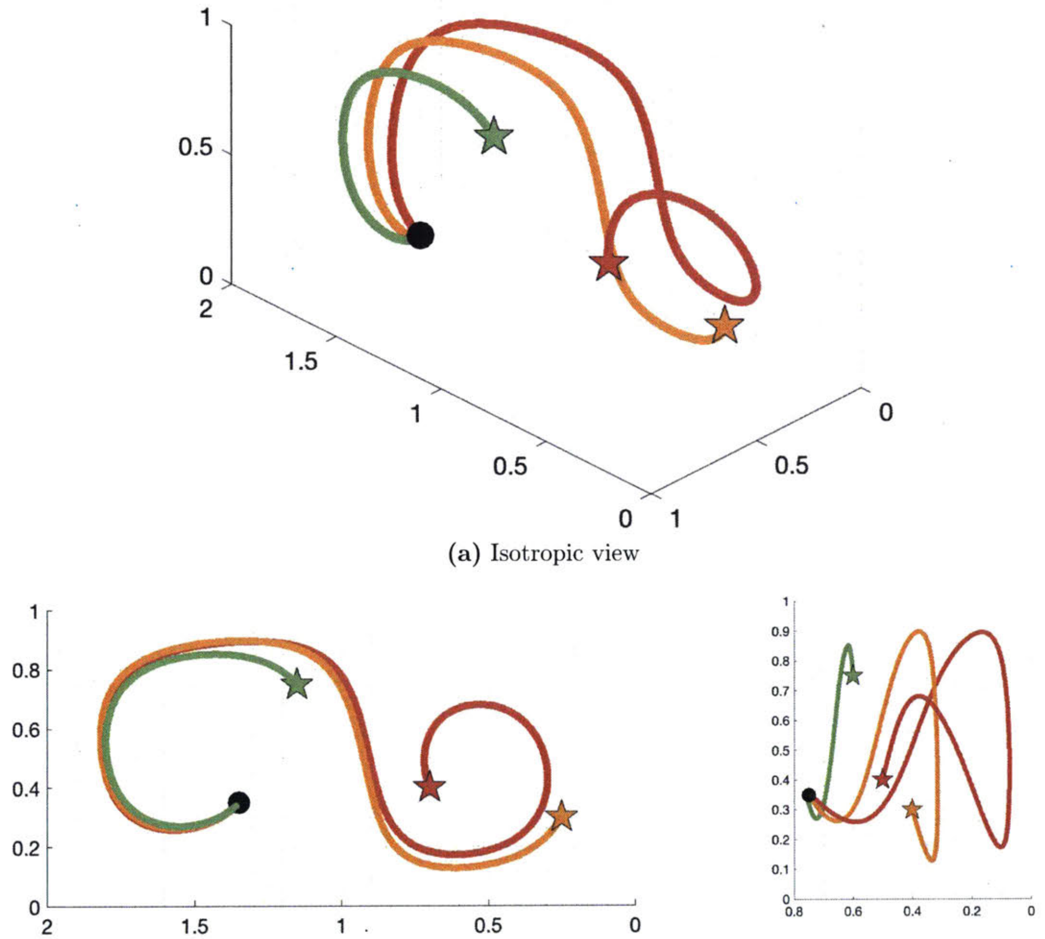
4.2.2 Three-Dimensional Steady ABC Flow

We now look at path planning in the steady Arnold-Beltrami-Childress (ABC) flow. This flow field is a 3 dimensional divergence-free velocity field which is an exact solution of Euler's equations. It is a notable case of a simple analytically described flow field with chaotic trajectories (Zhao et al., 1993). It is given by:

$$v_x = A \sin(z) + B \cos(y) \tag{4.25}$$

$$v_y = B \sin(x) + A \cos(z) \tag{4.26}$$

$$v_z = C \sin(y) + B \cos(x) \tag{4.27}$$



(a) Isotropic view

(b) Side view

(c) Front view

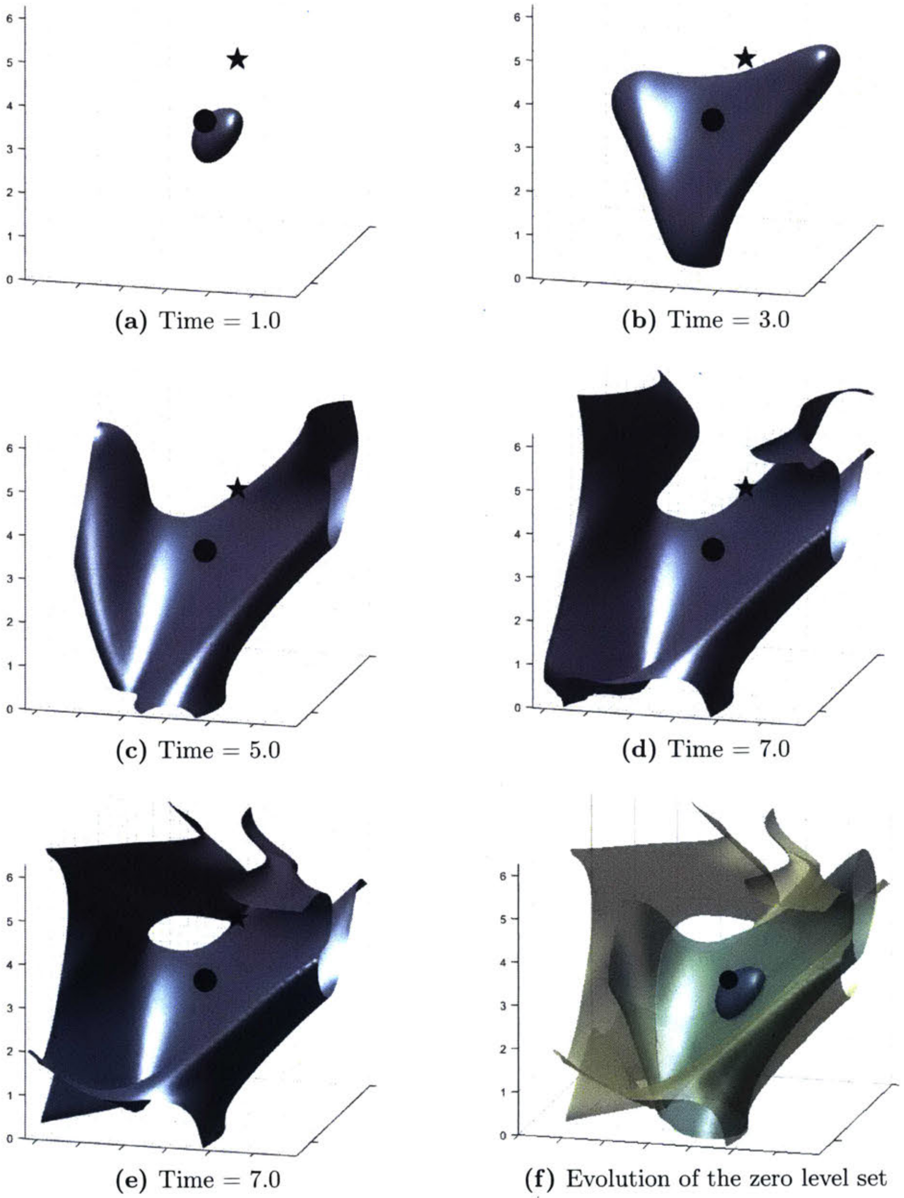
Figure 4-8: Optimal paths in the analytical double gyre – shear flow. The vehicles make optimal use of the gyres in order to reach their destination in the shortest amount of time. Even though the distance traveled by these vehicles is much larger than the straight line distance, this is done at a higher effective speed, thus reducing the required time.

We consider this flow over a domain of $(0, 2\pi)^3$. The simulation parameters as well as the start and the target positions are summarized in table 4.4. Figure 4-9 shows the evolution of the zero level set. The chaotic nature of the ABC flow is clearly visible through the zero level set evolution, wherein the surface is severely twisted, turned and warped as it evolves. The zero level set surface also merges multiple times with itself. All of these potential issues are handled satisfactorily by our numerical schemes, and the final result is an accurate zero level set evolution.

Table 4.4: Simulation parameters for path planning in steady 3D ABC flow

Parameter	**Value**
Domain size	$(0, 2\pi)^3$
Start position ($\mathbf{x}_s$)	$(3, 3, 3)$
Target position ($\mathbf{x}_f$)	$(2.7, 1.3, 4)$
A	$\sqrt{3}$
B	$\sqrt{2}$
C	1
Vehicle speed (F)	2
Grid resolution	$100 \times 100 \times 100$
Time step size	5×10^{-3}
Initial zero level set radius (r_0)	0.01
Optimal travel time	7.221

Figure 4-10 plots the optimal path as computed by the backtracking equation. It is clear that the straight line distance between the start and the end position is extremely small as compared to the actual distance traveled. This in turn implies that the vehicle maximally uses the advective velocity field to its advantage and travels this larger distance at a much higher average speed, thus requiring minimum journey time. Note that the vehicle travels very close to the domain boundary for some of its journey. As we implement open boundary conditions, if some part of the zero level set exits the domain, then it can never re-enter the domain. This implies that even



(a) Time = 1.0

(b) Time = 3.0

(c) Time = 5.0

(d) Time = 7.0

(e) Time = 7.0

(f) Evolution of the zero level set

Figure 4-9: Evolution of the zero level set in the steady 3D ABC flow. The start position is marked by a black circle, and black star represents the target. It can be seen that the zero level set is twisted, folded and stretched due to the background velocity field. Final figure shows the zero level sets at different times overlaid on each other. This makes the chaotic nature of the flow very clear.

though the paths may travel close to the boundary, our methodology correctly does not predict a path that leaves the domain at any point in time. This is because no velocity information is available outside of the domain, and if it was available, then, a larger domain should be used in the computations.

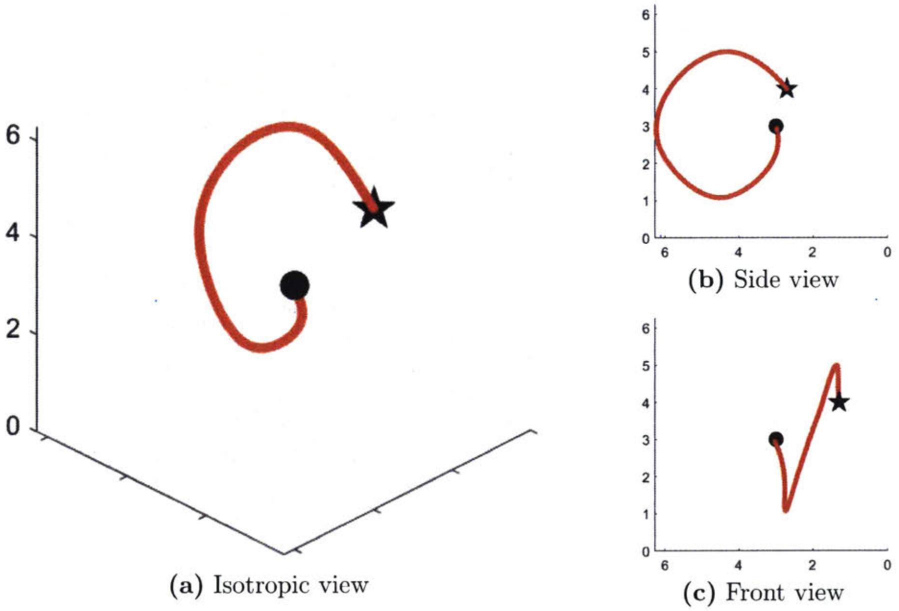


Figure 4-10: Optimal path for the 3D ABC flow. The optimal path is a curved trajectory, and the distance traveled by the vehicle is much larger than the straight line distance between the start and the target. This makes it clear that the vehicle maximally uses the advective velocity field to its advantage.

4.3 Realistic Four-Dimensional Ocean Flow Fields

In this section, we look at the time optimal path planning results for the different types of oceanic vehicles (as discussed in chapter 2) operating in realistic four-dimensional (time-space) ocean flow fields. The chosen region is in the Atlantic ocean, off the coast of New Jersey. This region is considered as it offers a variety of multi-scale oceanic flow features, such as the Gulf Stream, shelfbreak front, coastal jets, tidal currents and also the inflow from the Hudson river. Such conditions provide challenging environments for planning of autonomous missions. Figure 4-11 shows the simulation domain overlaid on a bathymetry plot of the region. This considered domain spans from 37.56°N to 40.99°N and from 73.97°W to 70.54°W. The white circle and the star denote the start and the end position for our planning missions respectively (both at the surface).

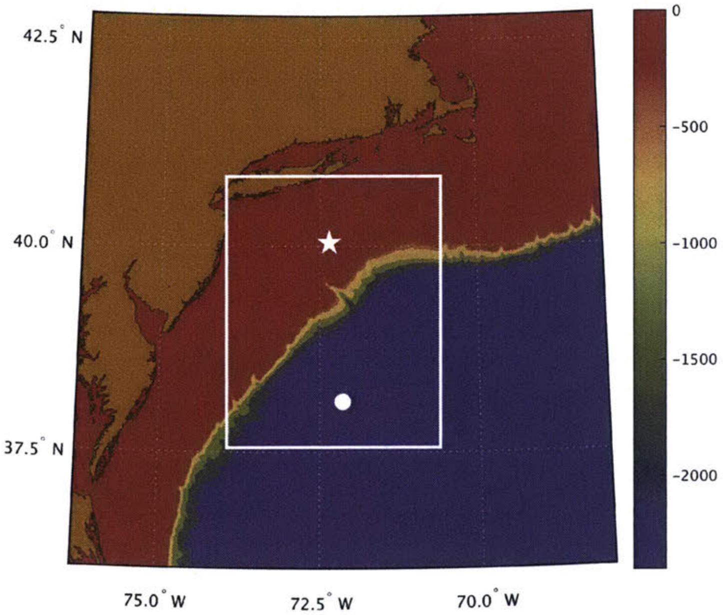


Figure 4-11: Chosen open domain for realistic path planning missions. The white rectangle denotes the domain, overlaid on a plot of the bathymetry of the local region, wherein the shelf-break region can clearly be seen. The circle marks the start position which is northwest of the Gulf Stream meander whereas the star marks the target position of the mission, just north of the Hudson Canyon.

The multi-scale ocean flow dynamics in this region is predicted using the MIT Multidisciplinary Simulation Estimation and Assimilation System (MSEAS) (Haley and Lermusiaux, 2010). The MSEAS software is used for fundamental research as well as realistic applications such as monitoring (Lermusiaux, 2007), real-time acoustic predictions (Beşiktepe et al., 2003; Xu et al., 2008) and environmental management (Cossarini et al., 2009).

The ocean fields used are from the data-assimilative simulations of the considered region during the period from August 28, 2006 up to September 9, 2006 (Haley et al., 2015; Subramani et al., 2017a; Chapman and Lynch, 2010). These simulations were carried out as a part of the Shallow Water '06 (SW06) initiative (WHOI, 2006; Lermusiaux et al., 2006; Newhall et al., 2007; Tang et al., 2007; Chapman and Lynch, 2010;

Lin et al., 2010). The MSEAS software solves the nonlinear free-surface hydrostatic primitive-equation (PE) model, configured with generalized-level vertical-coordinates and implicit two-way nested computational domains (Haley and Lermusiaux, 2010). In the horizontal, they have a 3 km and 1 km grid resolution, respectively, and in the vertical, they employ 100 levels optimized to the thermocline and flow structures.

The tidal-to-mesoscale ocean re-analysis is initialized with objectively-analyzed temperature, salinity, and velocity fields for Aug 14, 2006. Two multiscale-in-space analyses (Lermusiaux, 2002), inshore and offshore of the expected shelfbreak front, are combined using a shelfbreak-front feature model (Lermusiaux, 1999; Gangopadhyay et al., 2003). The Gulf Stream is initialized using synoptic and historical CTD profiles as well as estimates of its position based on SST and NAVOCEANO feature analyses. Barotropic tides based on the high-resolution TPXO7.2 surface-tide velocities and elevation (Egbert and Erofeeva, 2002) for Aug 14 2006 are merged with the subtidal initial fields, following (Haley et al., 2015). The re-analysis free-surface PE simulation is then integrated for 42 days. During integration, the ocean data collected during the AWACS and SW06 exercises as well as data of opportunity (NMFS, etc.) are assimilated. Finally, the numerical and sub-grid-scale parameters are tuned for the region by comparison of many PE simulations with independent in situ SW06 measurements.

In what follows, we first discuss path planning for the vehicles with isotropic motion, followed by the vehicles with anisotropic speeds. Specifically, we look at the motion of oceanic floats, which can move vertically by adjusting buoyancy, but are advected in the horizontal (Rudnick et al., 2004). Finally, we look at path planning for the vehicles whose vertical motion is known, by breaking down the problem in to a two-dimensional path planning problem, as discussed in section §(2.4).

4.3.1 Three-Dimensional Vehicles with Isotropic Speeds

The first 3D example that we consider is of a vehicle with isotropic speed traveling in the domain above. The start position of the mission is at $37.90°N, 72.22°W$. The vehicle travels to the target position located at $39.85°N, 72.30°W$, with a maximum speed of 50 m/s. As the vehicle speed is isotropic, it can travel freely in any of the 3D directions. As discussed in chapter 2, in order to be optimal in time, the vehicle has to always travel at the maximum possible speed (and consequently its heading is the only control parameter). The maximum allowable depth for the vehicle to travel at is 100 meters, which is well within the typical range of some oceanic gliders (Rudnick et al., 2004; Haley et al., 2009; Ramp et al., 2009).

We carry out our 3D path planning simulation on a grid of $128 \times 128 \times 50$ elements in the longitudinal, latitudinal and vertical directions respectively, where each grid cell has an approximate size of 2.32 km $\times$ 2.98 km $\times$ 2 m. We assume that the vehicle starts its journey on August 29 at 00:00:00 Z. A time step of 6 minutes (360 seconds) is used. The equations are non-dimensionalized according to the results obtained in section §(3.2). This non-dimensional form of the Hamilton-Jacobi equation is solved, and then the fluxes are multiplied by the appropriate dimensional constants to study the growth of the zero level set. We use the WENO scheme for the spatial terms, and the TVD RK3 for the time marching, which ensures an accurate evolution of the zero level set.

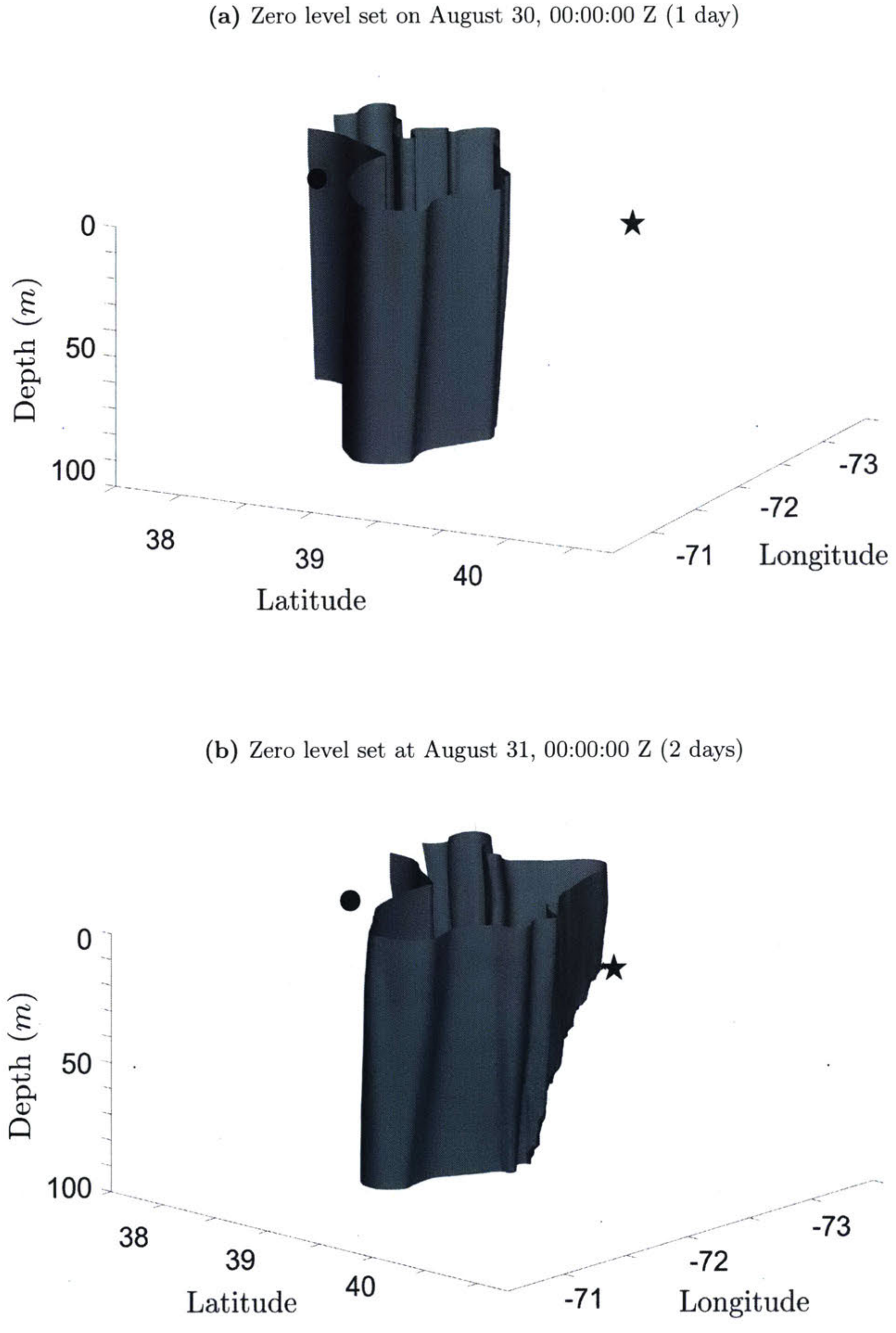
Figure 4-12 plots the growth of the zero level set (the reachability front) from the start position until it crosses the target. We observe that the zero level set quickly reaches the lower limit of the simulation domain, and then evolves as a curtain from there on. This curtain has some variations in the location along the vertical, and such creases can clearly be seen. This happens due to the following reasons: First, the flow in this region is mainly barotropic, dominated by eddies and wind-driven flows and with little or no reversal in the first 100 m depth. This means that the variation in the horizontal velocities along the vertical direction is much smaller than these velocities themselves. Further, there are very few locations where a reversal in the flow direction with depth in these top 100 m is observed. These phenomena prompt the zero level set to grow as curtains. Second, the scales in the graph are heavily skewed. The X and Y scales are in hundreds of kilometers, whereas the Z scale is in hundreds of meters. Even though the flow is primarily barotropic, there are smaller variations in the velocities along the depth. But these variations cause some shifts in the zero level set position along the depth which are further suppressed and made to appear smaller due to the skewed graph scales. It can be seen that the evolving zero level set is abruptly halted and it forms a jagged surface at some places in the lower region of the domain. This is because in these regions the zero level set hits the sea floor. We prohibit the zero level set from penetrating the sea floor (as the vehicle cannot move inside land). This is done by setting the advective velocity field $\mathbf{V}(\mathbf{x}, t)$ as well as the vehicle speed F equal to zero in these regions. This ensures that the optimal path never crosses through any land mass.

Figure 4-13 depicts the optimal path followed by the vehicle. The vehicle requires 3.84 days to travel along this optimal path. To compare, in the absence of any velocity field the vehicle would travel in a straight line joining the start and target positions and would require 4.97 days. Hence, it is clear that the vehicle makes efficient use of the oceanic velocities to reach the target in the shortest amount of time. In the initial phases of its travel, the vehicle tries to use the favorable flow of the Gulf Stream and hence travels slightly to the right. It also dives deeper in order to avoid the adverse currents at the surface. Later on in its journey, we can see that the vehicle is deflected eastwards due to the inflow from the Hudson river. Finally, the vehicle gradually rises up to the surface and reaches the destination position in optimal time.

4.3.2 Three-Dimensional Vehicles with Anisotropic Speeds

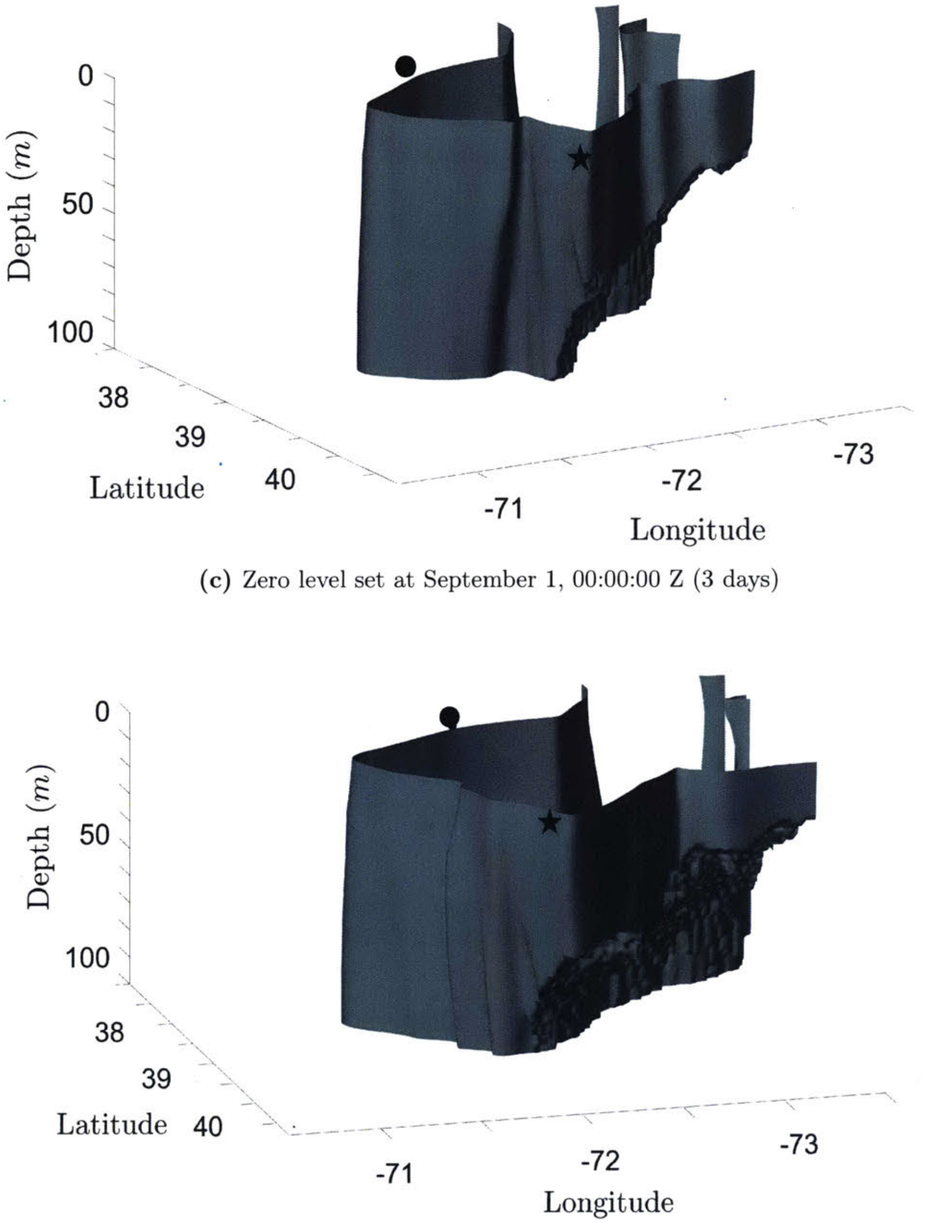
We now look at the path planning for vehicles with heading-dependent speeds and constrained motion. Particularly, we consider a vehicle which can travel freely in the vertical direction, but its motion in the horizontal plane is only through advection. Most oceanic floats are able to maneuver this way, wherein their vertical motion is a result of changing the effective buoyancy (Gould et al., 2004).

We assume that this float travels at 10 m/s vertically. The start and the end positions of the mission are the same as the previous case, with the start position being $37.90°N, 72.22°W$ and the target location being $39.85°N, 72.30°W$. The float is released on August 28, 00:00:00 Z, and is allowed to travel at a maximum depth of 100 meters. The simulations are carried on the same grid as described previously, with a



(a) Zero level set on August 30, 00:00:00 Z (1 day)

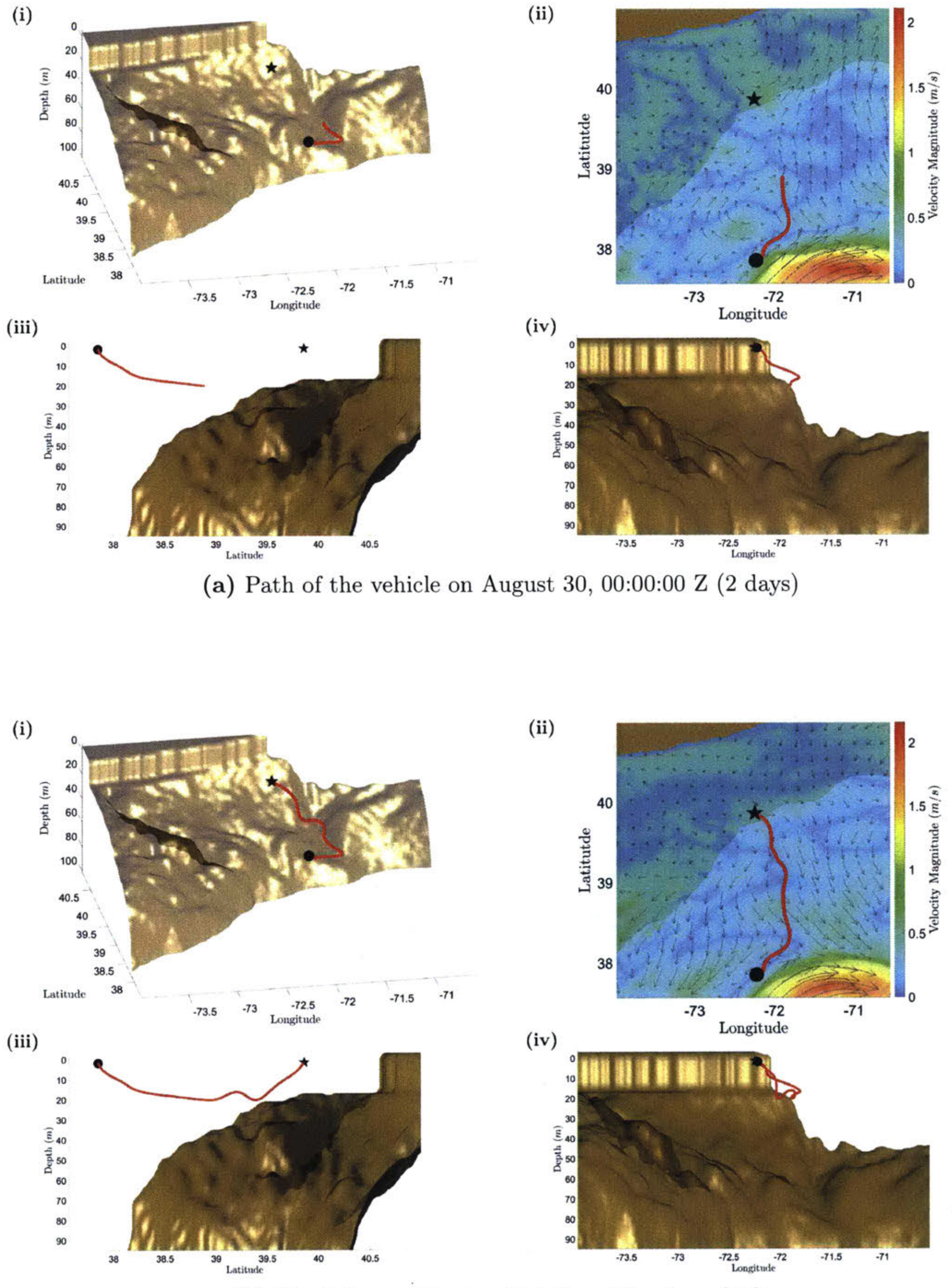
(b) Zero level set at August 31, 00:00:00 Z (2 days)



(c) Zero level set at September 1, 00:00:00 Z (3 days)

(d) Zero level set at September 1, 20:17:22 Z (3.84 days: end time)

Figure 4-12: Evolution of the zero level set in realistic ocean domain for a vehicle with isotropic speed, from August 30 to September 1, 2006. The start point of the mission is northwest of the Gulf Stream meander, and the target position is just north of the Hudson Canyon. The level set evolves as curtains due to the barotropic nature of the velocity field. Further, the level set growth is halted whenever it comes in contact with the sea-floor. This leads to the jagged surfaces in sub-figures (c) and (d). The level set evolution is shown from different viewpoints in order to better understand the global picture.



(a) Path of the vehicle on August 30, 00:00:00 Z (2 days)

(b) Final time optimal path followed by the vehicle

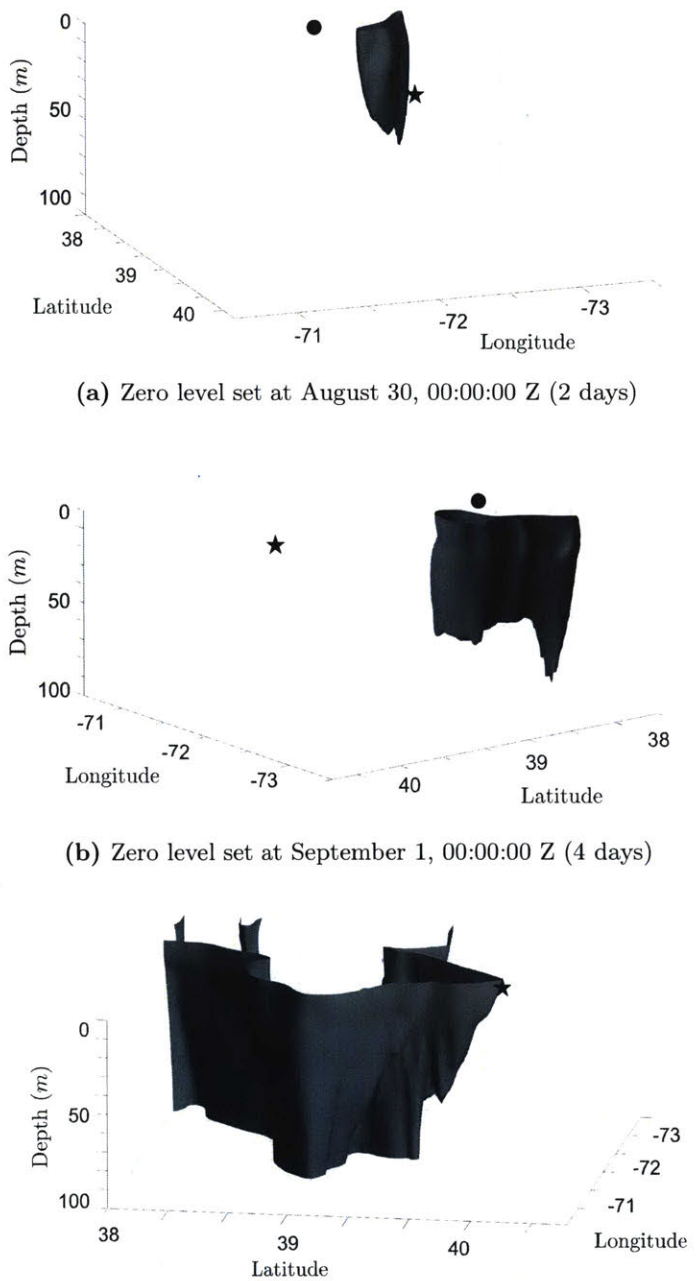
Figure 4-13: Time optimal path followed by a vehicle with isotropic speed in the mid-Atlantic bight region. Sub-figure (i) shows the isometric view of the optimal path. Sub-figures (ii), (iii), (iv) depict the top, side and front views of the path respectively. We also plot the instantaneous velocity field for reference in sub-figure (ii). The optimal paths are shown at an intermediate time and also at the final time. The vehicle initially tries to utilize the Gulf Stream, and is later pushed eastwards due to the flow from the Hudson Canyon. It also dives deeper in order to avoid the adverse currents at the ocean surface.

resolution of $128 \times 128 \times 50$, with the size of each cell being 2.32 km $\times$ 2.98 km $\times$ 2 m, and a time step of 6 minutes. The non-dimensional form of the Hamilton-Jacobi equation is solved using the WENO scheme for the spatial terms and the TVD RK3 time marching.

Note that, as this is a case of anisotropic path planning, a maximization needs to be carried out in order to compute the optimal propulsion term. In this specific case, we have two possible options. First, we can follow the methodology described in section §(3.3) in order to theoretically maximize the optimal propulsion term. Second, we can construct an offline maximization table that stores the optimal heading directions (and the corresponding optimal propulsion contribution) as a function of the level set normal direction. This approach is more general, and can be applied for cases when the description of the dependence of speed on the heading is not analytically known. The simulations are carried out using both the above mentioned approaches, and the results are seen to be the same (hence not shown here). Only the results utilizing theoretical maximization of the optimal propulsion term are discussed in this work.

Figure 4-14 shows the evolution of the zero level set for a float performing vertical motion. It is observed that the zero level set of the float evolves much slower than the one for the glider. The net travel time of the float is 9.53 days. This is expected, as the float does not have any horizontal velocity of its own. Similar to the previous case, it is observed that the growth of the zero level set locally stops as soon as it hits the sea floor. This ensures that the optimal path does not go through any land mass.

Sometimes, the optimal path may contain significant oscillations. This occurs because the optimal propulsion term is proportional to $\left|\frac{\partial\phi}{\partial z}\right|$. If this quantity is close to zero or if it frequently flips signs, then the corresponding path will contain oscillations. These oscillations may not always be physical and a float typically cannot respond to such drastic movements. A simple approach to eliminate such oscillations is to use a tolerance parameter. That is, we only consider the contribution due to the optimal propulsion term if the magnitude of $\left|\frac{\partial\phi}{\partial z}\right|$ is larger than some tolerance value τ. The value of τ controls the amount of oscillations in the optimal path, however setting this value to be too high can cause the path to deviate from optimality.

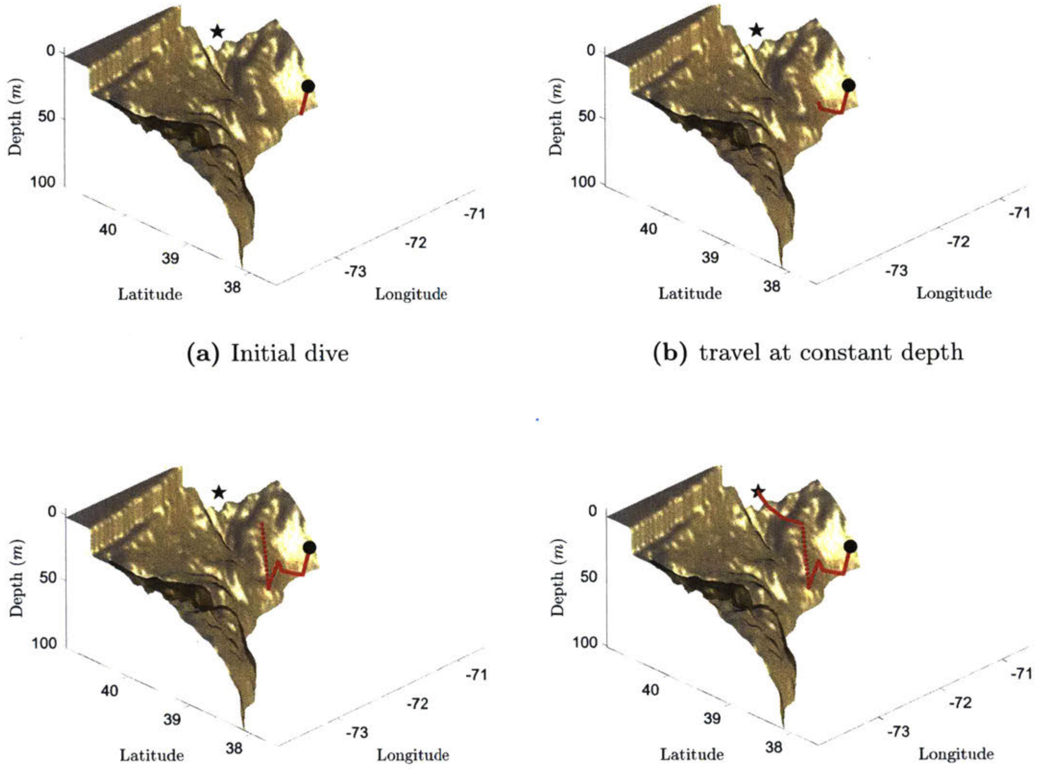


(a) Zero level set at August 30, 00:00:00 Z (2 days)

(b) Zero level set at September 1, 00:00:00 Z (4 days)

(c) Zero level set at Spetember 6, 12:40:02 Z (9.53 days: end time)

Figure 4-14: Evolution of the zero level set for float motion, from August 29 to September 6, 2006 in the mid-Atlantic bight region. The zero level set is observed to have greater variations with depth as compared to the isotropic vehicle motion case. Further, the evolution of the level set is comparatively slower, and it takes longer for the float to reach the destination. As the horizontal evolution of the reachability front is only due to oceanic flows, it evolves faster in the region of faster ocean currents. The level set evolution is shown from different viewpoints in order to better understand the global picture.



(a) Initial dive

(b) travel at constant depth

(c) Sharp dive and rise

(d) Travel at the surface

Figure 4-15: Time optimal path for a float operating in the realistic ocean, from August 30 to September 6, 2006 in the mid-Atlantic bight region. We observe that the float initially dives to a certain depth in order to avoid the adverse flow at the surface. It then travels at this constant depth for a while. After this, it dives and rises sharply and then concludes its journey by traveling at the ocean surface. Dives such as the one seen in sub-figure (c) can be eliminated by appropriately choosing the tolerance value.

Figure 4-15 shows the optimal path to be followed by the float. It can be seen that the float initially dives to a certain depth and then travels at this depth for a considerable time in order to avoid the adverse flow at the surface. It then dives deeper, and quickly rises to the surface to reach the target. Note that, as discussed before, such dives can be eliminated to respect the limits on the functional characteristics of the float by setting the appropriate value of the tolerance parameter τ.

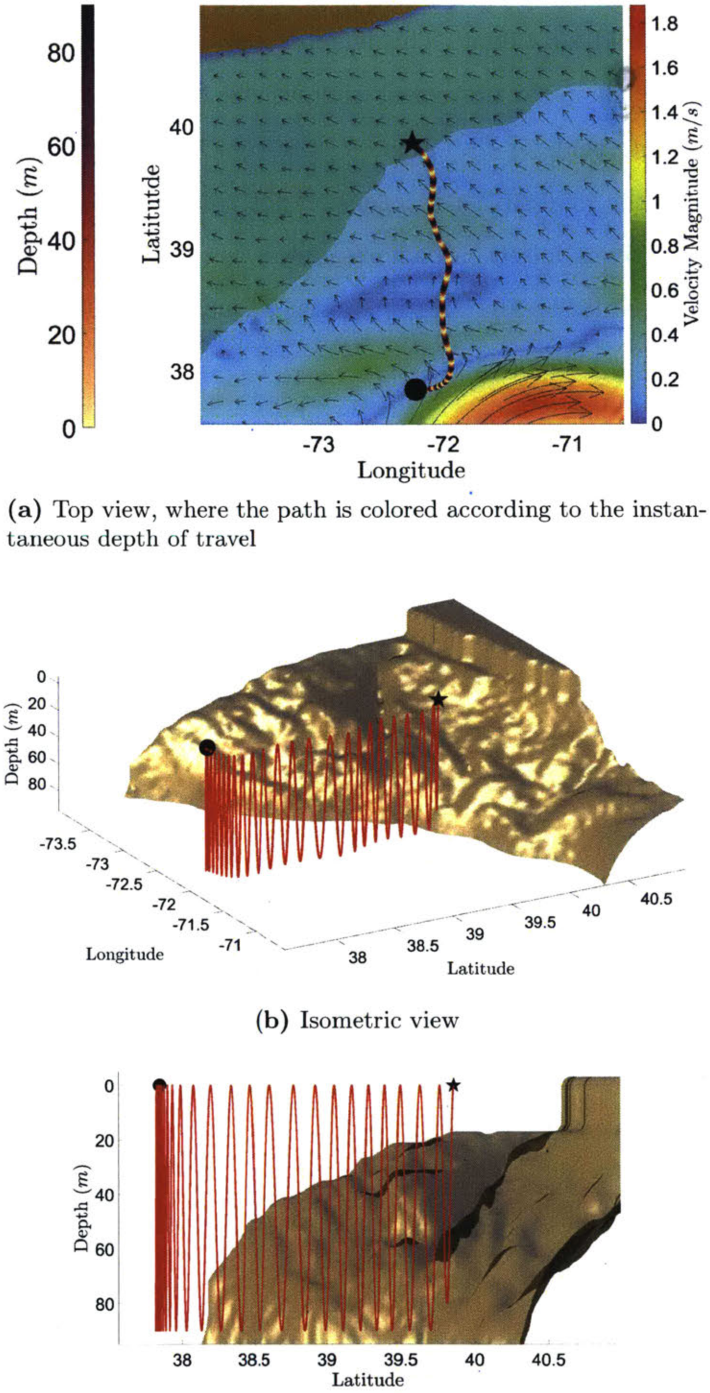
4.3.3 Three-Dimensional Vehicles with Fixed Vertical Motion

As the final example in this chapter, we look at a sea-glider operating in the aforementioned region that performs a sinusoidal yo-yo motion. This is the most typical motion that sea-glider perform in order to sample data (?Ramp et al., 2009; Haley et al., 2009; Leonard et al., 2010).

Specifically, we look at a sea glider that performs this sinusoidal motion with dives up to 90 meters in depth. The time-period of each of the dives is 4 hours, which implies an average vertical speed of 1.25 cm/s. The speed of the glider in the horizontal plane is 50 cm/s. (Stokey et al., 2005). As discussed before, in the case of known vertical motion of the vehicle, the 3 dimensional path planning problem reduces to a 2 dimensional problem. Hence, these simulations are carried out on a 2 dimensional grid of resolution 128 × 128 elements, using the WENO scheme for the spatial discretization and the TVD RK3 for time marching. While backtracking the optimal path, we add the deterministic displacement due to the vertical motion of the vehicle to the path being backtracked in 2D to obtain the full 3 dimensional optimal path.

During the forward evolution, the level set is evolved as a 2D parametric contour. If some of the points on this contour hit the ocean floor at some particular time, then the direction of travel of the vehicle is reversed only at these points. The vehicle would continue to move in the same direction at the other points until it hits the ocean floor or reaches the maximum allowable depth. This data of points at which such reversal of direction occurs is kept in memory and it is enforced while computing the optimal path backward in time. While planning paths for such vehicles with known vertical motion, it might happen that the vehicle reaches the location of the target, but is at a different depth. In such cases, we assume that once the vehicle reaches the destination coordinates (albeit at a different depth), it can rise / sink locally to reach the exact target position.

Figure 4-16 shows the optimal path to be followed by a sea-glider performing a sinusoidal yo-yo motion. The required time for this journey is 4.15 days (almost 25 dives), which is more than that taken by a vehicle with isotropic speed, but much lesser than that required for a float. This is expected, as we impose no constraints on the motion of the vehicle in the first case, whereas stricter constraints are enforced on the float motion. Note that for the last dive, the vehicle is close to the sea floor and hence it terminates the dive earlier in order to not collide with the land mass. The optimal path qualitatively follows a pattern similar to the first case, where it initially attempts to use the Gulf Stream, and it is later pushed eastwards due to the inflow from the Hudson River Canyon.



(a) Top view, where the path is colored according to the instantaneous depth of travel

(b) Isometric view

(c) Side view

Figure 4-16: Optimal path to be followed by a glider from August 30 to September 2, 2006 in the mid-Atlantic bight region. The glider performs a yo-yo motion with dives up to 90 meters deep, and completes approximately 25 dives while traveling to the destination, requiring 4.15 days for the journey. For the last dive, the glider does not dive to 90 m due to the obstruction from sea floor.

Chapter 5

Conclusions and Future Work

In this chapter, we first look at a summary of the work done in this thesis, followed by some of the key results. We then highlight some of the possible future work directions regarding this topic.

5.1 Summary and Conclusions

Autonomous vehicles play a crucial role in several oceanic applications. Their high level of autonomy and endurance is ideal for long-range data sampling as well as other research and societal missions. Many such tasks require the vehicles to travel to particular target positions in the shortest amount of time.

Underwater vehicles frequently experience dynamic ocean currents whose effects on the vehicle's motion cannot be ignored (Schmidt et al., 1996; Elisseeff et al., 1999; Yan et al., 2014). Further, such multi-scale currents may vary with depth (Cushman-Roisin and Beckers, 2011). These domains might also have shallow ocean depth and hence a constrained operable region. All such factors make it extremely challenging to compute optimal trajectories for these vehicles to follow. Ideally the vehicles should be able to efficiently use the oceanic velocities at various depths to their maximum advantage in order to optimally travel to the destination. Further, various types of autonomous vehicles have different motion constraints (Rudnick et al., 2004; Kobayashi et al., 2012), and hence the optimal path planning strategy has to take into account such constraints while computing a feasible and optimal trajectory.

In this thesis, we look at the theoretical foundations and the practical aspects of exact time-optimal path planning equations for three dimensional realistic ocean fields, as well as optimal planning strategies for different types of marine vehicles. The problem is broadly divided in to two segments: (i) given the start and the target positions as well as the dynamic flow field, we first study the growth of the reachability set. This set contains all the points to which there exists a trajectory from the start point, the evolution of which is governed by a modified Hamilton-Jacobi equation (Lolla et al., 2012, 2014b). We utilize the level set method to represent and exactly evolve the boundary of this reachability set (also called the reachability front). (ii) The second part involves the backward tracing of the optimal path once the reachability set contains the target position. These optimal paths correspond to the characteristics of the Hamilton-Jacobi level set differential equation, and are solved by tracking the trajectory of the particle that always travels on the reachability front and reaches the destination in the optimal time.

First, we review the theorem that forms the basis of the exact time-optimal path planning (Lolla, 2016; Lolla and Lermusiaux, 2017). This theorem permits us to modify the fundamental time optimal path planning for vehicles with heading-dependent speeds and also for vehicles that can only move is certain directions in 3D. We demonstrate the applications of the exact optimal path planning equations to different types of oceanic vehicles operating in fully three-dimensional domains, and prove that the three-dimensional problem can be reduced to two dimensions if the motion of the vehicle is constrained along some parametric direction, and this constraint is known a-priori (for example, oceanic vehicles such as sea-gliders perform deterministic oscillatory motion in the vertical direction).

Second, robust numerical schemes are developed to solve the path planning equations with numerical consistency and with a high order of accuracy in realistic three-dimensional ocean domains. Fundamental numerical issues arise because these domains have highly skewed length and velocity scales, and even the slightest numerical errors along a certain dimension may have large consequences along the other dimensions. To counter this, we develop a novel non-dimensional form of the Hamilton-Jacobi level set equation that overcomes the skewed-scaling of the lengths and the velocities. This is followed by a discussion about the high order accurate spatial and temporal schemes used for the evolution of the reachability set. Finally, we present the new numerically consistent implicit backward tracing schemes developed for the accurate computation of the optimal trajectory.

We validate the theory and the schemes using standard benchmarking, and also discuss the application of these to a wide range of 3D examples. The reachability set evolution and the optimal trajectory computation are demonstrated and discussed for various dynamic three-dimensional analytical and realistic flow fields, and for the different types of marine vehicles.

5.2 Future Work

The current work is by no means complete, and several potential directions of future work may be identified. We discuss some such possibilities below.

First, we currently solve the Hamilton-Jacobi forward evolution equation over the entire domain. However, as discussed in section §(3.4), a more efficient approach is to only solve this equation in the vicinity of the zero level set surface, as values of the level set function ϕ far away from the zero level set do not affect its evolution. This approach is called narrow band level set method (Adalsteinsson and Sethian, 1995). The narrow band level set method is extremely efficient in two dimension, but its marginal utility is not high for three-dimensional cases, and especially for long-range missions, where the zero level set surface and the corresponding narrow band may occupy a significant subset of the domain. Further, this method is not ideally suited for parallel computing environments, and its advantage is lost as the reinitialization of the narrow band needs to be performed serially. Hence, the present work does not use the narrow band level set method. A possible addition to the present work would be to implement the narrow band level set method for the forward evolution of the reachability front in three dimensions.

Recently, sea-trials were conducted to validate the time-optimal path planning algorithm (Subramani et al., 2017b; Edwards et al., 2017; Mirabito et al., 2017). These trials utilized the two dimensional version of the discussed work and the underwater vehicles traveled at constant, pre-determined depths. An interesting direction of future work (given the required resources) would involve testing the three-dimensional path planning algorithm by conducting actual sea-trials.

In this work, we only look at the optimality in time. In many cases, optimality with accurately respect to some other parameters may be desired. For example, path planning PDEs to minimize the energy spent by the vehicle while during its journey have been derived and applied to analytical and realistic cases (Subramani and Lermusiaux, 2016; Subramani et al., 2017a; Subramani, 2014). In general, our three-dimensional optimal path planning methodology can be extended to cases where the objective function to be optimized depends on different spatial and temporal variables.

An assumption in our work is that the velocity field, although dynamic, is exactly known. Oceanic flow-fields, due to their multi-scale nature, are typically associated with a level of uncertainty, and the exact flow-field may not be known a-priori. Researchers have studied the extension of current work to plan optimal paths in stochastic flow environments (Subramani et al., 2017c). Hence, a possible direction of future work is to extend three-dimensional path planning for the case of uncertain external flow-fields.

Finally, our theory can be extended to implement on-board routing (Lermusiaux, 2007; Heaney et al., 2007; Schofield et al., 2010). As discussed above, ocean field forecasts are typically uncertain and can be accurately represented by a probability density function (pdf) (Lermusiaux, 2006; Cossarini et al., 2009; Sapsis and Lermusiaux, 2009; Lermusiaux et al., 2010; Gawarkiewicz et al., 2011; Sapsis and Lermusiaux, 2012; Rixen et al., 2012; Ueckermann et al., 2013; Feppon and Lermusiaux, 2017; Lermusiaux et al., 2017). As the exact realization of the flow is unknown, on-board routing is required to correct the path of the vehicle during its motion. This can be achieved by initially planning a globally optimal trajectory, and then assimilating the vehicle positions and the observed velocities to correct the flow field at certain intervals. After assimilation, re-planning can be done from these positions in order to compute the modified optimal trajectory.

Appendix A

Reachability Set Evolution – Proof

Theorem. *Let $\boldsymbol{V}(\boldsymbol{x}, t)$ be a Lipschitz continuous velocity field in all its arguments. Assume that the set of permissible heading directions $\mathcal{H}$ is a subset of $\mathbb{R}^n$. Let the vehicle speed $F(\hat{\boldsymbol{h}}, t) : \mathcal{H} \times [0, \infty) \to \mathbb{R}^+$ be Lipschitz continuous in all of its arguments. Let $\boldsymbol{T}(\boldsymbol{x}_f : \boldsymbol{x}_s, 0)$ denote the optimal first arrival time at target $\boldsymbol{x}_f$, given that the vehicle started from start position $\boldsymbol{x}_s$ at time $t = 0$.*

We assume that the possible vehicle trajectories $\boldsymbol{X}(t)$ are governed by equation (2.15), with initial condition $\boldsymbol{X}(0) = \boldsymbol{x}_s$. Then, the evolution of the reachability front is given by the zero level set of the function $\phi : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}$, where $\phi(\boldsymbol{x}, t)$ is the unique viscosity solution of the following equation:

$$\frac{\partial \phi}{\partial t} + \max_{\hat{\boldsymbol{h}}} \left(F(\hat{\boldsymbol{h}}, t)\hat{\boldsymbol{h}}(t) \cdot \nabla \phi + \boldsymbol{V}(\boldsymbol{x}, t) \cdot \nabla \phi \right) = 0 \qquad \text{(A.1)}$$

For the initial conditions:

$$\phi(\boldsymbol{x}, 0) = |\boldsymbol{x} - \boldsymbol{x}_s| \qquad \text{(A.2)}$$

That is,

1. *The optimal arrival time $\boldsymbol{T}(\boldsymbol{x}_f : \boldsymbol{x}_s, 0)$ satisfies*

$$\boldsymbol{T}(\boldsymbol{x}_f : \boldsymbol{x}_s, 0) = \inf_{t \geq 0} \{t \mid \phi(\boldsymbol{x}_f, t) = 0\} \qquad \text{(A.3)}$$

2. *the optimal trajectory (or trajectories) $\boldsymbol{X}(t)$ are given by the characteristic lines of equation (2.19). That is, $\boldsymbol{X}(t)$ satisfies the following equation:*

$$\frac{d\boldsymbol{X}}{dt} = F(\hat{\boldsymbol{h}}, t)\hat{\boldsymbol{h}}(t) + \boldsymbol{V}(\boldsymbol{X}, t) \tag{A.4}$$

Where

$$\hat{\boldsymbol{h}}(t) = \arg\max_{\hat{\boldsymbol{h}}} \left(F(\hat{\boldsymbol{h}}, t)\hat{\boldsymbol{h}}(t) \cdot \nabla\phi\right) \tag{A.5}$$

□

Proof. We prove this theorem in 2 parts:

1. We first show that if the vehicle lies on the zero level set of $\phi(\mathbf{x}, t)$ at a particular time, then it will lie inside the zero level set at all later times if it moves in any other heading apart from the one prescribed by equation (A.1), and that there cannot exist any sequence of headings such that the vehicle will lie outside the zero level set of ϕ at any instant

2. Second, we show that once the vehicle is inside the zero level set of ϕ, then it can never return on the zero level set. This in turn reiterates the view of the zero level set of $\phi(\mathbf{x}, t)$ as the reachability front, and implies that the optimal first arrival time is when this front crosses the destination for the first time

These collectively imply that the vehicle cannot reach any position outside the zero level set of ϕ and is required to move optimally to stay on it at all times. This in turn suggests that the evolution of the reachability set (and front) governed by equation (A.1). Further, in order to always stay on the zero level set, the vehicle needs to follow the prescribed headings. Substituting these headings into equation (2.15), we obtain equation (A.4) and equation (A.5).

In order to show these results, we assume a fictitious vehicle at position $\mathbf{x}$ that lies on the zero level set of ϕ at time t, that is $\phi(\mathbf{x}, t) = 0$. In order to show that this vehicle will lie inside the front if the optimal heading is not followed, we seek the sign of $\phi(\mathbf{x}+\Delta\mathbf{x}, t+\Delta t)$ as $\Delta t \to 0^+$. Assume that this vehicle follows a set of headings denoted by $\widetilde{\mathbf{h}}(t)$, different from the optimal headings $\hat{\mathbf{h}}(t)$. Consider the limit:

$$\lim_{\Delta t \to 0^+} \frac{\phi(\mathbf{x}+\Delta\mathbf{x}, t+\Delta t)}{\Delta t} \tag{A.6}$$

As ϕ is assumed Lipschitz continuous, we have that $\phi(\mathbf{x}+\Delta\mathbf{x}, t+\Delta t) \to 0$ as $\Delta t \to 0^+$. This prompts us to use L'Hôpital's rule to evaluate the limit.

$$\begin{aligned} \lim_{\Delta t \to 0^+} \frac{\phi(\mathbf{x}+\Delta\mathbf{x}, t+\Delta t)}{\Delta t} &= \lim_{\Delta t \to 0^+} \frac{\frac{d\phi(\mathbf{x}+\Delta\mathbf{x}, t+\Delta t)}{dt}}{\frac{dt}{dt}} && \text{(A.7)} \\ &= \lim_{\Delta t \to 0^+} \left(\frac{\partial \phi}{\partial t} + \nabla\phi \cdot \frac{\Delta \mathbf{x}}{\Delta t} \right) && \text{(A.8)} \end{aligned}$$

We know that

$$\lim_{\Delta t \to 0^+} \frac{\Delta \mathbf{x}}{\Delta t} = \frac{d\mathbf{x}}{dt} = \mathbf{V}_{eff} = F_v(\widetilde{\mathbf{h}}, t)\widetilde{\mathbf{h}}(t) + \mathbf{V}(\mathbf{x}(t), t) \tag{A.9}$$

Hence, substituting this into equation (A.7), we obtain the following equation:

$$\lim_{\Delta t \to 0^+} \frac{\phi(\mathbf{x}+\Delta\mathbf{x}, t+\Delta t)}{\Delta t} = \frac{\partial \phi}{\partial t} + \nabla\phi \cdot \left(F_v(\widetilde{\mathbf{h}}, t)\widetilde{\mathbf{h}}(t) + \mathbf{V}(\mathbf{x}(t), t) \right) \tag{A.10}$$

From equation (2.19), we are given that the function ϕ evolves as

$$\frac{\partial \phi}{\partial t} = -\max_{\hat{\mathbf{h}}} \left(F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) \cdot \nabla\phi + \mathbf{V}(\mathbf{x}, t) \cdot \nabla\phi \right) \tag{A.11}$$

Substituting this into equation (A.10), we obtain equation (A.12).

$$\begin{aligned} \lim_{\Delta t \to 0^+} \frac{\phi(\mathbf{x}+\Delta\mathbf{x}, t+\Delta t)}{\Delta t} = &-\max_{\hat{\mathbf{h}}} \left(F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) \cdot \nabla\phi + \mathbf{V}(\mathbf{x}, t) \cdot \nabla\phi \right) \\ &+ \nabla\phi \cdot \left(F_v(\widetilde{\mathbf{h}}, t)\widetilde{\mathbf{h}}(t) + \mathbf{V}(\mathbf{x}(t), t) \right) \end{aligned} \tag{A.12}$$

As $\mathbf{V}(\mathbf{x}, t)$ does not depend on $\hat{\mathbf{h}}$ or $\widetilde{\mathbf{h}}$, equation (A.12) can be simplified to:

$$\lim_{\Delta t \to 0^+} \frac{\phi(\mathbf{x}+\Delta\mathbf{x}, t+\Delta t)}{\Delta t} = -\max_{\hat{\mathbf{h}}} \left(F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) \cdot \nabla\phi \right) + F_v(\widetilde{\mathbf{h}}, t)\widetilde{\mathbf{h}}(t) \cdot \nabla\phi \tag{A.13}$$

Note that the second term on the right hand side of this equation is the value of $F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) \cdot \nabla\phi$ for $\mathbf{h}\,\hat{}(t) = \tilde{\mathbf{h}}(t)$, whereas the first term is the maximum of the same quantity over all permissible $\hat{\mathbf{h}}(t)$. Hence, for all $\tilde{\mathbf{h}}(t)$, we have:

$$\max_{\hat{\mathbf{h}}} \left(F(\hat{\mathbf{h}}, t)\hat{\mathbf{h}}(t) \cdot \nabla\phi\right) \geq F_v(\tilde{\mathbf{h}}, t)\tilde{\mathbf{h}}(t) \cdot \nabla\phi \tag{A.14}$$

Equation (A.13) and equation (A.14) imply that

$$\lim_{\Delta t \to 0^+} \frac{\phi(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t)}{\Delta t} \begin{cases} < 0 \text{ for all } \tilde{\mathbf{h}}(t) \neq \hat{\mathbf{h}}(t) \\ = 0 \text{ for } \tilde{\mathbf{h}}(t) = \hat{\mathbf{h}}(t) \end{cases} \tag{A.15}$$

This implies that if any vehicle on the zero level set of ϕ is not steered according to the optimal heading sequence, then it falls inside the reachability set, as $\phi(\mathbf{x}, t) = 0$ and $\frac{\partial\phi}{\partial t} < 0 \implies \phi(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t) < 0$.

Further, equation (A.13) and equation (A.14) also imply that there cannot exist any $\tilde{\mathbf{h}}$ such that $\frac{\partial\phi}{\partial t} > 0$. This implies that, if the vehicle starts on the zero level set (that is $\phi(\mathbf{x}, t) = 0$), then there cannot exist any sequence of headings such that $\phi(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t) > 0$. That is, if that the vehicle started on the zero level set of ϕ, then it cannot go beyond the zero level set of ϕ at any time. These above inferences complete part 1 of the proof.

Also note that, for the optimal path, equation (A.13) suggests us that the speed of the vehicle should always be the maximum allowable speed for that heading. That is, $F_v(\hat{\mathbf{h}}, t) = F(\hat{\mathbf{h}}, t)$. This fixes the dependency between the two control parameters in the problem, and hence, the only real free parameter that needs to be computed is the vehicle heading.

We now show that if a vehicle lies inside the reachability set at some time t, there there exists no sequence of headings such that it can again travel along the boundary of the reachability set (i.e. the reachability front). Mathematically, this means that if for a vehicle at $\mathbf{x}$ at time t, $\phi(\mathbf{x}_0, t_0) < 0$, then there does not exist any sequence of heading $\mathbf{h}(t)$ such that $\phi(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t) = 0$.

For this purpose, let us assume that $\phi(\mathbf{x}_0, t_0) = k$, where $k < 0$. Consider a new

Lipschitz continuous function $\psi : \mathbb{R}^n \times [0, \infty)$ such that $\psi(\mathbf{x}, t) = \phi(\mathbf{x}, t) - k$. This means that the vehicle lies on the zero level set of ψ. Further, $\phi(\mathbf{x}, t) = 0 \implies \psi(\mathbf{x}, t) = -k$, that is $\psi(\mathbf{x}, t) > 0$. Note that as ψ is shifted from ϕ just from a constant, and hence it also satisfies the same evolution equation (A.1) (albeit with a different initial condition). Using the idea mentioned above, there cannot exist any path from $\mathbf{x}_0$ to $\mathbf{x}$, as $\psi(\mathbf{x}_0, t) = 0$ and $\psi(\mathbf{x}, t) > 0$. This means that once the vehicle is inside the zero level set, it cannot go back on it.

This completes the proof. □

Appendix B

Numerical Schemes

B.1 Hamilton-Jacobi ENO Scheme

The basis of the Essentially Non-Oscillatory (ENO) and the Weighted Essentially Non-Oscillatory (WENO) schemes lies in finding better approximations to the forward and backward differences given by equation (B.1) and equation (B.2). Particularly, these methods try to find the smoothest possible polynomial interpolants to compute the numerical fluxes.

$$\phi_x^+ = \frac{\phi(x + \Delta x) - \phi(x)}{\Delta x} \tag{B.1}$$

$$\phi_x^- = \frac{\phi(x) - \phi(x - \Delta x)}{\Delta x} \tag{B.2}$$

The process of finding these approximations is as follows: first we find ϕ by using the smoothest possible interpolation, and then differentiate it to obtain ϕ_x. The polynomial interpolations are constructed with the help of Newton's divided difference tables. The zeroth divided differences of ϕ are defined at the grid nodes and are given by equation (B.3)

$$D_i^0 \phi = \phi_i \tag{B.3}$$

The first order divided differences are defined midway between the gridpoints and are given by equation (B.4)

$$D^1_{i+\frac{1}{2}}\phi = \frac{D^0_{i+1}\phi - D^0_i\phi}{\Delta x} \tag{B.4}$$

Note that $D^1_{i+\frac{1}{2}}\phi$ and $D^1_{i-\frac{1}{2}}\phi$ are the first order forward and backward difference approximations to the first derivative of ϕ. The second order and the third order divided differences are defined at the nodes and midway between the nodes and are given by equation (B.5) and equation (B.6).

$$D^2_i\phi = \frac{D^1_{i+\frac{1}{2}}\phi - D^1_{i-\frac{1}{2}}\phi}{2\Delta x} \tag{B.5}$$

$$D^3_{i+\frac{1}{2}}\phi = \frac{D^2_{i+1}\phi - D^2_i\phi}{3\Delta x} \tag{B.6}$$

We can now write $\phi(x)$ in the form given by equation (B.7), where each of the term Q_p is constituted by the p^{th} order divided difference.

$$\phi(x) = Q_0 + Q_1(x) + Q_2(x) + Q_3(x) \tag{B.7}$$

Upon differentiation, Q_0 drops out and we get:

$$\phi_x(x_i) = Q'_1(x_i) + Q'_2(x_i) + Q'_3(x_i) \tag{B.8}$$

Now, we use this polynomial to approximate $(\phi_x^-)_i$ and $(\phi_x^+)_i$. We have $Q_1(x) = (D^1_{k+1/2}\phi)(x - x_i)$. To find ϕ_x^-, we substitute $k = i - 1$ and for ϕ_x^+, we consider $k = i$. Note that $Q'(x_i) = D^1_{k+1/2}\phi$. This implies that the contribution of $Q'(x_i)$ to equation (B.8) is the backward and forward difference in the cases of ϕ_x^- and ϕ_x^+ respectively. That is, first order accurate interpolation is indeed first order upwinding.

We now add the contributions due to $Q'_2(x)$ and $Q'_3(x)$ to increase the order of accuracy of our estimate. We have 2 choices for the second order divided difference (we chose $D^1_{k+1/2}\phi$ for the first order correction): $D^2_k\phi$ or $D^2_{k+1}\phi$. The choice here is dictated by our need of choosing smooth interpolants. As divided differences measure variation in the data, we compare $|D^2_k\phi|$ to $|D^2_{k+1}\phi|$ and choose the smaller one. Particularly, if $|D^2_k\phi| < |D^2_{k+1}\phi|$, then set $c = |D^2_k\phi|$ and $k^\star = k - 1$, otherwise, we set $c = |D^2_{k+1}\phi|$ and $k^\star = k$ ($k^\star$ will be used in the third order correction). Finally we have the second order correction as:

$$Q_2(x) = c(x - x_k)(x - x_{k+1}) \tag{B.9}$$

So that,

$$Q_2'(x_i) = c(2(i - k) - 1)\Delta x \tag{B.10}$$

Now to add the third order correction, we compare $|D^3_{k^\star+1/2}\phi|$ and $|D^3_{k^\star+3/2}\phi|$. If $|D^3_{k^\star+1/2}\phi| < |D^3_{k^\star+3/2}\phi|$, then we set $c^\star = D^3_{k^\star+1/2}\phi$, otherwise we set $c^\star = D^3_{k^\star+3/2}\phi$. The third order contribution is then given as follows.

$$Q_3(x) = c^\star(x - x^\star_k)(x - x_{k^\star+1})(x - x_{k^\star+2}) \tag{B.11}$$

So that,

$$Q_3'(x_i) = c^\star(3(i - k^\star)^2 - 6(i - k^\star) + 2)\Delta x^2 \tag{B.12}$$

The third order approximations to ϕ^+_x and ϕ^-_x are then given by equation (B.8), starting with $k = i - 1$ and $k = i$ respectively, which are then used in an upwind scheme to calculate the numerical fluxes.

B.2 Hamilton-Jacobi WENO Scheme

It can be shown that there are exactly 3 possible approximations to $(\phi^-_x)_i$ (and to $(\phi^+_x)_i$). We define following quantities to write the approximations to $(\phi^-_x)_i$ in a concise way (same can be done with $(\phi_x^+)_i$).

$$\begin{aligned}
v_1 &= \frac{\phi_{i-2} - \phi_{i-3}}{\Delta x} \\
v_2 &= \frac{\phi_{i-1} - \phi_{i-2}}{\Delta x} \\
v_3 &= \frac{\phi_{i} - \phi_{i-1}}{\Delta x} \\
v_4 &= \frac{\phi_{i+1} - \phi_{i}}{\Delta x} \\
v_5 &= \frac{\phi_{i+2} - \phi_{i+1}}{\Delta x}
\end{aligned}$$

Then the approximations to $(\phi_x^-)_i$ are given by:

$$\phi_x^1 = \frac{v_1}{3} - \frac{7v_2}{6} + \frac{11v_3}{6} \tag{B.13}$$

$$\phi_x^2 = -\frac{v_2}{6} + \frac{5v_3}{6} + \frac{v_4}{3} \tag{B.14}$$

$$\phi_x^3 = \frac{v_3}{3} + \frac{5v_4}{6} - \frac{v_5}{6} \tag{B.15}$$

It was pointed out that picking exactly one approximation amongst the 3 is an overkill in smooth regions. Hence, a weighted ENO (WENO) scheme was proposed which considered a convex combination of the three approximations to increase the order of accuracy to 5^{th} order in locally smooth regions. That is, the optimal approximation is given by:

$$\phi_x = \omega_1\phi_x^1 + \omega_2\phi_x^2 + \omega_3\phi_x^3 \tag{B.16}$$

where, $\omega_1 + \omega_2 + \omega_3 = 1$. In particular $\omega_1 = 0.1 + O(\Delta x^2)$, $\omega_2 = 0.6 + O(\Delta x^2)$ and $\omega_3 = 0.3 + O(\Delta x^2)$ give us optimal 5^{th} order approximation to ϕ_x in smooth regions. In order for this method to be well suited even for regions with sharp gradients, we defined the weights in the following way. We first estimate the smoothness of the stencils given by equation (B.13), equation (B.14) and equation (B.15) as:

$$S_1 = \frac{13}{12}(v_1 - 2v_2 + v_3)^2 + \frac{1}{4}(v_1 - 4v_2 + 3v_3)^2 \tag{B.17}$$

$$S_2 = \frac{13}{12}(v_2 - 2v_3 + v_4)^2 + \frac{1}{4}(v_2 - v_4)^2 \tag{B.18}$$

$$S_3 = \frac{13}{12}(v_3 - 2v_4 + v_5)^2 + \frac{1}{4}(3v_3 - 4v_4 + v_5)^2 \tag{B.19}$$

Using these smoothness estimates, we define

$$\alpha_1 = \frac{0.1}{(S_1 + \epsilon)^2} \tag{B.20}$$

$$\alpha_2 = \frac{0.6}{(S_2 + \epsilon)^2} \tag{B.21}$$

$$\alpha_3 = \frac{0.3}{(S_3 + \epsilon)^2} \tag{B.22}$$

where,

$$\epsilon = 10^{-6}\max\{v_1^2, \ldots, v_5^2\} + 10^{-99} \tag{B.23}$$

Here, the 10^{-99} is to avoid division by zero. The first term in ϵ is a scaling term that aids in the balance between the 5^{th} order WENO and the digital ENO weights (choosing only one approximation). Once the α values are calculated, we compute the weights by normalizing them with respect to $\Sigma\alpha$. That is:

$$\omega_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3} \tag{B.24}$$

$$\omega_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3} \tag{B.25}$$

$$\omega_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3} \tag{B.26}$$

These weights can be then used in order for the WENO scheme to perform well in locally smooth regions as well as regions with steep gradients.

Similar treatment is considered to approximate $(\phi_x^+)_i$, but forward difference approximations are used for v_is instead of the current backward difference approximations. Other equations stand the same.

B.3 Godunov Scheme

For the case of heading independent speed, the optimal propulsion term is given by:

$$F|\nabla\phi| = F\left(\left(\frac{\partial\phi}{\partial x}\right)^2 + \left(\frac{\partial\phi}{\partial y}\right)^2 + \left(\frac{\partial\phi}{\partial z}\right)^2\right)^{\frac{1}{2}} \tag{B.27}$$

Central idea of the Godunov scheme is to compute $|\nabla\phi|$ by looking at $F|\nabla\phi|$ as:

$$F|\nabla\phi| = \left(\frac{F\phi_x}{|\nabla\phi|}, \frac{F\phi_y}{|\nabla\phi|}, \frac{F\phi_z}{|\nabla\phi|}\right) \cdot \nabla\phi = \mathbf{V}_\phi \cdot \nabla\phi \tag{B.28}$$

Where $\mathbf{V}_\phi$ is a fictitious velocity that decides the approximation for the gradient terms to be used based on upwinding. Since upwinding only considers the sign of the velocity in all directions, we can ignore the factor of $\frac{F}{|\nabla\phi|}$, which is known to be positive.

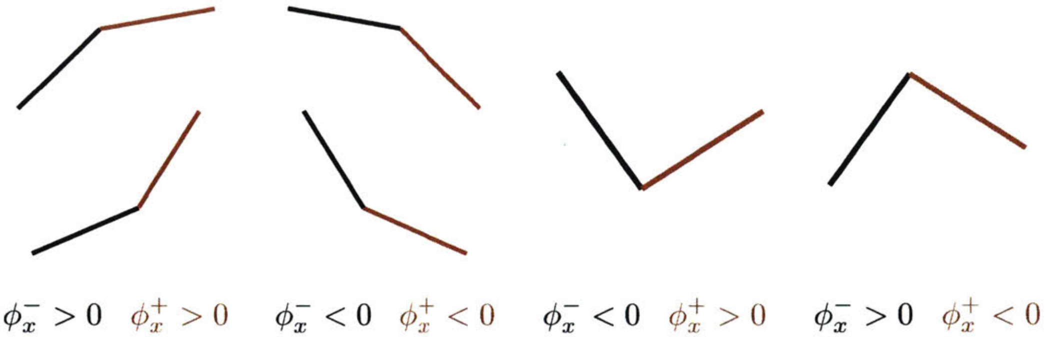
Neglecting $\frac{F}{|\nabla\phi|}$, it is clear that the approximation of ϕ_i is decided by the sign of ϕ_i, for $i = x, y, z$. Specifically, if the sign of ϕ_x is positive, then $\phi_x = \phi_x^-$, given by equation (B.2). Otherwise, $\phi_x = \phi_x^+$, given by equation (B.1). Similar treatment is followed for ϕ_y and ϕ_z.

A caveat is that we do not know the sign of ϕ_x. The formulation is straightforward if the signs of both ϕ_x^- and ϕ_x^+ are the same. If $\phi_x^-, \phi_x^+ < 0$, then it is safe to say that $\phi_x < 0$ and hence, $\phi_x = \phi_x^+$. Similarly, for $\phi_x^-, \phi_x^+ > 0$, we have $\phi_x = \phi_x^-$.

Consider the situation when ϕ_x^+ and ϕ_x^- have different signs. First, we look at the case when $\phi_x^- < 0$ and $\phi_x^+ > 0$. This corresponds to a "V" shaped region, and in order for no new information to be created (entropy condition to hold), each side of this V shape should expand outwards. Gudonov scheme achieves this by setting $\phi_x = 0$. This means that a region of expansion will have a locally flat ϕ, which satisfies the entropy condition. Instead of adding numerical diffusion like some other schemes (e.g. Roe-Fix method), Gudonov scheme chooses the most meaningful realizable solution. We now look at the case where $\phi_x^- > 0$ and $\phi_x^+ < 0$, which corresponds to an inverted V shape. This corresponds to coalescing fronts, and the entropy condition dictates that the information which reaches the concerned point first should be considered. That is, once information from a particular front reaches the concerned point, then its value is not influenced by the information that reaches later. As ϕ_x acts as a fictitious velocity, the larger value amongst $|\phi_x^-|$ and $|\phi_x^+|$ reaches the point x first and should therefore be considered. This implies that, for the case with $\phi_x^- > 0$ and $\phi_x^- < 0$, we set $\phi_x = \max(|\phi_x^-|, |\phi_x^+|)$. Figure B-1 summarizes how the values of ϕ_x are chosen in different cases. $|\nabla\phi|$ is computed once the appropriate value for $\nabla\phi$ is obtained (by choosing the correct values of ϕ_x, ϕ_y and ϕ_z using the above procedure).

The complete Godunov scheme can be concisely represented by equation (B.29) as shown.

$$\begin{aligned} \phi_x^2 &= \max\left(\max\left(\phi_x^-, 0\right)^2, \min\left(\phi_x^+, 0\right)^2\right) \\ \phi_y^2 &= \max\left(\max\left(\phi_y^-, 0\right)^2, \min\left(\phi_y^+, 0\right)^2\right) \\ \phi_z^2 &= \max\left(\max\left(\phi_z^-, 0\right)^2, \min\left(\phi_z^+, 0\right)^2\right) \end{aligned} \tag{B.29}$$


$\phi_x^- > 0$ $\phi_x^+ > 0$
$\phi_x^- < 0$ $\phi_x^+ < 0$
$\phi_x^- < 0$ $\phi_x^+ > 0$
$\phi_x^- > 0$ $\phi_x^+ < 0$

(a) $\phi_x = \phi_x^-$ **(b)** $\phi_x = \phi_x^+$ **(c)** $\phi_x = 0$ **(d)** $\phi_x = \max(|\phi_x^-|, |\phi_x^+|)$

Figure B-1: Upwinding with respect to the sign of ϕ_x to compute the optimal propulsion term. If both ϕ_x^- and ϕ_x^+ are positive, then upwinding dictates that $\phi_x = \phi_x^-$. Similarly for ϕ_x^-, $\phi_x^+ < 0$, we have $\phi_x = \phi_x^+$. For the case of $\phi_x^- > 0$, $\phi_x^+ < 0$, we set $\phi_x = 0$ as no new information can be created. Finally, for $\phi_x^- < 0$, $\phi_x^+ > 0$, we have $\phi_x = \max(|\phi_x^-|, |\phi_x^+|)$ due to the entropy condition.

B.4 Time Marching Schemes

For simplicity, let us write the Hamilton-Jacobi equation governing the forward evolution of ϕ as:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \mathcal{L}(\phi(\mathbf{x}, t), t) \tag{B.30}$$

Velocity is not included as a variable because it is deterministically known. Equation (B.30) is discretized using forward Euler scheme as follows:

$$\phi(\mathbf{x}, t + \Delta t) = \phi(\mathbf{x}, t) + \Delta t \cdot \mathcal{L}(\phi(\mathbf{x}, t), t) \tag{B.31}$$

For simplicity later on, we define a new operator given by $\widehat{\mathcal{L}}$ which essentially comprises of the forward Euler scheme. It is given by equation (B.32).

$$\widehat{\mathcal{L}}\,(\mathbf{y}) = (\mathbf{y} + \Delta t \cdot \mathcal{L}(\mathbf{y})) \tag{B.32}$$

Hence, we have:

$$\phi(\mathbf{x}, t + \Delta t) = \widehat{\mathcal{L}}(\phi(\mathbf{x}, t)) \tag{B.33}$$

Second order TVD RK method is the same as Heun's predictor-corrector method, given by equation (B.34), equation (B.35) and equation (B.36).

We first obtain an auxiliary value of $\phi(\mathbf{x}, t + \Delta t)$, denoted by $\phi(\mathbf{x}, t + \Delta t)^*$, by using forward Euler method.

$$\phi(\mathbf{x}, t + \Delta t)^* = \widehat{\mathcal{L}}(\phi(\mathbf{x}, t)) \tag{B.34}$$

This is followed by a second application of forward Euler method to obtain a first order estimate for, given by equation (B.35)

$$\phi(\mathbf{x}, t + 2\Delta t)^* = \widehat{\mathcal{L}}(\phi(\mathbf{x}, t + \Delta t)^*) \tag{B.35}$$

Finally, we perform an averaging step to arrive at the accurate answer.

$$\phi(\mathbf{x}, t + \Delta t) = \frac{1}{2}\left(\phi(\mathbf{x}, t) + \phi(\mathbf{x}, t + 2\Delta t)^*\right) \tag{B.36}$$

This scheme can be concisely written as in equation (B.37).

$$\phi(\mathbf{x}, t + \Delta t) = \phi(\mathbf{x}, t) + \Delta t \cdot \mathcal{L}\left(\phi(\mathbf{x}, t) + \frac{1}{2}\Delta t \cdot \mathcal{L}(\phi(\mathbf{x}, t), t), t + \frac{1}{2}\Delta t\right) \tag{B.37}$$

Note that this method in turn takes the average (a convex combination) of two independent Euler steps, and hence is TVD. Let us now look at the third order TVD RK scheme, as described below.

Similar to Heun's method, we first obtain the values of $\phi(\mathbf{x}, t + \Delta t)^*$ and $\phi(\mathbf{x}, t + 2\Delta t)^*$ using equation (B.34) and equation (B.35).

This is followed by a weighted averaging step to yield $\phi(\mathbf{x}, t + \frac{1}{2}\Delta t)$, given by equation (B.38).

$$\phi(\mathbf{x}, t + \frac{1}{2}\Delta t) = \frac{3}{4}\phi(\mathbf{x}, t) + \frac{1}{4}\phi(\mathbf{x}, t + 2\Delta t) \tag{B.38}$$

Now, we use this value of $\phi(\mathbf{x}, t + \frac{1}{2}\Delta t)$, and compute $\phi(\mathbf{x}, t + \frac{3}{2}\Delta t)$ again by using forward Euler method.

$$\phi(\mathbf{x}, t + \frac{3}{2}\Delta t)^* = \widehat{\mathcal{L}}(\phi(\mathbf{x}, t + \frac{1}{2}\Delta t)^*) \tag{B.39}$$

This is followed a final weighted averaging step, thus yielding $\phi(\mathbf{x}, t + \Delta t)$.

$$\phi(\mathbf{x}, t + \Delta t) = \frac{1}{3}\phi(\mathbf{x}, t) + \frac{2}{3}\phi(\mathbf{x}, t + \frac{3}{2}\Delta t) \tag{B.40}$$

Bibliography

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

D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148(1):2–22, 1999.

A. Alvarez, A. Caiti, and R. Onken. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE Journal of Oceanic Engineering*, 29 (2):418–429, 2004.

Ş. T. Beşiktepe, P. F. J. Lermusiaux, and A. R. Robinson. Coupled physical and biogeochemical data-driven simulations of Massachusetts Bay in late summer: real-time and post-cruise data assimilation. *Journal of Marine Systems*, 40–41:171–212, 2003. doi: 10.1016/S0924-7963(03)00018-6.

V. Bonnin. *From albatross to long range UAV flight by dynamic soaring*. PhD thesis, University of the West of England, 2016.

A. Bressan. Viscosity solutions of hamilton-jacobi equations and optimal control problems. *Lecture notes, Department of Mathematics, Penn State University*, 2011.

A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 723–730. IEEE, 2011.

N. R. Chapman and J. F. Lynch. Editorial: Special issue on the 2006 shallow water experiment. *IEEE Journal of Oceanic Engineering*, 35(1):1–2, 2010.

D. L. Chopp. Another look at velocity extensions in the level set method. *SIAM Journal on Scientific Computing*, 31(5):3255–3273, 2009.

C. I. Connolly, J. B. Burns, and R. Weiss. Path planning using laplace's equation. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 2102–2106. IEEE, 1990.

G. Cossarini, P. F. J. Lermusiaux, and C. Solidoro. Lagoon of Venice ecosystem: Seasonal dynamics and environmental guidance with uncertainty analyses and error subspace data assimilation. *Journal of Geophysical Research: Oceans*, 114(C6), June 2009. doi:10.1029/2008JC005080.

C. Coulliette and S. Wiggins. Intergyre transport in a wind-driven, quasigeostrophic double gyre: An application of lobe dynamics. *Nonlinear Processes in Geophysics*, 8(1/2):69–94, 2001.

M. G. Crandall and P.-L. Lions. Viscosity solutions of hamilton-jacobi equations. *Transactions of the American Mathematical Society*, 277(1):1–42, 1983.

M. G. Crandall, L. 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.

B. Cushman-Roisin and J.-M. Beckers. *Introduction to geophysical fluid dynamics: physical and numerical aspects*, volume 101. Academic Press, 2011.

R. Dechter and J. Pearl. Generalized best-first search strategies and the optimality of a. *Journal of the ACM (JACM)*, 32(3):505–536, 1985.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

D. R. Durran. Texts in applied mathematics. 1999.

J. Edwards, J. Smith, A. Girard, D. Wickman, D. N. Subramani, C. S. Kulkarni, J. P. J. Haley, C. Mirabito, S. Jana, and P. F. J. Lermusiaux. Data-driven learning and modeling of AUV operational characteristics for optimal path planning. In *Oceans '17 MTS/IEEE Conference*, Aberdeen, June 2017. In Press.

G. D. Egbert and S. Y. Erofeeva. Efficient inverse modeling of barotropic ocean tides. *Journal of Atmospheric and Oceanic Technology*, 19(2):183–204, 2002.

P. Elisseeff, H. Schmidt, M. Johnson, D. Herold, N. Chapman, and M. McDonald. Acoustic tomography of a coastal front in haro strait, british columbia. *The Journal of the Acoustical Society of America*, 106(1):169–184, 1999.

M. Falcone and H. Zidani. Numerical method for hjb equations. optimal control problems and differential games. *ANOC, 23-27 April 2012*, 2012.

F. Feppon and P. F. J. Lermusiaux. Dynamically orthogonal numerical schemes for efficient stochastic advection and lagrangian transport. *SIAM Review*, 2017. Subjudice.

J. H. Ferziger, M. Peric, and A. Leonard. Computational methods for fluid dynamics, 1997.

E. Frazzoli, Z.-H. Mao, J.-H. Oh, and E. Feron. Resolution of conflicts involving many aircraft via semidefinite programming. *Journal of Guidance, Control, and Dynamics*, 24(1):79–86, 2001.

G. Froyland and K. Padberg. Almost-invariant sets and invariant manifoldsâĂŤconnecting probabilistic and geometric descriptions of coherent structures in flows. *Physica D: Nonlinear Phenomena*, 238(16):1507–1523, 2009.

A. Gangopadhyay, A. R. Robinson, P. J. Haley, W. G. Leslie, C. J. Lozano, J. J. Bisagni, and Z. Yu. Feature-oriented regional modeling and simulations in the gulf of maine and georges bank. *Continental Shelf Research*, 23(3):317–353, 2003.

B. Garau, A. Alvarez, and G. Oliver. Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an a* approach. In *Robotics* *and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 194–198. IEEE, 2005.

B. Garau, M. Bonet, A. Alvarez, S. Ruiz, and A. Pascual. Path planning for autonomous underwater vehicles in realistic oceanic current fields: Application to gliders in the western mediterranean sea. *Journal of Maritime Research*, 6(2):5–22, 2014.

G. Gawarkiewicz, S. Jan, P. F. J. Lermusiaux, J. L. McClean, L. Centurioni, K. Taylor, B. Cornuelle, T. F. Duda, J. Wang, Y. J. Yang, T. Sanford, R.-C. Lien, C. Lee, M.-A. Lee, W. Leslie, P. J. Haley, Jr., P. P. Niiler, G. Gopalakrishnan, P. Velez-Belchi, D.-K. Lee, and Y. Y. Kim. Circulation and intrusions northeast of Taiwan: Chasing and predicting uncertainty in the cold dome. *Oceanography*, 24(4):110–121, 2011. doi: http://dx.doi.org/10.5670/oceanog.2011.99.

S. Gottlieb and C.-W. Shu. Total variation diminishing runge-kutta schemes. *Mathematics of computation of the American Mathematical Society*, 67(221):73–85, 1998.

S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43(1):89–112, 2001.

J. Gould, D. Roemmich, S. Wijffels, H. Freeland, M. Ignaszewsky, X. Jianping, S. Pouliquen, Y. Desaubies, U. Send, K. Radhakrishnan, et al. Argo profiling floats bring new era of in situ ocean observations. *Eos*, 85(19):179–184, 2004.

P. J. Haley, Jr. and P. F. J. Lermusiaux. Multiscale two-way embedding schemes for free-surface primitive equations in the "Multidisciplinary Simulation, Estimation and Assimilation System". *Ocean Dynamics*, 60(6):1497–1537, Dec. 2010. doi: 10.1007/s10236-010-0349-4.

P. J. Haley, Jr., P. F. J. Lermusiaux, A. R. Robinson, W. G. Leslie, O. Logoutov, G. Cossarini, X. S. Liang, P. Moreno, S. R. Ramp, J. D. Doyle, J. Bellingham, F. Chavez, and S. Johnston. Forecasting and reanalysis in the Monterey Bay/California Current region for the Autonomous Ocean Sampling Network-II experiment. *Deep Sea Research Part II: Topical Studies in Oceanography*, 56(3–5): 127–148, Feb. 2009. doi:10.1016/j.dsr2.2008.08.010.

P. J. Haley, Jr., A. Agarwal, and P. F. J. Lermusiaux. Optimizing velocities and transports for complex coastal regions and archipelagos. *Ocean Modeling*, 89:1–28, 2015. doi: 10.1016/j.ocemod.2015.02.005.

P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

K. D. Heaney, G. Gawarkiewicz, T. F. Duda, and P. F. J. Lermusiaux. Nonlinear optimization of autonomous undersea vehicle sampling strategies for oceanographic data-assimilation. *Journal of Field Robotics*, 24(6):437–448, 2007. ISSN 1556-4967. doi: 10.1002/rob.20183. URL `http://dx.doi.org/10.1002/rob.20183`.

Y. J. Heo and W. K. Chung. Rrt-based path planning with kinematic constraints of auv in underwater structured environment. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2013 10th International Conference on*, pages 523–525. IEEE, 2013.

B. D. Hessels. Time-optimal Path Planning for Sea-surface Vehicles Under the Effects of Strong Currents and Winds. Bachelor's thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, June 2014.

V. T. Huynh, M. Dunbabin, and R. N. Smith. Convergence-guaranteed time-varying rrt path planning for profiling floats in 4-dimensional flow. In *Australasian Conference on Robotics and Automation 2014*, pages 1–9. Australian Robotics and Automation Association, 2014.

M. Y. Javaid, M. Ovinis, T. Nagarajan, and F. B. Hashim. Underwater gliders: A review. In *MATEC Web of Conferences*, volume 13, page 02020. EDP Sciences, 2014.

G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted eno schemes. *Journal of computational physics*, 126(1):202–228, 1996.

D. B. Johnson. A note on dijkstra's shortest path algorithm. *Journal of the ACM (JACM)*, 20(3):385–388, 1973.

R. Keck. *Reinitialization for level set methods.* Technische Universität Kaiserslautern, Fachbereich Mathematik, 1998.

A. P. Kiraly, J. P. Helferty, E. A. Hoffman, G. McLennan, and W. E. Higgins. Three-dimensional path planning for virtual bronchoscopy. *IEEE Transactions on Medical Imaging*, 23(11):1365–1379, 2004.

T. Kobayashi, K. Asakawa, T. Kawano, T. Hyakudome, M. Matsuura, K.-i. Amaike, K. Watanabe, T. Ino, T. Suga, et al. Deep ninja: A new profiling float for deep ocean observation. In *The Twenty-second International Offshore and Polar Engineering Conference.* International Society of Offshore and Polar Engineers, 2012.

J. Krozel, C. Lee, and J. S. Mitchell. Turn-constrained route planning for avoiding hazardous weather. *Air Traffic Control Quarterly*, 14(2):159–182, 2006.

D. Kruger, R. Stolkin, A. Blum, and J. Briganti. Optimal auv path planning for extended missions in complex, fast-flowing estuarine environments. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4265–4270. IEEE, 2007.

J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 995–1001. IEEE, 2000.

S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

S. M. LaValle and J. J. Kuffner Jr. Rapidly-exploring random trees: Progress and prospects. 2000.

N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay. *Journal of Field Robotics*, 27(6):718–740, 2010.

P. F. J. Lermusiaux. Data assimilation via Error Subspace Statistical Estimation, part II: Mid-Atlantic Bight shelfbreak front simulations, and ESSE validation. *Monthly Weather Review*, 127(7):1408–1432, July 1999. doi: 10.1175/1520-0493(1999)127<1408:DAVESS>2.0.CO;2.

P. F. J. Lermusiaux. On the mapping of multivariate geophysical fields: Sensitivities to size, scales, and dynamics. *Journal of Atmospheric and Oceanic Technology*, 19 (10):1602–1637, 2002.

P. F. J. Lermusiaux. Uncertainty estimation and prediction for interdisciplinary ocean dynamics. *Journal of Computational Physics*, 217(1):176–199, 2006. doi: 10.1016/j.jcp.2006.02.010.

P. F. J. Lermusiaux. Adaptive modeling, adaptive data assimilation and adaptive sampling. *Physica D: Nonlinear Phenomena*, 230(1):172–196, 2007.

P. F. J. Lermusiaux, P. J. Haley, Jr., W. G. Leslie, O. Logoutov, and A. R. Robinson. Autonomous Wide Aperture Cluster for Surveillance (AWACS): Adaptive Sampling and Search Using Predictive Models with Coupled Data Assimilation and Feedback - Harvard Page, 2006. URL `http://mseas.mit.edu/archive/AWACS/index_AWACS.html`.

P. F. J. Lermusiaux, J. Xu, C.-F. Chen, S. Jan, L. Chiu, and Y.-J. Yang. Coupled ocean–acoustic prediction of transmission loss in a continental shelfbreak region: Predictive skill, uncertainty quantification, and dynamical sensitivities. *IEEE Journal of Oceanic Engineering*, 35(4):895–916, Oct. 2010. doi: 10.1109/JOE.2010.2068611.

P. F. J. Lermusiaux, P. J. Haley, Jr., G. G. Gawarkiewicz, and S. Jan. Evaluation of multiscale ocean probabilistic forecasts: Quantifying, predicting and exploiting uncertainty. *Ocean Dynamics*, 2017. To be submitted.

W. G. Leslie, P. J. Haley, Jr., P. F. J. Lermusiaux, M. P. Ueckermann, O. Logutov, and J. Xu. MSEAS Manual. MSEAS Report 06, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 2010. URL `http://mseas.mit.edu/?p=2237`.

R. J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

C. Li, C. Xu, C. Gui, and M. D. Fox. Level set evolution without re-initialization: a new variational formulation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 430–436. IEEE, 2005.

Y.-T. Lin, A. E. Newhall, T. F. Duda, P. F. Lermusiaux, and P. J. Haley. Merging multiple-partial-depth data time series using objective empirical orthogonal function fitting. *IEEE Journal of Oceanic Engineering*, 35(4):710–721, 2010.

S. V. T. Lolla. Path planning in time dependent flows using level set methods. Master's thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, September 2012.

S. V. T. Lolla. *Path Planning and Adaptive Sampling in the Coastal Ocean.* PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, Feb. 2016.

T. Lolla and P. F. J. Lermusiaux. A forward reachability equation for minimum-time path planning in strong dynamic flows. *SIAM Journal on Control and Optimization*, 2017. Sub-judice.

T. Lolla, M. P. Ueckermann, K. Yiğit, P. J. Haley, Jr., and P. F. J. Lermusiaux. Path planning in time dependent flow fields using level set methods. In *IEEE*

International Conference on Robotics and Automation (ICRA), 14-18 May 2012, pages 166–173, 2012. doi: doi: 10.1109/ICRA.2012.6225364.

T. Lolla, P. J. Haley, Jr., and P. F. J. Lermusiaux. Time-optimal path planning in dynamic flows using level set equations: Realistic applications. *Ocean Dynamics*, 64(10):1399–1417, 2014a. doi: doi:10.1007/s10236-014-0760-3.

T. Lolla, P. F. J. Lermusiaux, M. P. Ueckermann, and P. J. Haley, Jr. Time-optimal path planning in dynamic flows using level set equations: Theory and schemes. *Ocean Dynamics*, 64(10):1373–1397, 2014b. doi: doi:10.1007/s10236-014-0757-y.

T. Lolla, P. J. Haley, Jr., and P. F. J. Lermusiaux. Path planning in multiscale ocean flows: coordination and dynamic obstacles. *Ocean Modelling*, 94:46–66, 2015. doi: 10.1016/j.ocemod.2015.07.013.

T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, 1979.

E. Lust and J. Stevens. Autonomous vehicle for gathering oceanographic data in littoral regions. *Preliminary design review report*, 2015.

C. Min. On reinitializing level set functions. *Journal of computational physics*, 229 (8):2764–2772, 2010.

C. Mirabito, D. N. Subramani, T. Lolla, J. P. J. Haley, A. Jain, P. F. J. Lermusiaux, C. Li, D. K. P. Yue, Y. Liu, F. S. Hover, N. Pulsone, J. Edwards, K. E. Railey, and G. Shaw. Autonomy for surface ship interception. In *Oceans '17 MTS/IEEE Conference*, Aberdeen, June 2017. In Press.

S. Mittal and K. Deb. Three-dimensional offline path planning for uavs using multiobjective evolutionary algorithms. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 3195–3202. IEEE, 2007.

W. Mulder, S. Osher, and J. A. Sethian. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics*, 100(2):209–228, 1992.

A. E. Newhall, T. F. Duda, K. Von der Heydt, J. D. Irish, J. M. Kemp, S. Lerner, S. P. Liberatore, Y. Lin, J. F. Lynch, A. Maffei, A. K. Morozov, A. Shmelev, C. Sellers, and W. Witzell. Acoustic and oceanographic observations and configuration information for the whoi moorings from the SW06 experiment. WHOI Technical Report WHOI-2007-04, Woods Hole Oceanographic Institution, 2007.

S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.

S. Osher and R. P. Fedkiw. Level set methods: an overview and some recent results. *Journal of Computational physics*, 169(2):463–502, 2001.

S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

A. Ovsyannikov, V. SabelâĂŹnikov, and M. Gorokhovski. A new level set equation and its numerical assessments. In *Proceedings of the Summer Program*, page 315, 2012.

J. Pedlosky. *Geophysical fluid dynamics.* Springer Science & Business Media, 2013.

D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A pde-based fast local level set method. *Journal of computational physics*, 155(2):410–438, 1999.

A. A. Pereira, J. Binney, G. A. Hollinger, and G. S. Sukhatme. Risk-aware path planning for autonomous underwater vehicles using predictive ocean models. *Journal of Field Robotics*, 30(5):741–762, 2013.

P.-O. Persson. The level set method. *Lecture Notes, Numerical Methods for PDEs (2.097), MIT*, 16, 2005.

Y. Petillot, I. T. Ruiz, and D. M. Lane. Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar. *IEEE Journal of Oceanic Engineering*, 26(2):240–251, 2001.

J. Prete and J. Mitchell. Safe routing of multiple aircraft flows in the presence of time-varying weather data. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 4791, 2004.

S. R. Ramp, R. E. Davis, N. E. Leonard, I. Shulman, Y. Chao, A. Robinson, J. Marsden, P. F. Lermusiaux, D. M. Fratantoni, J. D. Paduan, et al. Preparing to predict: the second autonomous ocean sampling network (aosn-ii) experiment in the monterey bay. *Deep Sea Research Part II: Topical Studies in Oceanography*, 56(3): 68–86, 2009.

A. Richards and J. P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *American Control Conference, 2002. Proceedings of the 2002*, volume 3, pages 1936–1941. IEEE, 2002.

P. L. Richardson. High-speed robotic albatross: Unmanned aerial vehicle powered by dynamic soaring. 2012.

M. Rixen, P. F. J. Lermusiaux, and J. Osler. Quantifying, predicting, and exploiting uncertainties in marine environments. *Ocean Dynamics*, 62(3):495–499, 2012. doi: doi:10.1007/s10236-012-0526-8.

D. Roemmich, G. C. Johnson, S. C. Riser, R. E. Davis, J. Gilson, W. B. Owens, S. L. Garzoli, C. Schmid, and M. Ignaszewski. The argo program: Observing the global ocean with profiling floats. 2009.

D. L. Rudnick, R. E. Davis, C. C. Eriksen, D. M. Fratantoni, and M. J. Perry. Underwater gliders for ocean research. *Marine Technology Society Journal*, 38(2): 73–84, 2004.

G. Russo and P. Smereka. A remark on computing distance functions. *Journal of Computational Physics*, 163(1):51–67, 2000.

V. Sabelnikov, A. Y. Ovsyannikov, and M. Gorokhovski. Modified level set equation and its numerical assessment. *Journal of Computational Physics*, 278:1–30, 2014.

G. P. Sachs, J. Lenz, and F. Holzapfel. Dynamic soaring of albatrosses over land. In *AIAA Atmospheric Flight Mechanics (AFM) Conference*, page 4842, 2013.

E. K. Sackmann, A. L. Fulton, and D. J. Beebe. The present and future role of microfluidics in biomedical research. *Nature*, 507(7491):181–189, 2014.

T. P. Sapsis and P. F. J. Lermusiaux. Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Physica D: Nonlinear Phenomena*, 238 (23–24):2347–2360, Dec. 2009. doi:10.1016/j.physd.2009.09.017.

T. P. Sapsis and P. F. J. Lermusiaux. Dynamical criteria for the evolution of the stochastic dimensionality in flows with uncertainty. *Physica D: Nonlinear Phenomena*, 241(1):60–76, 2012. doi: doi:10.1016/j.physd.2011.10.001.

H. Schmidt, J. G. Bellingham, M. Johnson, D. Herold, D. M. Farmer, and R. Pawlowicz. Real-time frontal mapping with auvs in a coastal environment. In *OCEANS'96. MTS/IEEE. Prospects for the 21st Century. Conference Proceedings*, volume 3, pages 1094–1098. IEEE, 1996.

O. Schofield, S. Glenn, J. Orcutt, M. Arrott, M. Meisinger, A. Gangopadhyay, W. Brown, R. Signell, M. Moline, Y. Chao, S. Chien, D. Thompson, A. Balasuriya, P. F. J. Lermusiaux, and M. Oliver. Automated sensor networks to advance ocean science. *Eos Trans. AGU*, 91(39):345–346, Sept. 2010. doi: 10.1029/2010EO390001.

J. A. Sethian. Curvature flow and entropy conditions applied to grid generation. *Journal of Computational Physics*, 115(2):440–454, 1994.

J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

J. A. Sethian. Adaptive fast marching and level set methods for propagating interfaces. *Acta Math. Univ. Comenianae*, 67(1):3–15, 1998.

J. A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999a.

J. A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999b.

J. A. Sethian and P. Smereka. Level set methods for fluid interfaces. *Annual review of fluid mechanics*, 35(1):341–372, 2003.

S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena*, 212(3):271–304, 2005.

C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988.

R. L. Siddon. Fast calculation of the exact radiological path for a three-dimensional ct array. *Medical physics*, 12(2):252–255, 1985.

R. N. Smith, A. Pereira, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme. Autonomous underwater vehicle trajectory design coupled with predictive ocean models: A case study. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4770–4777. IEEE, 2010.

J. Smoller. Discontinuous solutions of conservation laws. In *Shock Waves and Reaction Diffusion Equations*, pages 239–264. Springer, 1994.

P. E. Souganidis. Existence of viscosity solutions of hamilton-jacobi equations. *Journal of differential equations*, 56(3):345–390, 1985.

R. P. Stokey, A. Roup, C. von Alt, B. Allen, N. Forrester, T. Austin, R. Goldsborough, M. Purcell, F. Jaffre, G. Packard, et al. Development of the remus 600 autonomous underwater vehicle. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1301–1304. IEEE, 2005.

D. N. Subramani. Energy Optimal Path Planning Using Stochastic Dynamically Orthogonal Level Set Equations. Master's thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, September 2014.

D. N. Subramani and P. F. J. Lermusiaux. Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization. *Ocean Modeling*, 100:57–77, 2016. doi: 10.1016/j.ocemod.2016.01.006.

D. N. Subramani, P. J. H. Jr., and P. F. J. Lermusiaux. Energy-optimal path planning in the coastal ocean. *Journal of Geophysical Research: Oceans*, 2017a. Sub-judice.

D. N. Subramani, P. F. J. Lermusiaux, J. P. J. Haley, C. Mirabito, S. Jana, C. S. Kulkarni, A. Girard, D. Wickman, J. Edwards, and J. Smith. Time-optimal path planning: Real-time sea exercises. In *Oceans '17 MTS/IEEE Conference*, Aberdeen, June 2017b. In Press.

D. N. Subramani, Q. J. Wei, and P. F. J. Lermusiaux. Stochastic time-optimal path planning in uncertain dynamic flows. *Ocean Dynamics*, 2017c. To be submitted.

M. Sun, Z. Wang, and X.-S. Bai. Assessment and modification of sub-cell-fix method for re-initialization of level-set distance function. *International journal for numerical methods in fluids*, 62(2):211–236, 2010.

M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1):146–159, 1994a.

M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1):146–159, 1994b.

M. Sussman, E. Fatemi, P. Smereka, and S. Osher. An improved level set method for incompressible two-phase flows. *Computers & Fluids*, 27(5):663–680, 1998.

P. K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM journal on numerical analysis*, 21(5):995–1011, 1984.

D. Tang, J. N. Moum, J. F. Lynch, P. Abbot, R. Chapman, P. H. Dahl, T. F. Duda, G. Gawarkiewicz, S. Glenn, J. A. Goff, et al. Shallow water'06: A joint acoustic propagation/nonlinear internal wave physics experiment. 2007.

P. Testor, G. Meyers, C. Pattiaratchi, R. Bachmayer, D. Hayes, S. Pouliquen, L. Petit de la Villeon, T. Carval, A. Ganachaud, L. Gourdeau, et al. Gliders as a component of future observing systems. OceanObs' 09, 2010.

M. P. Ueckermann and P. F. J. Lermusiaux. 2.29 Finite Volume MATLAB Framework Documentation. MSEAS Report 14, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 2012. URL `http://mseas.mit.edu/?p=2567`.

M. P. Ueckermann, P. F. J. Lermusiaux, and T. P. Sapsis. Numerical schemes for dynamically orthogonal equations of stochastic fluid and ocean flows. *Journal of Computational Physics*, 233:272–294, Jan. 2013. doi: doi:10.1016/j.jcp.2012.08.041.

C. Urmson and R. Simmons. Approaches for heuristically biasing rrt growth. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1178–1183. IEEE, 2003.

Y. Wang and G. S. Chirikjian. A new potential field method for robot path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 977–982. IEEE, 2000.

Q. J. Wei. Time-optimal path planning in uncertain flow fields using stochastic dynamically orthogonal level set equations. Bachelor's thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, June 2015.

WHOI. Shallow water experiment 2006. `http://acoustics.whoi.edu/sw06/`, 2006.

M. P. Wiedeman. Dimensions of blood vessels from distributing artery to collecting vein. *Circulation research*, 12(4):375–378, 1963.

J. Witt and M. Dunbabin. Go with the flow: Optimal auv path planning in coastal environments. In *Australian Conference on Robotics and Automation*, volume 2008, 2008.

E. Wong and K. Fu. A hierarchical orthogonal space approach to three-dimensional path planning. *IEEE Journal on Robotics and Automation*, 2(1):42–53, 1986.

J. Xu, P. F. J. Lermusiaux, P. J. Haley Jr., W. G. Leslie, and O. G. Logutov. Spatial and Temporal Variations in Acoustic propagation during the PLUSNet-07 Exercise in Dabob Bay. In *Proceedings of Meetings on Acoustics (POMA)*, volume 4, page 11. Acoustical Society of America 155th Meeting, 2008. doi: doi:10.1121/1.2988093.

W. Yan, X. Bai, X. Peng, L. Zuo, and J. Dai. The routing problem of autonomous underwater vehicles in ocean currents. In *OCEANS 2014-TAIPEI*, pages 1–6. IEEE, 2014.

A. Zamuda and J. D. H. Sosa. Differential evolution and underwater glider path planning applied to the short-term opportunistic sampling of dynamic mesoscale ocean structures. *Applied Soft Computing*, 24:95–108, 2014.

H.-K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. In *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*, pages 194–201. IEEE, 2001.

X.-H. Zhao, K.-H. Kwek, J.-B. Li, and K.-L. Huang. Chaotic and resonant streamlines in the abc flow. *SIAM Journal on Applied Mathematics*, 53(1):71–77, 1993.

Y. J. Zhao. Optimal patterns of glider dynamic soaring. *Optimal control applications and methods*, 25(2):67–89, 2004.

Y. J. Zhao and Y. C. Qi. Minimum fuel powered dynamic soaring of unmanned aerial vehicles utilizing wind gradients. *Optimal control applications and methods*, 25(5): 211–233, 2004.