Time-optimal path planning in dynamic flows using level set equations: theory and schemes

Ocean Dynamics manuscript No.
(will be inserted by the editor)

Time-Optimal Path Planning in Dynamic Flows using Level Set Equations: Theory and Schemes.

T. Lolla · P. F. J. Lermusiaux · M. P. Ueckermann · P. J. Haley Jr.

Received: date / Accepted: date

Abstract We develop an accurate partial differential equation based methodology that predicts the time-optimal paths of autonomous vehicles navigating in any continuous, strong and dynamic ocean currents, obviating the need for heuristics. The goal is to predict a sequence of steering directions so that vehicles can best utilize or avoid currents to minimize their travel time. Inspired by the level set method, we derive and demonstrate that a modified level set equation governs the time-optimal path in any continuous flow. We show that our algorithm is computationally efficient and apply it to a number of experiments. First, we validate our approach through a simple benchmark application in a Rankine vortex flow for which an analytical solution is available. Next, we apply our methodology to more complex, simulated flow-fields such as unsteady double-gyre flows driven by wind stress and flows behind a circular island. These examples show that time-optimal paths for multiple vehicles can be planned, even in the presence of complex flows in domains with obstacles. Finally, we present, and support through illustrations, several remarks that describe specific features of our methodology.

Keywords path planning · level set · reachability · dynamic flows · ocean sampling · AUVs · gliders · time-optimal · energy-optimal · obstacles · generalized gradients · viscosity solutions

T. Lolla
Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Mass. Ave., Cambridge, MA - 02139
E-mail: ltapovan@mit.edu

P. F. J. Lermusiaux
Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Mass. Ave., Cambridge, MA - 02139
E-mail: pierrel@mit.edu

M.P. Ueckermann
Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Mass. Ave., Cambridge, MA - 02139
E-mail: mpuecker@mit.edu

P. J. Haley Jr.
Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Mass. Ave., Cambridge, MA - 02139
E-mail: phaley@mit.edu

1 Introduction

The problem of path planning has a long history in several branches of science and engineering, especially robotics. However, it does not have a universal solution, primarily due to the broad usage of the term and the wide spectrum of complexity associated with it. In most recent cases, these paths are planned for autonomous robots performing tasks with little human intervention. In the most general sense, path planning refers to a set of rules provided to the autonomous robot for navigating from one configuration to another in an optimal fashion, i.e., by optimizing an objective performance criterion. Since a wide variety of tasks are assigned to autonomous robots, varied path planning rules are utilized.

Autonomous Underwater Vehicles (AUVs) are employed for ocean mapping, commercial exploration, naval reconnaissance and harbor protection. By making measurements of field quantities of interest in the ocean, they enable ocean prediction and other types of scientific research (Lermusiaux, 2007; Schofield et al, 2010). Their path planning may involve minimization of travel time or energy spent by the vehicle. This planning must also take into account the possibly dynamic nature of the environment and limited capabilities of the robot itself. The challenge therefore, is to develop rigorous theories and computationally efficient schemes that accommodate both environmental forcing and robotic constraints while at the same time provide an exact, optimal path for the robot. Applications shown in this paper focus on time-optimal path planning for swarms of underwater robots such as gliders and propelled vehicles. Nevertheless, the methodology is valid for a much wider class of vehicles including small vessels, ships, aircrafts and ground vehicles (if under advection by the environment). These vehicles are employed in a wide range of industries and human activities. Thus, accurate path planning can lead to major savings on resources such as fuel and also limit environmental impacts.

Underwater gliders are ideal for long range sampling missions due to their low power consumption and high levels of autonomy (Lermusiaux et al, 2014). Their endurance however comes at the expense of smaller travel speeds. In many cases, the glider speed becomes comparable to, or even less than that of ocean currents in which it operates. Thus, the dynamic nature of the ocean currents and their effect on vehicle speed should not be neglected. In addition, as these vehicles have become more reliable and affordable, their simultaneous use in sampling and exploratory missions has become viable (Bahr et al, 2009; Fiorelli et al, 2004; Ramp et al, 2009; Haley et al, 2009; Schofield et al, 2010), possibly with coordination (Leonard et al, 2007; Zhang et al, 2007; Bhatta et al, 2005), enabling inter-vehicle information exchange (Bahr et al, 2009; Paley et al, 2008; Davis et al, 2009). This naturally raises the central question of how to optimally navigate swarms of vehicles through these possibly strong and dynamic ocean currents, which often have large variability in both space and time. Moreover, similar to our common use of weather predictions, it is essential to utilize current predictions (up to the predictability limit) for this planning. As most gliders and AUVs receive position fixes or communicate only intermittently, we wish to predict their optimal controls ahead of time by using current forecasts.

We present a rigorous (partial differential equation based) methodology inspired by the level set method, to compute continuous time-optimal paths of swarms of underwater vehicles, obviating the need for heuristic approaches. The methodology predicts the exact fastest path between any two points along with the sequence of vehicle steering directions that realize this fastest path. The methodology automatically generates vehicle trajectories that avoid obstacles, both stationary and mobile.

Next, we first briefly review prior results on robotic and underwater path planning. In §2, we formally define our problem and introduce the relevant notation. In §3, we briefly review level set methods and develop the basis of our approach to path planning. The main theoretical results are presented in §4. Numerical and implementation details are discussed in §5. In §6, we present some applications, ranging from simple benchmark test cases to more complex and realistic flow-fields. A summary and conclusions are presented in §7. Applications in realistic multiscale ocean flows and complex geometry are provided in the companion paper (Lolla et al, 2014b).

1.1 Prior work

Traditionally, robotic path planning has focused on generating safe trajectories, away from hazardous regions and obstacles. The common difficulty here is in handling the large number of degrees of freedom (DOF) of the robot. Every extension to this basic problem adds in computational complexity (Lolla, 2012; Latombe, 1991). Motion planning for multi DOF systems such as robotic arms (Canny, 1988; Latombe, 1991), including cooperative control (Paley et al, 2008; Leonard and Fiorelli, 2001) and coordination (Bahr et al, 2009; Davis et al, 2009) have been extensively studied. Path planning through unsteady flow-fields has received far less attention in comparison. The challenge here is that the currents directly affect the displacement of the vehicle, making the cost of movement variable, and anisotropic at different points in space (Isern-Gonzalez et al, 2012). In this case, even the seemingly simple task of generating feasible tracks becomes challenging. Most robotic path planning algorithms use dynamic programming based approaches such as Dijkstra's method and the A* algorithm (Rhoads et al, 2010). When applied to dynamic flow environments, they often lead to infeasible paths or have a large computational cost when the environment becomes complex. Algorithms that compute discrete vehicle paths (i.e. on a grid) do not remain optimal when extended to a continuous setting. Finally, it is not uncommon for these algorithms to remain stuck in local minima.

Rapidly exploring Random Trees (RRTs) (Lavalle, 1998; Kuffner and LaValle, 2000) are a randomized approach to path planning for obstacle avoidance that use random sampling to explore the robot workspace. Their ability to quickly and uniformly explore a large workspace has led to their widespread usage in several path planning applications including robotics (Yang et al, 2010; Bruce and Veloso, 2002; Melchior and Simmons, 2007) and ocean cases (Rao and Williams, 2009). However, they don't provide the global optimal and are not suited to cases where the environment is highly dynamic and has strong effects on the robots.

Graph search techniques, such as A* have been used for underwater path planning (Rao and Williams, 2009; Carroll et al, 1992; Garau et al, 2009). A major difficulty here is defining a good heuristic function, as the performance of A* crucially depends on it (Lolla, 2012). A* uses a discretized representation of the domain and the predicted vehicle path may not always pass through the grid points. To correct this, adaptive grid restructuring must be performed. A* performs reasonably well for simple steady flow-fields, but may fail for more realistic flows. A discussion of the computational complexity of A* is provided in §4.4.

Fast marching methods (Sethian, 1999a) have also been applied to underwater path planning. These are similar to Dijkstra's algorithm, but solved in a continuous domain. They solve an Eikonal equation (Sethian, 1999b) to isotropically compute the arrival time function at different points in space. In Petres et al (2007), the regular (isotropic) fast marching method is modified to create an anisotropic version where the cost function depends on the flow-fields. For related approaches using wavefront expansions for underwater path planning, see (Soulignac et al, 2009; Thompson et al, 2010, 2009; Kruger et al, 2007).

Potential field techniques (Warren, 1990; Barraquand et al, 1992) have been widely used for robotic collision avoidance algorithms. The key idea is to introduce an artificial potential field on the obstacles that prevents vehicles from getting very close to them, thus, generating safe paths. Although this approach generates only locally optimal solutions, it is inexpensive, allowing real-time computations. It has been used for underwater path planning (Witt and Dunbabin, 2008), using a cost function that depends on the total vehicle drag, travel time and obstacles in the field. Voronoi diagrams have also been used to solve obstacle avoidance problems in static environments (Garrido et al, 2006) and in flow-fields (Bakolas and Tsiotras, 2010).

Variational calculus based approaches have also been used in underwater path planning (Davis et al, 2009): governing equations for minimal time routes in steady flows are derived and related to Snell's law in optics. Routing strategies to maximize the field mapping skill are also discussed. Such use of path planning for information maximization and adaptive sampling is developed in (Binney et al, 2010; Smith et al, 2010; Choi and How, 2010; Heaney et al, 2007; Yilmaz et al, 2008; Wang et al, 2009).

The solution to the minimum time navigation problem in dynamic flows is governed by a Hamilton-Jacobi-Bellman (HJB) equation (Bryson and Ho, 1975). Rhoads et al (2010) derive a set of Euler-Lagrange equations for the optimal trajectory, which are solved using an extremal field approach. This approach requires tracking a potentially large family of 1-D curves backward in time, for several choices of the arrival time at the end point. Other underwater path planning approaches include Lagrangian Coherent Structures (Zhang et al, 2008), case based reasoning (Vasudevan and Ganesan, 1996) and evolution (Alvarez et al, 2004). We refer to (Lolla, 2012; Lolla et al, 2014c) for more extensive reviews.

2 Problem Statement

Let $\Omega \subseteq \mathbb{R}^n$ be an open set and $F > 0$. Consider a vehicle (P) moving in Ω under the influence of a dynamic flow-field, $\mathbf{V}(\mathbf{x}, t) : \Omega \times [0, \infty) \to \mathbb{R}^n$. We wish to predict a steering rule for P that minimizes its travel time between given start and end points, denoted by $\mathbf{y_s}$ and $\mathbf{y_f}$ respectively. In other words, the goal is to develop an algorithm that predicts the sequence of headings that would result in the fastest time path from $\mathbf{y_s}$ to $\mathbf{y_f}$. Let a general continuous trajectory from $\mathbf{y_s}$ to $\mathbf{y_f}$ be denoted as $\mathbf{X}_P(\mathbf{y_s}, t)$ (see Fig. 1). The vehicle motion, being composed of both nominal motion due to steering and advection due to the flow-field, is governed by the kinematic relation

$$\frac{\mathrm{d}\mathbf{X}_P}{\mathrm{d}t} = \mathbf{U}(\mathbf{X}_P(\mathbf{y_s}, t), t) = F_P(t)\,\hat{\mathbf{h}}(t) + \mathbf{V}(\mathbf{X}_P(\mathbf{y_s}, t), t)\,, \tag{1}$$

where $F_P(t)$ is the speed of the vehicle relative to the flow, with $0 \le F_P(t) \le F$, $\hat{\mathbf{h}}(t)$ is the vehicle heading (steering) direction at time t and $\mathbf{U}(\mathbf{X}_P(\mathbf{y_s}, t), t)$ is the total vehicle velocity. Let $\widetilde{T}(\mathbf{y}) : \Omega \to \mathbb{R}$ denote the 'first arrival time' function, i.e. the first time the vehicle reaches any given $\mathbf{y}$, starting from $\mathbf{y_s}$. Clearly, $\widetilde{T}(\mathbf{y_s}) = 0$. The limiting conditions on $\mathbf{X}_P(\mathbf{y_s}, t)$ are

$$\mathbf{X}_P(\mathbf{y_s}, 0) = \mathbf{y_s}\,, \quad \mathbf{X}_P(\mathbf{y_s}, \widetilde{T}(\mathbf{y_f})) = \mathbf{y_f}\,. \tag{2}$$

We aim to predict the optimal controls for $\hat{\mathbf{h}}(t)$ and $F_P(t)$ that minimize $\widetilde{T}(\mathbf{y_f})$ subject to the equation of motion (1) and limiting conditions (2). (1) and (2) can be interpreted as constraints for this minimization problem. Let the optimal travel time to reach $\mathbf{y_f}$ be $T^\star(\mathbf{y_f})$ and the corresponding optimal trajectory be $\mathbf{X}_P^\star(\mathbf{y_s}, t)$.

Here, we assume that $\mathbf{V}(\mathbf{x}, t)$ is exactly known. In realistic ocean applications, forecast flow-fields are always associated with some levels of uncertainty (Lermusiaux, 2006; Lermusiaux et al, 2006). $\mathbf{V}(\mathbf{x}, t)$ can correspond to, for example, the mode or the mean of the predicted flow-field. Planning paths in predicted probabilistic flows (Sapsis and Lermusiaux, 2009; Ueckermann et al, 2013) are reported in (Lermusiaux et al,

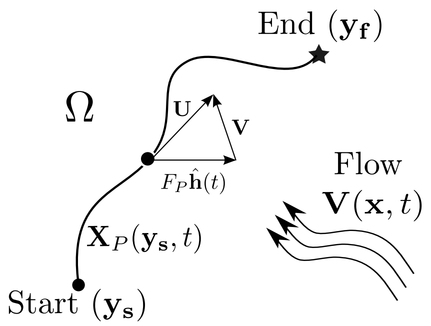


Fig. 1: Motion of P in an unsteady flow-field, $\mathbf{V}(\mathbf{x},t)$. Its trajectory $\mathbf{X}_P(\mathbf{y_s},t)$ connects the start ($\mathbf{y_s}$) and end ($\mathbf{y_f}$) points and satisfies (1)–(2). The total velocity, $\mathbf{U}$ is the vector sum of the steering velocity $F_P(t)\,\hat{\mathbf{h}}(t)$ and flow-field $\mathbf{V}(\mathbf{x},t)$.

2014; Pereira et al, 2013). We consider cases where the distance travelled by the vehicle is much larger than its dimensions thereby assuming the interaction between the vehicle and the flow-field to be purely kinematic. The notation $|\bullet|$ in this paper will denote the L^2 norm of $\bullet$. We assume that $F_P(t)$ and $\hat{\mathbf{h}}(t)$ are Lipschitz continuous in t and that $\mathbf{V}(\mathbf{x},t)$ is bounded and Lipschitz continuous in both $\mathbf{x}$ and t, i.e. $\exists\ C, C_{\mathbf{V}} > 0$ such that

$$\max_{\mathbf{x}\in\Omega, t\geq 0}\{|\mathbf{V}(\mathbf{x},t)|\} \leq C \quad \text{and} \tag{3}$$

$$|\mathbf{V}(\mathbf{x}_1,t_1) - \mathbf{V}(\mathbf{x}_2,t_2)| \leq C_{\mathbf{V}}\left(|\mathbf{x}_1 - \mathbf{x}_2| + |t_1 - t_2|\right), \quad \mathbf{x}_1, \mathbf{x}_2 \in \Omega, t_1, t_2 \geq 0\,. \tag{4}$$

3 Approach

3.1 Control and Reachability

The computation of time-optimal paths in a dynamic flow-field is not trivial. The complexity arises in part, due to the number of control choices available to the vehicle. At every point in its trajectory, the vehicle has an infinite number of heading (steering) directions to choose from (see Fig. 2). For every such heading direction chosen at t, it has again an infinite number of heading choices at the next instant. Thus, it is not trivial to predict the instantaneous vehicle headings that will lead to the quickest path.

Instead of aiming for the exact solution, approximate solutions are often sought. A class of practical schemes is based on heuristic control decisions for the vehicle. For example, a heuristic steering rule can be to always steer in the direction of the end point (LaValle, 2006). However, such approaches are neither guaranteed to be optimal, nor guaranteed to find a feasible trajectory. The problem becomes more complicated when the flow-fields are dynamic; the heuristic control then becomes a function of the velocity field, at least near the vehicle. One solution could be to keep track of the vehicle trajectories for every possible control decision choice, and then choose the sequence of headings that leads to the least travel time. However, this method would be extremely expensive and require a lot of storage.

Our approach to path planning is inspired by the computation of the *reachable set* from a given starting point. A reachable (or attainable) set is defined as the set of points that can be visited by the vehicle at a given time. The boundary of such a set is called the *reachability front*. By tracking the evolution of the reachability front, one can determine when it first reaches the end point. The path traced by the point on the reachability front that first reaches the end point will be the optimal path we wish to compute.

The reachable set $\mathcal{R}(\mathbf{y_s},t)$ (see Fig. 2) at time $t \geq 0$ is the set of all points $\mathbf{y} \in \Omega$ such that there exists a trajectory $\widetilde{\mathbf{X}}_P(\mathbf{y_s},\tau)$ satisfying (1), with $\widetilde{\mathbf{X}}_P(\mathbf{y_s},0) = \mathbf{y_s}$ and $\widetilde{\mathbf{X}}_P(\mathbf{y_s},t) = \mathbf{y}$. Note that the subset of trajectories $\widetilde{\mathbf{X}}_P(\mathbf{y_s},t)$ that reach $\mathbf{y_f}$ is denoted as $\mathbf{X}_P(\mathbf{y_s},t)$.

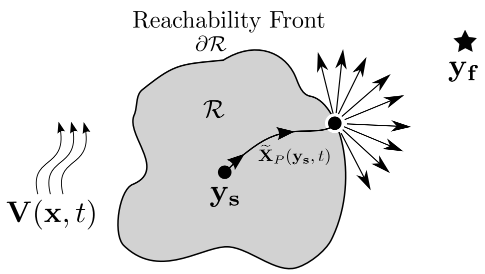


Fig. 2: Reachability front $\partial\mathcal{R}(\mathbf{y_s},t)$ and infinite possible steering directions: $\partial\mathcal{R}$ denotes the boundary of the reachable set $\mathcal{R}(\mathbf{y_s},t)$ (set of points that can be visited at time t).

From this definition of a reachable set (and front) one can ask some key questions which include: if the reachability front exists, can one prove that its evolution is directly linked to that of the time-optimal path in any dynamic flow? what are the equations governing the dynamics of this front and path? and, how can they be computed efficiently? Level set methods, briefly reviewed next, provide leads for the answers. After that, we derive a new level set equation that governs the reachability front (Fig. 2) and time-optimal paths from the origin $\mathbf{y_s}$.

3.2 Modified Level Set Equation and Time-Optimal Paths

Consider a front $\partial\mathcal{R}$, for example, the interface between two immiscible fluids. Level set methods are convenient tools to track the evolution of such a front. They can model the dynamics of the implicit front and capture the interaction between the evolution of the front and fluid forcing. They were originally introduced to solve problems related to fluid-interface motion and front evolution problems (Osher and Sethian, 1988). They can also handle problems in which the speed of the interface depends on various local, global or other independent properties of the system.

Level set methods evolve an interface (a front) by embedding it as a hyper-surface in one higher dimension. For example, an interface in 2D is represented as the zero contour of a 2D scalar field and the evolution of this scalar field governs the movement of the front. This effectively transforms the problem to a 3D one, time being the third dimension. This higher dimensional embedding is what allows for automatic handling of merging and pinching of fronts and other topological changes. Level sets are an implicit representation of the front as opposed to an explicit one. They offer several advantages over an explicit representation (Sethian, 1999b; Osher and Fedkiw, 2003). For any $C \in \mathbb{R}$, the $C-$*level set* of a function $\phi : \mathbb{R}^n \to \mathbb{R}$ is the set $\{\mathbf{x} : \phi(\mathbf{x}) = C\}$.

The choice of $\phi(\mathbf{x})$ is often somewhat arbitrary. The most common function used for this purpose is the *signed distance function*, denoted by $\phi_\rho(\mathbf{x})$. As the name suggests, a distance function $\rho(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}_+$ is the minimum distance of $\mathbf{x}$ from the front, i.e. $\rho(\mathbf{x}) := \min_{\mathbf{x}_i \in \partial\mathcal{R}} |\mathbf{x} - \mathbf{x}_i|$. A signed distance function $\phi_\rho(\mathbf{x})$, is defined as:

$$\phi_\rho(\mathbf{x}) := \begin{cases} \rho(\mathbf{x}), & \text{if } \mathbf{x} \text{ is outside the front}, \\ -\rho(\mathbf{x}), & \text{if } \mathbf{x} \text{ is inside the front}. \end{cases} \tag{5}$$

Clearly, $\phi(\mathbf{x}) = 0$ for all $\mathbf{x} \in \partial\mathcal{R}$, implying that the front is implicitly represented as the zero level set of $\phi(\mathbf{x})$. For all points outside the front, $\phi(\mathbf{x}) > 0$, and for all points inside the front, $\phi(\mathbf{x}) < 0$. Signed distance is a preferred choice for $\phi(\mathbf{x})$ because it is smooth and maintains fixed amplitude gradients in the field.

The level set equation governing the evolution of a front moving in a direction normal to itself at a constant speed $F\,(>0)$ and in a stationary environment (i.e. with zero external flow-field) is (Osher and Fedkiw, 2003):

$$\frac{\partial \phi}{\partial t} + F|\nabla\phi| = 0\,. \tag{6}$$

In (6), the front's motion can be thought of as being driven by an internal velocity, $F\,\hat{\mathbf{n}} = F\,\frac{\nabla\phi}{|\nabla\phi|}$. Considering now the motion of field ϕ solely driven by an external flow $\mathbf{V}(\mathbf{x},t)$, the governing advection equation is

$$\frac{\partial \phi}{\partial t} + \mathbf{V}(\mathbf{x},t)\cdot\nabla\phi = 0\,. \tag{7}$$

If in addition to the external flow-field of (7), the front is also internally driven by its own velocity as in (1), the advection equation (7) becomes

$$\frac{\partial \phi}{\partial t} + \Big(F_P(t)\,\hat{\mathbf{h}}(t) + \mathbf{V}(\mathbf{x},t)\Big)\cdot\nabla\phi = 0\,, \tag{8}$$

where, as in (1), $F_P(t)\,\hat{\mathbf{h}}(t)$ is the velocity of the vehicle relative to the flow-field, of magnitude $0 \le F_P(t) \le F$ and heading direction $\hat{\mathbf{h}}(t)$. If the initial conditions to (8) are given level set conditions, then (8) defines a family of level set equations, each member of the family corresponding to a specific choice of $F_P(t)$ and $\hat{\mathbf{h}}(t)$.

The comparison of (8) to (6) indicates that the heading and magnitude of the relative velocity of the vehicle are free time-dependent control variables of our problem. It also raises the following question: should time-optimal paths be those of vehicles driven in a direction normal to the time-dependent level set similar to (6), even if that level set is externally advected as in (8)?

In §B, we state and prove a theorem that shows that the time-optimal trajectory, if it exists, is indeed obtained by a combination of (6) and (8). The relevant background theory is discussed in §A. Specifically, we show that the viscosity solution to the Hamilton-Jacobi equation

$$\frac{\partial \phi^o}{\partial t} + F|\nabla\phi^o| + \mathbf{V}(\mathbf{x},t)\cdot\nabla\phi^o = 0 \ \text{ in } \ \Omega\times(0,\infty)\,, \tag{9}$$

with initial conditions

$$\phi^o(\mathbf{x},0) = |\mathbf{x} - \mathbf{y_s}| \tag{10}$$

governs the reachable set $\mathcal{R}(\mathbf{y_s},t)$, viz., $\mathcal{R}(\mathbf{y_s},t) = \{\mathbf{x} : \phi^o(\mathbf{x},t) \le 0\}$. In other words, the reachable set coincides with the region(s) where ϕ^o is non-positive. As a result, the minimum time to reach the end point $\mathbf{y_f}$ (i.e. $T^\star(\mathbf{y_f})$) corresponds to the first time the zero level-set of ϕ^o arrives at $\mathbf{y_f}$ (see (33)). Furthermore, we show that the optimal trajectory $\mathbf{X}^\star_P(\mathbf{y_s},t)$ satisfies

$$\frac{\mathrm{d}\mathbf{X}^\star_P}{\mathrm{d}t} = F\,\frac{\nabla\phi^o(\mathbf{X}^\star_P,t)}{|\nabla\phi^o(\mathbf{X}^\star_P,t)|} + \mathbf{V}(\mathbf{X}^\star_P,t), \quad t \in (0, T^\star(\mathbf{y_f})) \tag{11}$$

whenever ϕ^o is differentiable at $(\mathbf{X}^\star_P(\mathbf{y_s},t),t)$. This implies that the vehicle's optimal relative speed equals F, and its optimal heading is normal to the level sets of

ϕ^o. Critically, we show that (9), which is solved to generate all the results shown in this paper, is valid *for all F and $\mathbf{V}$ cases*, even when the flow $\mathbf{V}$ is stronger than F. We also show that in the special case when F is always larger than the flow speed ($F > |\mathbf{V}|$), the minimum arrival time function is also governed by a modified boundary value Eikonal equation (34), which may be efficiently solved using a standard fast marching method (Sethian, 1999a). In the following section, we provide several remarks extending the theorem in §B. Examples corroborating some of the remarks are presented in §5.

3.3 Remarks

Reachability/Existence of Feasible Paths: For a given problem configuration ($\mathbf{y_s}$, $\mathbf{y_f}$, $\mathbf{V}(\mathbf{x}, t)$, and F), the solution to (9) can be used to predict whether or not the vehicle can reach $\mathbf{y_f}$ (or any given point in space) within a specified time limit, $T_{\max}$. For the latter, either the optimal zero level set cannot reach $\mathbf{y_f}$ in finite time, indicating that it is impossible for the vehicle to reach $\mathbf{y_f}$, or may reach $\mathbf{y_f}$, but not within the allowed time limit, $T_{\max}$. In all other cases, the level set method can compute the time-optimal paths to $\mathbf{y_f}$. We refer to §5.2.1 for an illustration.

Applicability of modified Eikonal equation: When the maximum relative vehicle speed F is smaller than the flow-speed $|\mathbf{V}(\mathbf{x}, t)|$ for some $\mathbf{x} \in \Omega$ and $t \geq 0$, the minimum arrival time field $T^o(\mathbf{y})$ may be discontinuous since some points may be visited more than once in the optimal trajectory. At these points, the gradients ∇T^o are not defined and multiple arrival times need to be stored in order to compute the correct optimal trajectory. The modified Eikonal eq. (34) does not admit continuous viscosity solutions in this case. We refer to §5.3.1 for an example.

Optimal Start Time: The initial conditions (10) indicate that the vehicle starts moving at time $t_s = 0$. However, in some cases, the vehicle may reach the end point $\mathbf{y_f}$ faster, if it is deployed at a later start time, $t_s > 0$. §5.3.2 discusses an example of such a scenario.

Forbidden Regions: Time-optimal paths of vehicles moving in dynamic flow-fields may be updated/corrected when 'forbidden' or unsafe regions are introduced in the domain. These regions do not affect the flow-field and are areas in space which the vehicle must avoid. Examples are discussed in (Lolla et al, 2012; Lermusiaux et al, 2014).

Relations to Optimal Control: (9) is a Hamilton-Jacobi equation with Hamiltonian $H(\mathbf{x}, t, \nabla\phi^o) = F|\nabla\phi^o| + \mathbf{V}(\mathbf{x}, t) \cdot \nabla\phi^o$. A problem closely related to ours is the optimal 'time-to-go' problem (Rhoads et al, 2010). Its closed-loop optimal control law can be derived from a dynamic programming principle (Bryson and Ho, 1975; Cannarsa and Sinestrari, 2004). This governing equation for the optimal time-to-go is a HJB equation, and has a structure similar to (9). HJB equations also form the basis of several approaches to compute the reachability fronts in areas of game theory and differential games (Mitchell et al, 2005; Bokanowski et al, 2010).

Optimal Trajectories and Costates: The time-optimal control problem that we study here can also be viewed as a calculus of variations problem. This formulation establishes the existence of a *costate* $\mathbf{q}^\star_P(t) : [0, T^\star(\mathbf{y_f})] \to \mathbb{R}^n$ corresponding to the optimal trajectory $\mathbf{X}^\star_P(\mathbf{y_s}, t)$ and its control (Athans and Falb, 2006). $\mathbf{q}^\star_P(t)$ equals $\nabla\phi^o(\mathbf{X}^\star_P(\mathbf{y_s}, t), t)$, whenever it is defined. Furthermore, the trajectories $\mathbf{X}^o_P(\mathbf{y_s}, t)$ correspond to characteristics of (9) that emanate from $\mathbf{y_s}$.

Uniqueness (single vs. multiple optimal paths): In some situations, there may exist multiple optimal paths to $\mathbf{y_f}$. This happens when two or more characteristics of (9) emanating from $\mathbf{y_s}$ merge at $\mathbf{y_f}$, making ϕ^o non-differentiable at $\mathbf{y_f}$. The viscosity solution to (9) automatically allows for the formation of such singularities or 'shocks'. For end points lying on these shock lines, there exist multiple costates, each corresponding to one of the optimal trajectories. Numerical procedures to treat such cases are mentioned in §C. See §5.3.3 for an example.

Regularity of ϕ^o: The regularity assumption on ϕ^o at points $(\mathbf{X}^o_P(\mathbf{y_s}, t), t)$ for $t > 0$ in part 2 of Theorem 4 (§B) is not a strong one. The value functions arising in several types of optimal control problems (e.g. fixed time problems) are regular (Cannarsa and Sinestrari, 2004). Locally Lipschitz functions that are either differentiable or locally convex or locally semi-convex at a point in their domain are regular there. More details and references may be found in §A.

4 Numerical Implementation and Discussion

4.1 Algorithm and Numerical scheme: Basics

Our path planning algorithm consists of the following two steps:

1. **Forward Propagation**: In this step, the reachability front is evolved by solving the modified level

set eq. (9) forward in time, from the start ($\mathbf{y_s} = \mathbf{0}$). The front is evolved until it reaches the end point ($\mathbf{y_f}$).

2. **Backward Vehicle Tracking**: The optimal vehicle trajectory, $\mathbf{X}^\star_P(\mathbf{y_s}, t)$ and control are computed after the reachability front reaches the end point, by solving (12) backward in time, starting from $\mathbf{y_f}$ at time $T^\star(\mathbf{y_f}) = T^o(\mathbf{y_f})$, i.e.,

$$\frac{\mathrm{d}\mathbf{X}^\star_P(\mathbf{y_s}, t)}{\mathrm{d}t} = -\mathbf{V}(\mathbf{X}^\star_P, t) - F\frac{\nabla\phi^o(\mathbf{X}^\star_P, t)}{|\nabla\phi^o(\mathbf{X}^\star_P, t)|}$$
$$\text{with} \quad \mathbf{X}^\star_P(\mathbf{y_s}, T^\star(\mathbf{y_f})) = \mathbf{y_f}\,. \tag{12}$$

We note that (12) corresponds to (11), when it is solved backward in time. For any $0 < t \leq T^\star(\mathbf{y_f})$, if ϕ^o is not differentiable at $(\mathbf{X}^\star_P(\mathbf{y_s}, t), t)$ the optimal trajectories are obtained by integrating

$$\frac{\mathrm{d}\mathbf{X}^\star_P(\mathbf{y_s}, t)}{\mathrm{d}t} = -\mathbf{V}(\mathbf{X}^\star_P, t) - F\frac{\mathbf{q}^\star_P(t)}{|\mathbf{q}^\star_P(t)|}$$

backward in time, where, $\mathbf{q}^\star_P(t)$ is the costate corresponding to each trajectory $\mathbf{X}^\star_P(t)$.

The numerical schemes used to solve (9)–(12) and their implementation over the full spatial domain are outlined in §C and detailed in (Lolla, 2012; Lolla et al, 2014c). §C also discusses the case when $\mathbf{y_f}$ lies on a shock line (see *Uniqueness* remark in §3.3).

4.2 Algorithm and Numerical scheme: Narrow Band

Since we are interested only in the evolution of the reachability front and not the behavior of ϕ^o away from the front, we can use a narrow band approach (Adalsteinsson and Sethian, 1995) in the forward propagation step above: (9) is then solved only within a band of points around the zero level set instead of the whole domain. Due to this, significant reduction in computational effort is achieved.

In this scheme, points within a band around the front are tagged as *alive* and points far away from the front are marked *far*. Points near the edge of the *alive* set are marked *close*. At each time step, (9) is solved for points in the *alive* set. Points from the *close* set that enter the *alive* set are assigned ϕ^o values using a fast marching method (Adalsteinsson and Sethian, 1995). When these points are brought into the *alive* set, the *close* set is updated. Similarly, points that leave the *alive* set are added to the *close* set. Since (9) is solved in a much smaller domain, the computational cost of the narrow band scheme is significantly lower than that of the regular level set method. Here, we implemented the narrow band scheme of Adalsteinsson and Sethian (1995).

4.3 Representation of ϕ^o

There are several possible representations of ϕ^o, whose evolution is governed by (9). Their theoretical and numerical properties are now outlined. The level set method does not place any strict restrictions on the choice of ϕ^o as long as it is Lipschitz continuous (Osher and Sethian, 1988; Russo and Smereka, 2000). The viscosity solution to the Cauchy problem (9) is unique and locally Lipschitz (Bressan, 2011; Tonon, 2011). If the forward evolution (9) is solved *exactly* (i.e. no numerical errors), any Lipschitz continuous ϕ^o will yield the correct evolution of the reachability front $\partial\mathcal{R}$ and the correct optimal path $\mathbf{X}^o_P(\mathbf{y_s}, t)$. However, the numerical solution of (9) is dependent on the specific choice of ϕ^o. Usually, ϕ^o is chosen to be the signed distance function ($\phi_\rho(\mathbf{x})$), due to its several favorable properties: it is smooth, and maintains gradients of fixed magnitude everywhere, especially close to the front. This leads to a more stable and accurate front evolution. Detrimental effects of the loss of this representation are well documented (Sussman et al, 1994; Chopp, 1993). Next, we describe how ϕ^o deviates from a signed distance field during the course of front evolution.

Classic level sets and signed distance functions. When $\mathbf{V}(\mathbf{x}, t)$ is identically zero, (9) reduces to the classic level set eq. (6). If ϕ^o is initialized to be the signed distance function, then $|\nabla\phi^o| = 1$ initially, wherever ϕ^o is differentiable. For the rate of change, we have

$$\begin{aligned}\frac{1}{2}\frac{\partial|\nabla\phi^o|^2}{\partial t} &= \nabla\phi^o \cdot \nabla\left(\frac{\partial\phi^o}{\partial t}\right) \\ &= -F\nabla\phi^o \cdot \nabla|\nabla\phi^o|\,,\end{aligned}$$

considering the cases where all derivatives are well-defined. Initially, since $|\nabla\phi^o| = 1$, $\frac{\partial|\nabla\phi^o|^2}{\partial t} = 0$. Hence, $|\nabla\phi^o| = 1$ at all future times. This means that eq. (6) (i.e. (9) with no external velocity field) theoretically preserves the signed distance property of ϕ^o. However, due to the numerical approximations, this property is gradually lost. This causes neighboring level sets to either bunch up (large gradients) or spread out (small gradients). This problem, in general, cannot be alleviated by using higher order schemes (Mulder et al, 1992).

Path planning level sets and signed distance functions. For general velocity fields, $\mathbf{V}(\mathbf{x}, t)$ is not identically zero and the level set is governed by (9). In this case, we obtain

$$\begin{aligned}\frac{1}{2}\frac{\partial|\nabla\phi^o|^2}{\partial t} &= \nabla\phi^o \cdot \nabla\left(\frac{\partial\phi^o}{\partial t}\right) \\ &= -F\nabla\phi^o \cdot \nabla|\nabla\phi^o| - \nabla\phi^o \cdot \nabla\left(\mathbf{V} \cdot \nabla\phi^o\right)\,.\end{aligned}$$

Even if $|\nabla\phi^o| = 1$ initially, the second term of the right-hand-side is non-zero in general. Thus, ϕ^o will not remain a signed distance field under (9), even with exact computations. Numerically, in the absence of large enough grid resolution, this can result in sizable errors in the computation of quantities such as $\nabla\phi^o$ etc. Thus, one needs to either sufficiently resolve the regions close to the front or maintain the gradients of ϕ^o within reasonable bounds. Methods for maintaining a signed distance representation may be found in (Chopp, 2009; Adalsteinsson and Sethian, 1999; Russo and Smereka, 2000). See Lolla et al (2014c) for further discussions and additional references.

4.4 Computational Cost

In this section, we quantify the asymptotic computational complexity of our path planning algorithm and highlight challenges in obtaining similar estimates for other common algorithms.

We solve (9) numerically using a finite-volume (FV) approach for both the full domain level set and the narrow band version. The asymptotic complexity of the algorithm is a function of the grid size. In this paper, we present results for 2D path planning and hence, (9) is solved on a 2D grid. Let us assume that there are roughly n grid points in each direction and a total of N grid points in the whole domain, i.e. $N = \mathcal{O}(n^2)$.

Cost of solving level set equation: We start first with the full domain level set. If (9) is solved in the full domain, the computational cost per time step is $\mathcal{O}(n^2)$ for any classic PDE solver. If a narrow band approach is used to solve (9), this cost reduces significantly to $\mathcal{O}(nd)$ per time step (Adalsteinsson and Sethian, 1995), assuming a bandwidth d. The number of time steps (K) needed is directly related to the optimal travel time: $K \approx T^o(\mathbf{y_f})/\Delta t$. Since $T^o(\mathbf{y_f})$ is not known a priori, it is not possible to compute K without solving (9) in the first place. Furthermore, since we use an explicit time integration scheme, Δt is chosen to satisfy the CFL condition (Osher and Fedkiw, 2003), making Δt inversely proportional to n. As a result, K increases in direct proportion to n.

Cost of re-initialization: Re-initialization of ϕ^o incurs significant expense. Its contribution towards the overall computational cost depends on its frequency (number of time steps without re-initialization) and on the scheme used. The procedure of computing the distance of every grid point to the level set front is an $\mathcal{O}(n^3)$ operation. This cost drops to $\mathcal{O}(n^2 \log n)$ if a fast marching method is employed (Sethian, 1999a). For the narrow band version, the cost of computing the distances of all points inside the narrow band to the front is $\mathcal{O}(nd^2)$. In each of these cases, the re-initialization cost is more than the corresponding level set cost (per time step). Due to this, it is essential to choose the re-initialization scheme and frequency with caution so that it does not dominate the overall computational cost.

Cost of other algorithms: It is more challenging to estimate the computational costs of the approximate algorithms discussed in §1, in part because they are iterative schemes and, in continuous settings, they provide optimal solutions only in infinite time. Most of these schemes do not have rigorous estimates of rates of convergence or computational cost. For example, the A* method computes approximate trajectories by restricting the vehicle motion onto a grid. It maintains an *open* list (points that can possibly lie on optimal path) and a *closed* list (points that are no longer in consideration) at every step. In addition, there is a sorted priority queue of path segments and estimates of total cost to reach the end point. Due to the dynamic flow-field, the cost of each arc becomes time-dependent. Since the optimal path may visit some points more than once, no grid point may be removed from the *open* list, i.e., no branches of the graph may be pruned. Hence, the worst case complexity of A* scales exponentially with the length of the optimal path. As a result, for realistic flows even in two or three dimensions and at the grid sizes needed to resolve them, the size of the A* search space becomes prohibitive.

Randomized methods like RRTs are quick in practice and their main utility lies in uniformly exploring high dimensional control spaces. Owing to the probabilistic nature of RRTs, it is challenging to obtain rigorous estimates of their cost for path planning in dynamic flows. See (Lolla, 2012) for a detailed discussion. We are not aware of published rigorous estimates of the computational costs of other approximate algorithms for time-optimal path planning in dynamic currents.

5 Applications

In this section, we illustrate our path planning algorithm by means of three sets of examples. The first set (§5.1) is based on a canonical vortex flow. This serves as a benchmark, allowing comparison to an analytical solution. In the second set (§5.2), we utilize more complex and realistic ocean flows to highlight the features of our algorithm. In the final set (§5.3), we consider specific test cases, which support the remarks given in §3.3.

5.1 Benchmark Application: Path planning in Rankine Vortex

In this application, we consider a vortex flow, characterized in polar coordinates as $\mathbf{V}(r,\theta,t) = v_\theta(r)\,\hat{\boldsymbol{\theta}}$, where $\hat{\boldsymbol{\theta}}$ is the unit vector in the circumferential direction and $v_\theta(r)$ is the flow-field speed, depending on the type of vortex. We are interested in computing the fastest time trajectory from $\mathbf{y_s} = \mathbf{0}$ to $\mathbf{y_f} : (r = R, \theta = 0)$. As earlier, let a first arrival time of the vehicle be $\widetilde{T}(\mathbf{y_f})$ and the optimal first arrival time be $T^o(\mathbf{y_f}) = T^\star(\mathbf{y_f})$.

Analytical solution for general flow $v_\theta(r)$:
Let the to-be-optimized velocity of the vehicle relative to the flow be $F_P(t)\,\hat{\mathbf{h}}(t) = F_r(t)\,\hat{\mathbf{r}} + F_\theta(t)\,\hat{\boldsymbol{\theta}}$, with $F_r(t)^2 + F_\theta(t)^2 \leq F^2$. The total velocity is

$$\frac{\mathrm{d}\mathbf{X}_P}{\mathrm{d}t} = F_r(t)\,\hat{\mathbf{r}} + \left[F_\theta(t) + v_\theta(r)\right]\hat{\boldsymbol{\theta}}\,, \tag{13}$$

with $\mathbf{X}_P(\mathbf{y_s}, 0) = \mathbf{0}$ and $\mathbf{X}_P(\mathbf{y_s}, \widetilde{T}(\mathbf{y_f})) = \mathbf{y_f}$. Separating the radial and angular components of $\frac{\mathrm{d}\mathbf{X}_P}{\mathrm{d}t}$ gives $\dot{r} = F_r(t)$ and $r\dot{\theta} = F_\theta(t) + v_\theta(r)$. Upon integrating the radial component we obtain

$$R = \int_0^{\widetilde{T}(\mathbf{y_f})} F_r(t)\,\mathrm{d}t \;\leq\; \int_0^{\widetilde{T}(\mathbf{y_f})} F\,\mathrm{d}t = F\,\widetilde{T}(\mathbf{y_f})\,,$$

implying that $\widetilde{T}(\mathbf{y_f}) \geq \frac{R}{F}$. Hence, R/F is a lower bound for $\widetilde{T}(\mathbf{y_f})$. We now generate a trajectory that satisfies (13) and meets this bound thereby proving that $T^o(\mathbf{y_f}) = R/F$. Such a trajectory can be generated by setting $F_r(t) = F$ and $F_\theta(t) = 0$. For this choice of the vehicle speed, we obtain $\dot{r} = F$ and $\dot{\theta} = \frac{v_\theta(r)}{r}$. Integration of these equations yields $r(t) = Ft$ and

$$\theta(R) = \theta_0 + \int_0^R \frac{v_\theta(r)}{Fr}\,\mathrm{d}r\,. \tag{14}$$

Here, θ_0 is the initial heading angle and may be computed using (14) since $\theta(R)$ is known from the coordinates of $\mathbf{y_f}$. Hence, the optimal control is

$$F_P^o(t)\,\hat{\mathbf{h}}^o(t) = F\,\hat{\mathbf{r}}, \quad \text{with} \quad \theta_0 = \theta(R) - \int_0^R \frac{v_\theta(r)}{Fr}\,\mathrm{d}r\,.$$

This optimal solution can also be obtained by using our level set algorithm. The only information needed from the forward evolution of the level set to solve (12) is the direction of the normals to the intermediate level set contours. In this problem, we could have guessed the shapes of the contours without solving (9). Since the flow-field is symmetric and purely circumferential, the zero level set contours are circles centered at the origin (see Fig. 3b) with their outward normals coinciding with radial directions ($\hat{\mathbf{n}}^o = \hat{\mathbf{r}}$). Using this observation, we may directly solve (12), starting from the heading $\hat{\mathbf{h}}^o = \hat{\mathbf{r}}$ at $\mathbf{y_f}$ to compute the initial heading angle θ_0 (where the normal to the point level set is undefined).

This problem is almost identical to crossing a river/jet in the fastest time. In order to do this, one needs to head normal to the flow at all times, so that the maximum component of the vehicle's velocity is directed towards the opposite bank (Lolla et al, 2012). Similarly, in our case one needs to steer normal to the streamlines of the flow (i.e. $\hat{\mathbf{r}}$) to obtain the fastest time path.

Rankine Vortex Solution
We exemplify our algorithm with a Rankine vortex flow, $v_\theta(r) = \frac{\Gamma r}{2\pi\sigma^2}$, which resembles a solid body rotation of the fluid and is seen in many practical vortex flows. Γ is the total circulation around the origin and σ is the radius of the vortex. Here we use non-dimensional values, $\Gamma = 20$, $\sigma = 1.5$ and $F = 1$. The coordinates of $\mathbf{y_f}$ are ($R = 1$, $\theta = 0$). From (14), the initial heading angle is $\theta_0 = -\frac{\Gamma R}{2\pi F\sigma^2} \approx -1.41$ rad $\approx -81.1\,^\circ$ and the optimal trajectory is

$$r^\star(t) = Ft\,, \quad \theta^\star(t) = \frac{\Gamma(Ft - R)}{2\pi F\sigma^2}\,. \tag{15}$$

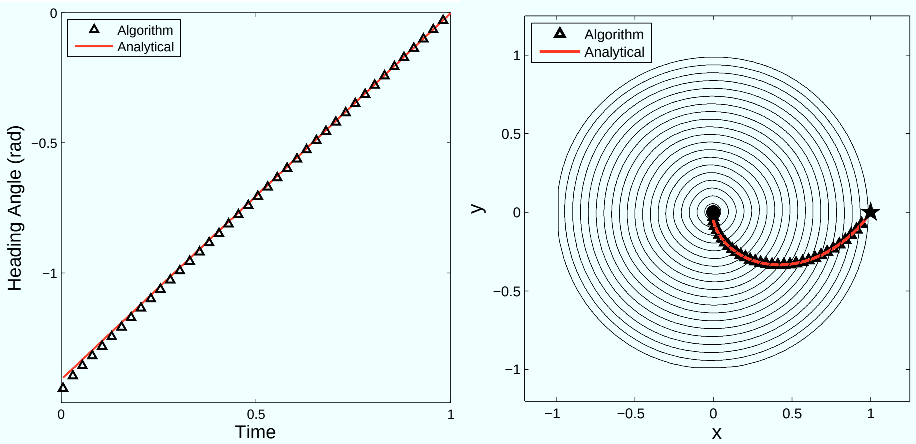
Shapes of the zero level set contours at different times and the optimal trajectory obtained by solving (12) are plotted in Fig. 3b. A 200×200 grid and a time step of 10^{-3} are used to solve (9), with open boundary conditions on ϕ^o (see §C for more on boundary conditions). Fig. 3a compares the headings predicted by the level set algorithm with their analytical values and provides evidence that our algorithm works correctly. Through this example, we emphasize that the only information needed from the solution of (9) is the time evolution of the zero level set front. If the level set contours can be determined a priori, only (12) needs to be solved.

5.2 Path Planning in More Realistic Flows

In this section, we apply our path planning methodology to more complex but numerically simulated flow-fields. These examples also illustrate certain unique features and capabilities of our approach.

5.2.1 Double-Gyre Flow

The wind-driven double-gyre flow is modeled using a barotropic single layer-model in a square basin of size $L = 1$ described in detail in (Dijkstra and Katsman, 1997; Simmonet et al, 2009) (see also (Pedlosky, 1998),(Cushman-Roisin and Beckers, 2010)). The intent is to simulate the idealized near-surface double-gyre ocean circulation



(a) (b)

Fig. 3: (a) Optimal heading angles, (b) optimal path and circular intermediate reachability fronts of a vehicle navigating in a Rankine vortex flow. Black: path predicted by level set algorithm, Red: analytical, i.e. governed by (15).

at mid-latitudes. The mid-latitude easterlies and trade winds in the northern hemisphere drive a cyclonic gyre and an anticyclonic gyre, and the corresponding zonal jet in between. This eastward jet would correspond to the Gulf Stream in the Atlantic and to the Kuroshio and its extension in the Pacific. This idealized flow is modeled by the non-dimensional equations of motion

$$\frac{\partial u}{\partial t} = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}}\Delta u - \frac{\partial\left(u^2\right)}{\partial x} - \frac{\partial\left(uv\right)}{\partial y} + fv + a\tau_x, \tag{16a}$$

$$\frac{\partial v}{\partial t} = -\frac{\partial p}{\partial y} + \frac{1}{\mathrm{Re}}\Delta v - \frac{\partial\left(vu\right)}{\partial x} - \frac{\partial\left(v^2\right)}{\partial y} - fu + a\tau_y, \tag{16b}$$

$$0 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}, \tag{16c}$$

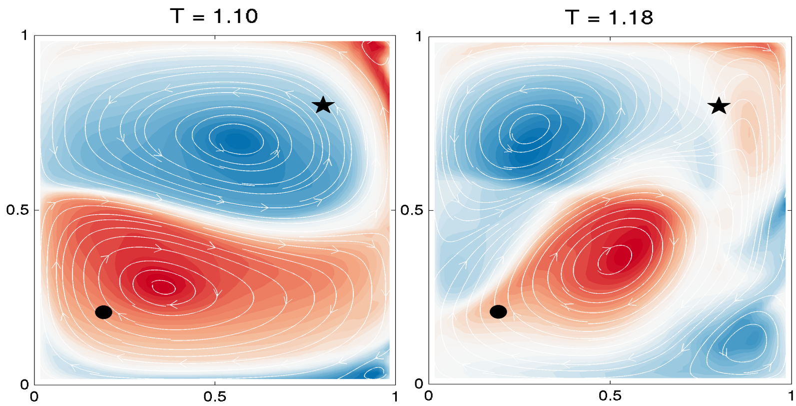
where Re is the flow Reynolds number taking values from 10 to 10^4, $f = \tilde{f} + \beta y$ the non-dimensional Coriolis coefficient, and $a = 10^3$ the strength of the wind stress. In non-dimensional terms, we use $\tilde{f} = 0$, $\beta = 10^3$. The flow in the basin is forced by an idealized steady zonal wind stress, $\tau_x = -\frac{1}{2\pi}\cos 2\pi y$ and $\tau_y = 0$.

Free slip boundary conditions are imposed on the northern and southern walls ($y = 0, 1$) and no-slip boundary conditions on the eastern and western walls ($x = 0, 1$). A 64×64 grid and a non-dimensional time step of 10^{-4} are used to solve both (16) (generation of flow-field) and (9) (forward level set evolution). Open boundary conditions (see §C) are implemented on all the walls for (9). In what follows, we present results for Re = 150.

The governing flow-field equations (16) are solved using a second order accurate Navier-Stokes solver, which is a component of a modular finite volume framework (Ueckermann and Lermusiaux, 2011). The framework uses a uniform, two-dimensional staggered C-grid for the spatial discretization. The diffusion operator in (16) is discretized using a second order central difference scheme. The advection operator is discretized using a Total Variation Diminishing (TVD) scheme with the monotonized central (MC) limiter (Van Leer, 1977). The time discretization uses a first-order accurate, semi-implicit projection method, where the diffusion and pressure terms are treated implicitly, and the advection is treated explicitly (Ueckermann et al, 2013). In Fig. (4), we show a few snapshots of the computed flow-field streamlines, overlaid on a color plot of vorticity, at different non-dimensional times. The forward evolution (9) is solved using the numerical scheme described in §C.

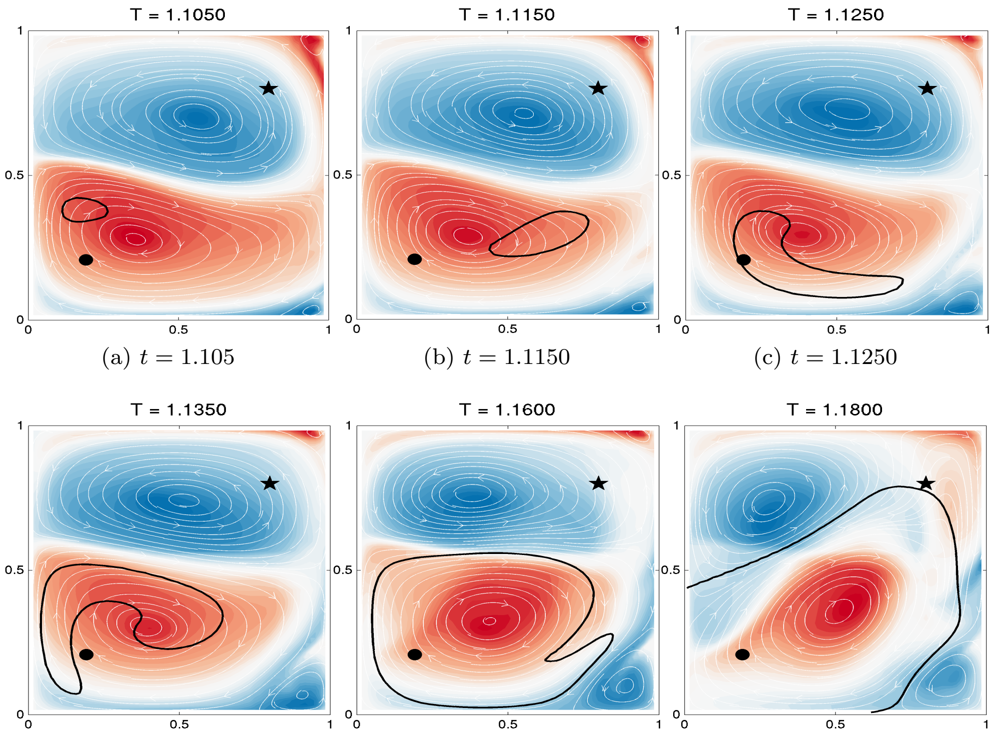
In this example, we (i) examine the performance of our methodology for path planning in a strong and dynamic flow-field, and (ii) illustrate an example to determine if a vehicle can reach a given end point within a specified time limit. Here, we choose $\mathbf{y_s} = (0.2, 0.2)$ and $\mathbf{y_f} = (0.8, 0.8)$. The vehicle is allowed to move after an offset time $t_s = 1.10$, i.e. the flow-field experienced by the vehicle at the start of its motion is the flow-field at time t_s. Fig. (4a) depicts the points $\mathbf{y_s}$, $\mathbf{y_f}$ and also the flow-field at the time t_s.

Fig. (5) shows the evolution of the zero level set front when $F = 5$. The optimal trajectory obtained by solving (12) is plotted in Fig. (6). Due to the strong flow-field, the vehicle has to perform two revolutions



(a) $t = 1.10$ (b) $t = 1.18$

Fig. 4: Snapshots of the double-gyre flow-field at different times: flow streamlines (white) overlaid on color plots of vorticity (range: [-15,15]). The start (circle) and end (star) points are also depicted. All physical quantities shown are non-dimensional.



(a) $t = 1.105$ (b) $t = 1.1150$ (c) $t = 1.1250$

(d) $t = 1.1350$ (e) $t = 1.1600$ (f) $t = 1.1800$

Fig. 5: Time evolution of the reachability front (black) in the double-gyre flow-field for a start time $t_s = 1.10$ and relative speed $F = 5$. The evolution of the flow-field, colored by vorticity, is also shown.

around the lower eddy before it finds a favorable current that drives it towards $\mathbf{y_f}$.

Using this double-gyre flow-field, we study another important aspect of path planning which is to determine whether a vehicle can reach a given end point within a specified time limit, $T_{\max}$. For this example, we use a starting time $t_s = 0.4$. We examine the effect of varying F, setting all other parameters the same as before. If we set $F = 8$, the optimal travel time is computed to be 0.0343 (see Fig. 7b). Upon reducing F to 6, the optimal travel time increases to 0.0856 - more than twice the earlier value. The optimal trajectory is

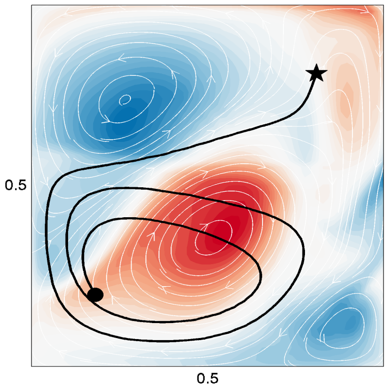


Fig. 6: Time-optimal trajectory (black) from $\mathbf{y_s} = (0.2, 0.2)$ to $\mathbf{y_f} = (0.8, 0.8)$ in the double-gyre flow overlaid on the final flow-field, colored by vorticity.

also significantly different. Our level set methodology can predict if a vehicle can reach $\mathbf{y_f}$ within time $T_{\max}$. The reachability front at time $t = 0.035$ for $F = 6$ is shown in Fig. 7c. Since the front has not yet reached $\mathbf{y_f}$, we conclude that it is not possible for the vehicle to reach $\mathbf{y_f}$ within $T_{\max} = 0.035$. In the general case, (9) needs to be solved until the front reaches $\mathbf{y_f}$, or until time $T_{\max}$, whichever is smaller. In the first case, the optimal trajectory can be computed, and in the second, the algorithm terminates, providing the reachability set at time $T_{\max}$.

5.2.2 Flow Past Circular Island: all-to-all broadcast

We now consider the case of open flow in a smooth ocean channel with a circular island obstacle (see Fig. 8). This is a highly unsteady flow-field that exhibits varied vortex shedding (a function of the Re) in the wake. Through this example, we: (i) illustrate performance for swarms of vehicles in a strong and dynamic flow-field, (ii) demonstrate how obstacles to the flow (and vehicle) are naturally handled by the algorithm and, (iii) illustrate that the algorithm can be parallelized when paths for multiple vehicles have to be planned.

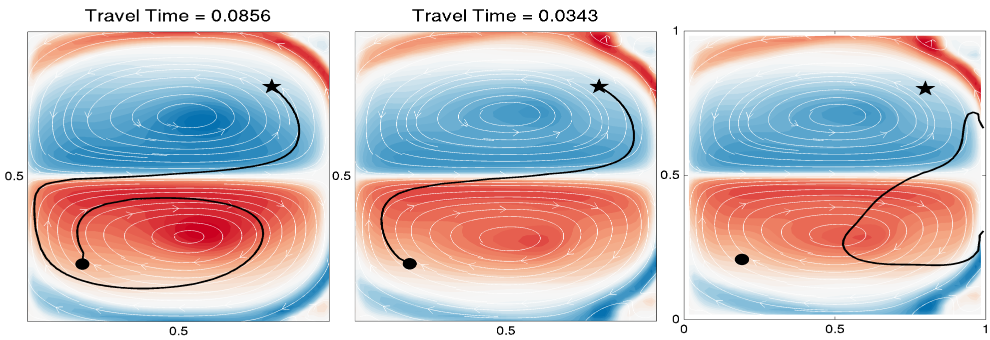
In this example, 11 swarms (black circles) of 11 vehicles each are initially located upstream of the obstacle. Each swarm has one designated leader who must receive information from representative vehicles of each of the other 10 swarms. The information exchange must take place in the fastest time, at specific locations downstream (shown by colored markers in Fig. 9), where swarms are reformed. Each leader travels to the end point corresponding to its swarm and each follower travels to one of the other end points. This situation is an *all-to-all broadcast* in distributed computing and communication, where every node broadcasts its information to all other nodes. Thus, the goal for these vehicles is to reach their end points in the fastest time, by utilizing (or avoiding) the multi-scale flow structures in their path. In addition, none of the vehicles should collide with the cylindrical obstacle, i.e. the paths of all the vehicles should be both safe and optimal.

In the example shown, Re = 1000. The flow is driven by a deterministic uniform-flow at the inlet (left of domain), with slip velocity boundary conditions at the top and bottom, and open boundary conditions at the outlet (see Fig. 8). The governing flow-field equations are given by (16), without the Coriolis and wind stress terms (i.e. $f = 0$, $\tau_x = \tau_y = 0$). The obstacle in the domain is handled by masking out the appropriate region in the mesh. A 200×30 grid and a non-dimensional time step of 5×10^{-4} are used in solving both (16) (flow-field) and (9) (forward evolution). Snapshots of the resultant flow-field at different times are shown in Fig. (9).

We choose $F = 0.5$ and evolve a level set (eq. (9)) corresponding to each of the 11 start points. In solving (9), we use mask the grid points that lie under the obstacle (see §C). Open boundary conditions are imposed on ϕ^o at all other domain edges.

Fig. (10) shows the time evolution of level set fronts for three different start points overlaid on plots of flow-fields, colored by vorticity. We see that the level set fronts do not penetrate the obstacle, but 'wrap' around it. This feature of level sets leads to collision-free (safe) trajectories. The level set fronts from each start point are evolved until every end point has been crossed. The crossing times of each end point are recorded because backtracking (eq. (12)) is performed from the time each end point is reached. The optimal vehicle trajectories corresponding to each start point are plotted in Fig. (11). As expected, none of the paths pass through the obstacle. Fig. (11j) contains all of the vehicle paths, clearly illustrating the all-to-all broadcast, with connections from each start point to every end point.

This example shows that our methodology generates collision-free vehicle trajectories in addition to time-optimal paths, at no additional computational expense. Also, the number of level sets that need to be evolved depends on the number of different start points, and not on the number of end points. Paths to every end point corresponding to a single start point can be planned by evolving just one level set field. In the case of multiple end point points, the level set needs to be evolved until all of the end points have been reached. Thus, this algorithm can be efficiently parallelized to independently compute optimal vehicle tracks from multiple start points. Other examples of path planning in



(a) $F = 6$ (b) $F = 8$ (c) $F = 6$

Fig. 7: Time-optimal trajectories for two vehicles in the double-gyre flow-field. (a) The first vehicle ($F = 6$) takes 0.0856 units of time to reach the end point whereas (b) the second vehicle ($F = 8$) takes only 0.0343 units of time. (c) The reachability front at time $t = 0.035$ for the slower vehicle ($F = 6$).

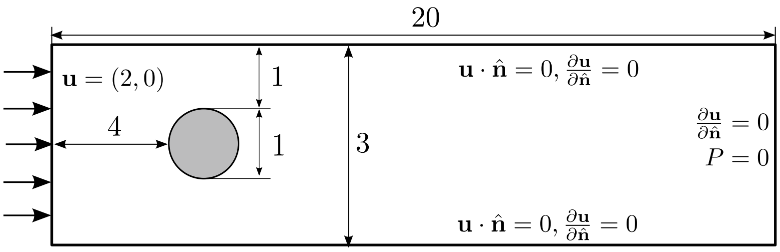


Fig. 8: Schematic of flow past circular island test case. Flow enters the left edge of the domain at a non-dimensional speed of 2 and encounters a circular island, leading to the formation of vortices downstream of the island.

other flows can be found in (Lolla, 2012; Lolla et al, 2012; Lermusiaux et al, 2014).

5.3 Path Planning Examples Complementing §3.3

5.3.1 Applicability of modified Eikonal equation

We consider a 1-D problem with $y_s = 0$, $y_f = 4$ and $F = 1$. Let $\mathbf{V}(x, t) = -2\sin(\pi t)\,\hat{\mathbf{i}}$ (see Fig. 12). This is an oscillating flow-field in one dimension.

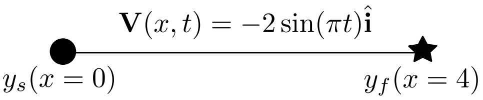


Fig. 12: 1D flow-field and domain

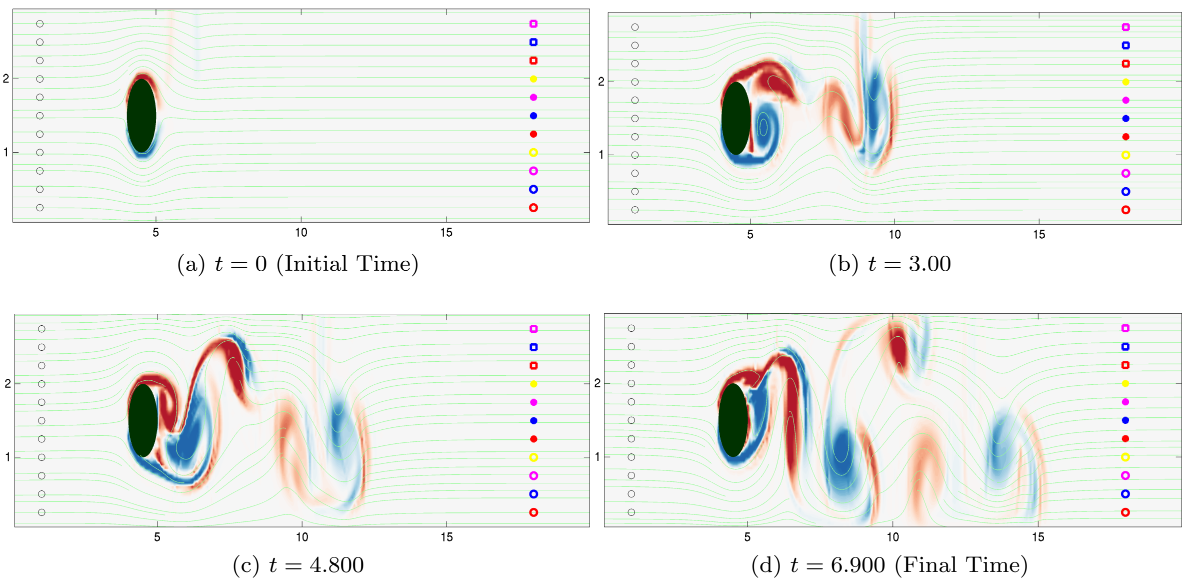
Since its motion is restricted to the x-axis, the vehicle has only two heading choices at any time: it can either be steered to the right or to the left. From Theorem 4, only vehicles that are steered at maximum (relative) speed F can remain on the reachability front. In this case, the reachability front consists of only two points, corresponding to positions of two vehicles, one steered to the left and the other to the right at relative speed F. Since $y_f > y_s$ and the flow is spatially uniform, the optimal trajectory $X_P^\star(y_s, t)$ is realized when the vehicle always moves to the right at relative speed F and satisfies

$$\frac{\mathrm{d}X_P^\star}{\mathrm{d}t} = F + \mathbf{V}(X_P^\star, t) \cdot \hat{\mathbf{i}} = 1 - 2\sin(\pi t)\,. \qquad (17)$$

Integrating (17) with initial condition $X_P^\star(y_s, 0) = 0$ yields

$$X_P^\star(y_s, t) = t + \frac{2}{\pi}\left(\cos(\pi t) - 1\right)\,. \qquad (18)$$

This continuous trajectory is plotted in blue in Fig. 13a. Using $X_P^\star$, $T^o(y)$ can be computed as $T^o(y) = \min_t\{t : X_P^\star(y_s, t) = y\}$. Note that the argument y should not be confused with the ordinate; here it represents a general point in the 1-D domain. $T^o(y)$ is plotted in red in the same figure. We can clearly see the discontinuity in T^o near points 0.08 and 2.08. This happens because at certain times, the vehicle experiences a strong flow adverse to its rightward motion due to which, it is forced to reverse its trajectory until a favorable current advects it towards y_f. As a result, the vehicle visits some



(a) $t = 0$ (Initial Time)

(b) $t = 3.00$

(c) $t = 4.800$

(d) $t = 6.900$ (Final Time)

Fig. 9: Snapshots of flow-field behind the circular island at different times. Streamlines are overlaid on the flow-field colored by vorticity (range: [-15,15]).

points (such as $y = 0.08$) in its optimal path *more than once*. At such points where $T^o(y)$ is not continuous, the gradient $\nabla T^o(y)$ is undefined and (34) does not admit a continuous viscosity solution. This makes it necessary to keep track of subsequent arrival times (in addition to the first one) to compute the optimal path. Solving (9) gives the optimal solution, even with strong adverse flow-fields since the level set front always corresponds to the reachability front. By predicting and tracking this front, our algorithm records multiple arrival times, providing the solution for both weak and strong flows.

Let us consider the same 1D example but now with a flow-field, $\mathbf{V}(x,t) = -0.95\sin(\pi t)\,\hat{\mathbf{i}}$. This flow is not strong since its magnitude is at most 0.95, which is smaller than F. The optimal trajectory in this case is plotted in blue in Fig. 13b. The optimal first arrival time field $T^o(y)$ is superposed in red. Here, these curves are identical since the vehicle does not experience currents of speeds larger than F along its path. In this case, $T^o(y)$ is the continuous viscosity solution of (34).

5.3.2 Determination of Starting Time

In addition to the optimal control, the level set methodology can also be used to determine *when* vehicles must be deployed to reach their end points in the quickest time. In most of the previous examples, the vehicle starts its motion at time $t_s = 0$. In some cases, if the vehicle is allowed to start at a later time (unknown a priori) it may be able to arrive at the end point sooner than if it starts at $t_s = 0$. This can happen if the vehicle experiences strong adverse currents at the start which advect it away from the end point. In such cases, the vehicle may reach the end point sooner if deployed (from a ship, for example) after the adverse current has passed.

We now present an example where this situation occurs, and how our approach can be used to determine t_s. We use the same 1-D example as in §5.3.1. The flow-field is given by $\mathbf{V}(x,t) = -2\sin(\pi t)\,\hat{\mathbf{i}}$. Here, we set $F = 1$, $y_s = 0$, and $y_f = 2$. As seen earlier, the optimal trajectory satisfies (17). Let us assume that the vehicle is deployed at a variable start time $t_s \geq 0$, so that $X_P^{\star}(y_s, t_s) = 0$. Our goal now, is to minimize the arrival time at $y_f = 2$ by a suitable choice of t_s. Integrating (17) and setting the limits yields

$$X_P^{\star}(y_s,t) = (t - t_s) + \frac{2}{\pi}\left(\cos(\pi t) - \cos(\pi t_s)\right), \quad t \geq t_s\,. \tag{19}$$

This family of optimal trajectories and corresponding optimal arrival times at y_f can be computed for different values of $t_s \geq 0$. Sample trajectories corresponding to starting times $t_s = 0, 0.5, \frac{5}{6}, 1.5, 2$ are plotted in Fig. 14a.

We observe that the trajectory corresponding to $t_s = 0$ reaches y_f *later* than the one corresponding to $t_s = 0.5$. This is because a strong flow in the $-\hat{\mathbf{i}}$ direction for $\frac{1}{6} \leq t \leq \frac{5}{6}$ forces the vehicle to reverse its path. The optimal t_s here is when the flow speed reduces to F, which occurs at $t_s = 1 - \frac{1}{\pi}\sin^{-1}(0.5) = \frac{5}{6}$. In Fig. 14b, the arrival times at $y_f = 2$ are plotted as a

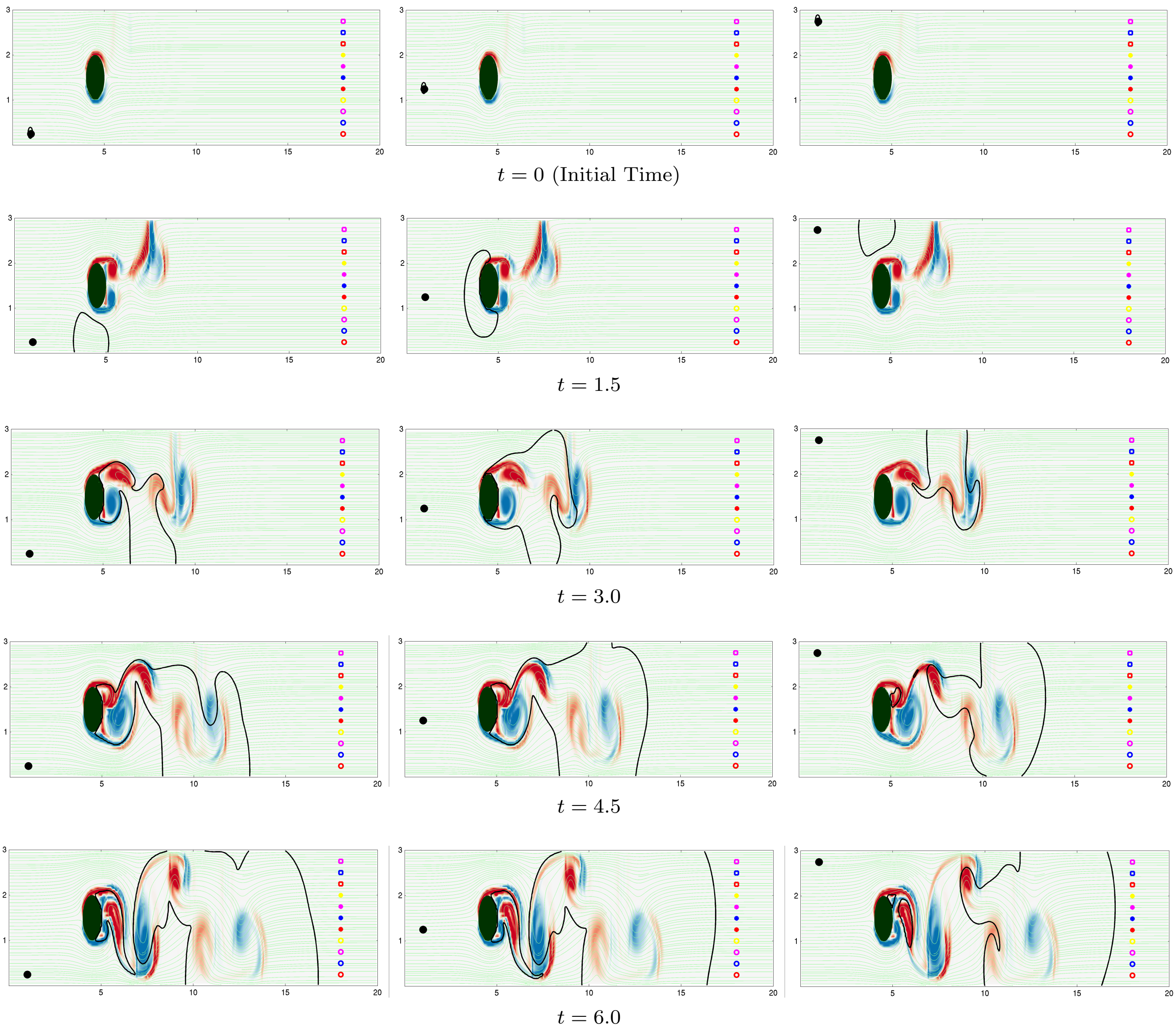


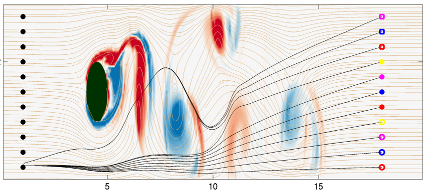
Fig. 10: *Flow past circular island*: time evolution of level set front corresponding to three different start points (marked in black). In all cases, the level sets 'wrap' around the island and never pass through it.

function of t_s. This curve clearly shows that the fastest arrival time is for $t_s = \frac{5}{6}$.

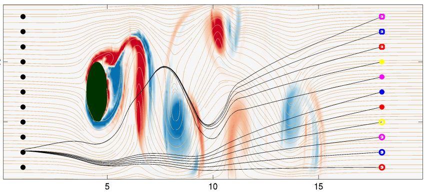
Our methodology can compute the optimal t_s by keeping track of reachability fronts corresponding to several starting times. Instead of one reachability front, we will now track an ensemble of fronts, each for one choice of t_s. The starting time corresponding to the level set front that reaches the end point fastest, is the optimal starting time. Once this is known, the optimal path can be calculated by solving the backtracking equation. Although this approach requires solving an ensemble of independent forward level set eqs. (9), it is inexpensive due to the low computational cost. The algorithm also lends itself to easy implementation of heuristics to decide when to evolve new level set fronts in order to reduce the computational cost for this problem. For example, one admissible heuristic could be to evolve level sets when the flow at the start point is favorable (directed towards the end point).

5.3.3 Multiple Optimal Paths

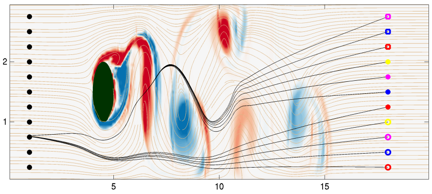
In some situations, for a given problem configuration $(\mathbf{y_s}, \mathbf{y_f}, F, \mathbf{V}(\mathbf{x}, t))$, there may exist multiple optimal trajectories with the same travel time. We now present such a scenario, showing that even though two end points are nearby each other in space, the optimal path to these points can be very different. The end point at



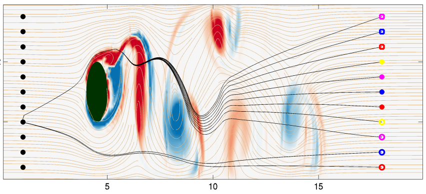
(a) Start Point 1



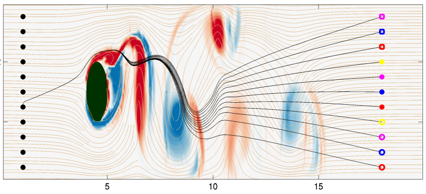
(b) Start Point 2



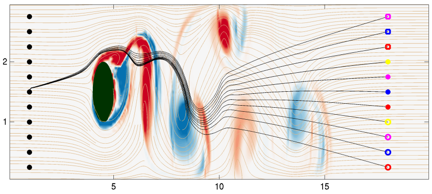
(c) Start Point 3



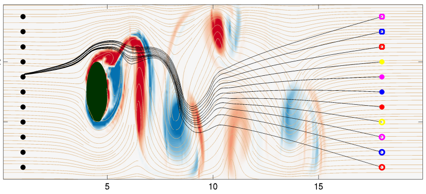
(d) Start Point 4



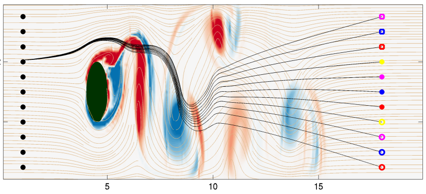
(e) Start Point 5



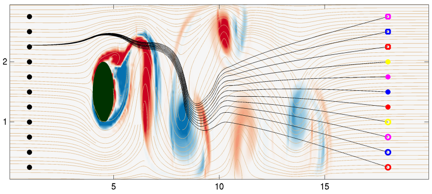
(f) Start Point 6



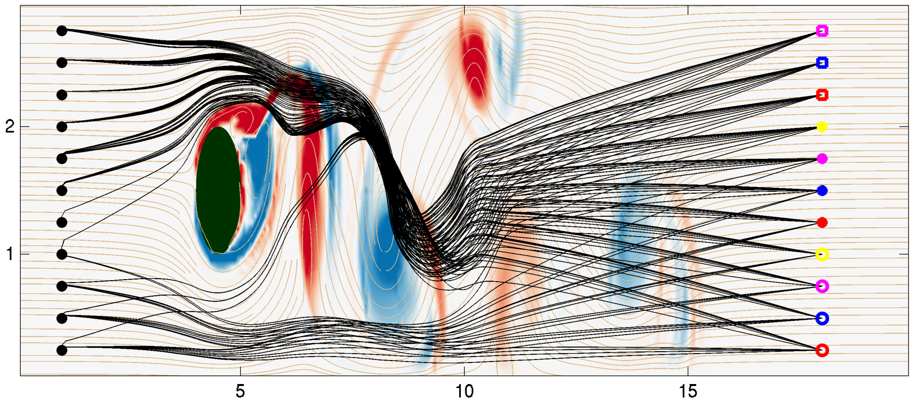
(g) Start Point 7



(h) Start Point 8



(i) Start Point 9



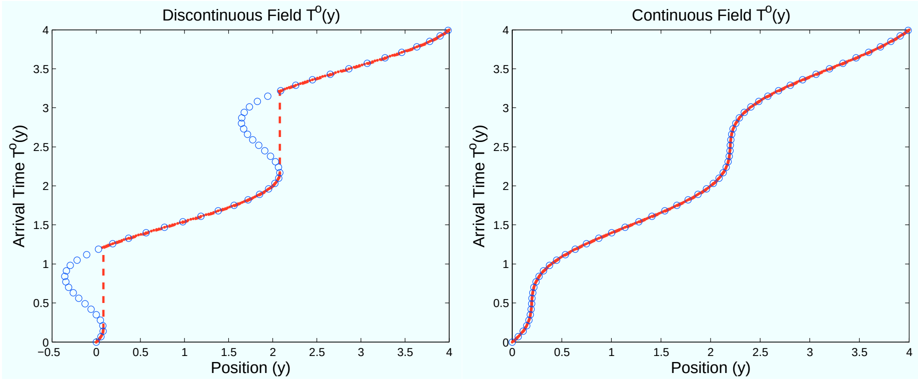
(j) All Start Points 1-11 and time-optimal paths

Fig. 11: *Flow past circular island: all-to-all broadcast.* Safe and time-optimal trajectories corresponding to different start points. Vehicle paths (black) are overlaid on the flow-field, colored by vorticity (shown in range: [-15,15]).

the limit between the above two points admits two possible optimal paths. Theoretically, these are points at which characteristics of (9) merge, and are quite general (e.g. lines in 2D, surfaces in 3D etc.). We consider the example of a jet flow in a 2D domain (Lolla et al, 2012).

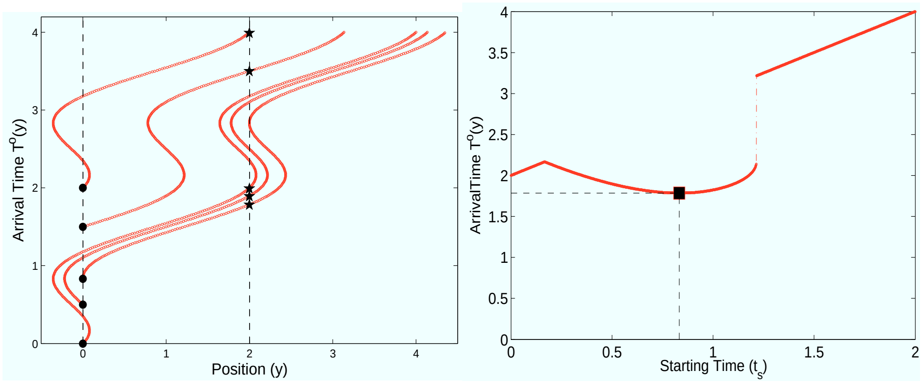
In this problem, two vehicles ($F = 1$) start at the same position $\mathbf{y_s} = (1, 1)$ and same time, $t_s = 0$. Their end points are $\mathbf{y_f^1} = (2, 0.8)$ and $\mathbf{y_f^2} = (1.95, 0.75)$. The time-optimal trajectories are plotted in Fig. 15. We observe that even though $\mathbf{y_f^1}$ and $\mathbf{y_f^2}$ are nearby each other, the optimal paths are very different: one of the trajectories is a straight line from start to end and is not affected by the jet while the second one makes use of the jet to minimize travel time.

The viscosity solution to (9) allows the formation of singularities (e.g. corners) in the level set front (Lolla, 2012; Sethian, 1999b). This behavior occurs in this example: there exists a 'shock' line formed by the level sets to the end point on which, multiple optimal paths exist. This line is marked in Fig. 15. The evidence for existence of such lines can be obtained by solving (9) alone, without the backtracking (12). Several other sim-



(a) $\mathbf{V}(\mathbf{x}, t) = -2\sin(\pi t)\,\hat{\mathbf{i}}$ (b) $\mathbf{V}(\mathbf{x}, t) = -0.95\sin(\pi t)\,\hat{\mathbf{i}}$

Fig. 13: Optimal vehicle trajectory (blue) and optimal first arrival time field, $T^o(y)$ (red) for the 1-D flow in §5.3.1. In (a), the adverse flow-field leads to discontinuities in $T^o(y)$. In (b), the flow is never adverse to vehicle motion and $T^o(y)$ is continuous.



(a) (b)

Fig. 14: (a) Optimal trajectories for different starting times t_s (denoted by filled circles). The first arrival time at $y_f = 2$ for each trajectory is marked by filled stars. A smaller t_s does not necessarily lead to a smaller arrival time at y_f. (b) Plot of first arrival times at $y_f = 2$ versus different t_s. The minimum arrival time is obtained for $t_s = \frac{5}{6}$ - corresponding arrival time marked in black.

ilar examples can be constructed in which there exist multiple optimal paths to some end points.

6 Conclusions

In this paper, we have developed a novel methodology to predict the time-optimal trajectories of multiple vehicles navigating in strong and dynamic flow-fields, such as ocean currents. To do so, we derived a modified level set equation that governs the evolution of a reachability front. The reachability front is then evolved from the vehicle start point until it reaches the end point, combining nominal vehicle motion due to steering and advection due to the flow. The optimal trajectory and vehicle heading directions are then extracted from the time history of the evolution of the reachability front by solving a backtracking problem. The approach is interdisciplinary: it is inspired by ideas in fluid mechanics, ocean science and computational sciences (level set and numerical methods) and applies them to path planning, which has roots in robotics and optimal control.

As the methodology is based on solving partial differential equations, it is rigorous and obviates the need for heuristics. We illustrated the theory and schemes using analytical flows as well as unsteady double-gyre flows driven by wind stress and flows behind a circular island. The latter case showed that stationary obstacles that affect both the flow and the vehicle motions can be easily accommodated. The extension to moving

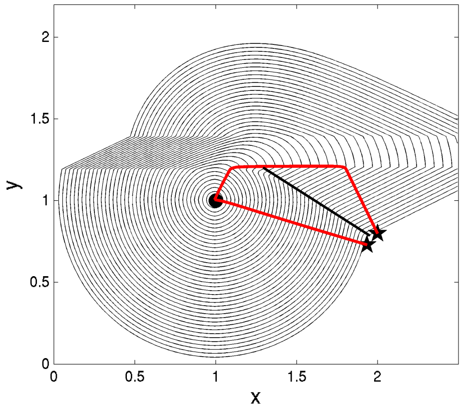


Fig. 15: Optimal paths (red) overlaid on intermediate level set contours (black) for a jet flow (§5.3.3). Nearby end points $(2, 0.8)$ and $(1.95, 0.75)$ produce very different optimal paths. The 'shock' line (thick black) is the set of points to which multiple optimal paths exist.

obstacles and forbidden regions (which affect only vehicle motions and not the flow-field) is straightforward and has discernible societal applications (e.g. ships, airplanes). Though we have only focused on underwater path planning here, our methodology is general and applies to many other flows (e.g. atmospheric, microscopic) and vehicles (e.g. UAVs, bio-robots). We have also studied several other idealized and realistic scenarios, including cases with moving obstacles and forbidden regions (Lolla et al, 2012, 2014a,c; Lermusiaux et al, 2014).

As we illustrated, the low computational cost allows the use of our methodology to plan paths for multiple vehicles simultaneously. Coordinated path planning, which has been extensively studied and developed recently (Leonard and Fiorelli, 2001; Paley et al, 2008; Leonard et al, 2007), renders certain types of missions possible, which otherwise, could not be executed by single-vehicle systems. A possible future direction is to integrate our approach with existing schemes for efficient and optimal coordination. Secondly, in this work, we have assumed the flow-fields to be exactly known. In some cases, such as oceanic applications, the predicted flows are uncertain. It is then possible to extend our methodology to plan paths in a stochastic setting by optimizing suitable path statistics (Lolla et al, 2014c). As more information about the forecasted flow-field becomes available, the paths can be updated using onboard routing. Here, we have focused only on continuous trajectory optimization problems. In some practical situations such as those involving underwater gliders (Lolla, 2012), communication between the glider and the controller may only be possible at discrete times (Schneider and Schmidt, 2010; Hollinger et al, 2012; Cheung et al, 2013; Cheung and Hover, 2013). In such realistic cases, we need discrete control averaged over time. This is discussed in (Lolla et al, 2014a). Finally, we can also explore the extension of our methodology to plan paths that optimize the energy spent by the vehicles (Subramani, 2014), instead of travel time.

7 Acknowledgements

We are very thankful to the MSEAS group, in particular Mr. K. Yiğit and Mr. T. Sondergaard for helpful discussions. We thank Dr. T. Duda for suggestions on an earlier version of this manuscript and two anonymous reviewers for their comments. We are grateful to the Office of Naval Research for support under grants N00014-09-1-0676 (Science of Autonomy A-MISSION) and N00014-12-1-0944 (ONR6.2) to the Massachusetts Institute of Technology (MIT). MPU and PFJL also thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for the Postgraduate Scholarship partially supporting the graduate studies and research of MPU at MIT.

A Preliminaries

In this Appendix, we describe some of the relevant definitions and terminology needed for the theoretical results. Most of the material presented in this §A may be found in (Bardi and Capuzzo-Dolcetta, 2008; Clarke et al, 1998; Cannarsa and Sinestrari, 2004; Frankowska, 1989; Bressan, 2011). In what follows, we let $n \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^n$ be an open set and $\xi : \Omega \to \mathbb{R}$.

Remark 1 *Let $\xi \in \mathcal{C}(\Omega)$. Let $\partial_+\xi(\mathbf{x}_0)$ and $\partial_-\xi(\mathbf{x}_0)$ denote the sets of super- and sub-differentials (Bardi and Capuzzo-Dolcetta, 2008; Clarke et al, 1998) of ξ at $\mathbf{x}_0$. Then $\mathbf{q} \in \partial_+\xi(\mathbf{x}_0)$ (resp. $\partial_-\xi(\mathbf{x}_0)$) if and only if there exists a function $\gamma \in \mathcal{C}^1(\Omega)$ such that $\gamma(\mathbf{x}_0) = \xi(\mathbf{x}_0)$, $\nabla\gamma(\mathbf{x}_0) = \mathbf{q}$ and the function $\gamma - \xi$ has a strict local minima (resp. maxima) at $\mathbf{x}_0$.*

Definition 1 ***(Generalized Gradient.)*** *Let ξ be locally Lipschitz at $\mathbf{x}_0$. For any $\mathbf{u} \in \mathbb{R}^n$, let $\xi^g(\mathbf{x}_0; \mathbf{u})$ denote the* generalized directional derivative *of ξ at $\mathbf{x}_0$ (Clarke et al, 1998). The set of* generalized gradients *of ξ at $\mathbf{x}_0$ is the non-empty set*

$$\partial\xi(\mathbf{x}_0) = \{\mathbf{q} \in \mathbb{R}^n : \forall\, \mathbf{u} \in \mathbb{R}^n, \mathbf{q} \cdot \mathbf{u} \le \xi^g(\mathbf{x}_0; \mathbf{u})\} . \qquad (20)$$

Definition 2 ***(Regular Function.)*** *ξ is said to be* regular *at $\mathbf{x}_0 \in \Omega$ if it is Lipschitz near $\mathbf{x}_0$ and admits directional derivatives $\xi^d(\mathbf{x}_0; \mathbf{u})$ for all $\mathbf{u} \in \mathbb{R}^n$, with $\xi^g(\mathbf{x}_0; \mathbf{u}) = \xi^d(\mathbf{x}_0; \mathbf{u})$.*

Properties of Regular Functions.

1. If ξ is continuously differentiable at $\mathbf{x}_0$, then it is regular at $\mathbf{x}_0$. Furthermore, $\xi^d(\mathbf{x}_0; \mathbf{u}) = \nabla\xi(\mathbf{x}_0) \cdot \mathbf{u} = \xi^g(\mathbf{x}_0; \mathbf{u})$ for all $\mathbf{u} \in \mathbb{R}^n$.
2. If ξ is convex and Lipschitz near $\mathbf{x}_0$, then it is regular at $\mathbf{x}_0$.

3. Let ξ be regular at $\mathbf{x}_0 \in \Omega$. Then,

$$\partial_- \xi(\mathbf{x}_0) = \partial \xi(\mathbf{x}_0)\,. \tag{21}$$

Definition 3 ***(Viscosity Solution.)*** *Let $F \geq 0$ and let $\mathbf{V}(\mathbf{x}, t)$ satisfy assumptions (3)–(4). Consider the Hamilton-Jacobi equation*

$$\frac{\partial \phi}{\partial t} + F|\nabla \phi| + \mathbf{V}(\mathbf{x}, t) \cdot \nabla \phi = 0 \quad in \quad \Omega \times (0, \infty)\,, \tag{22}$$

with initial conditions

$$\phi(\mathbf{x}, 0) = \nu(\mathbf{x})\,, \tag{23}$$

where $\nu : \Omega \to \mathbb{R}$ is Lipschitz continuous. A function $\phi \in \mathcal{C}(\Omega \times [0, \infty))$ is a viscosity subsolution *of (22) if for every $(\mathbf{x}, t) \in \Omega \times (0, \infty)$ and $(\mathbf{q}, p) \in \partial_+ \phi(\mathbf{x}, t)$,*

$$p + F|\mathbf{q}| + \mathbf{V}(\mathbf{x}, t) \cdot \mathbf{q} \leq 0\,. \tag{24}$$

A function $\phi \in \mathcal{C}(\Omega \times [0, \infty))$ is a viscosity supersolution *of (22) if for every $(\mathbf{x}, t) \in \Omega \times (0, \infty)$ and $(\mathbf{q}, p) \in \partial_- \phi(\mathbf{x}, t)$,*

$$p + F|\mathbf{q}| + \mathbf{V}(\mathbf{x}, t) \cdot \mathbf{q} \geq 0\,. \tag{25}$$

ϕ is said to be a viscosity solution *of (22) if it is both a viscosity subsolution and a viscosity supersolution.*

Theorem 1 *(Frankowska, 1989) A locally Lipschitz function $\phi : \Omega \times (0, \infty) \to \mathbb{R}$ is a viscosity solution to (22) if and only if for every $(\mathbf{x}, t) \in \Omega \times (0, \infty)$,*

$$\max_{(\mathbf{q},p) \in \partial \phi(\mathbf{x},t)} \{p + F|\mathbf{q}| + \mathbf{V}(\mathbf{x}, t) \cdot \mathbf{q}\} = 0 \tag{26}$$

and for all $(\mathbf{q}, p) \in \partial_- \phi(\mathbf{x}, t)$,

$$p + F|\mathbf{q}| + \mathbf{V}(\mathbf{x}, t) \cdot \mathbf{q} = 0\,. \tag{27}$$

Theorem 2 *(Clarke et al, 1998)* ***(Lebourg's Mean Value Theorem.)*** *Let $\mathbb{S} \subseteq \mathbb{R}$ be an open set. Let $x, y \in \mathbb{S}$ and suppose that $f : \mathbb{S} \to \mathbb{R}$ is Lipschitz on an open set containing the segment $[x, y]$. Then there exists $0 < \lambda < 1$ such that*

$$f(y) - f(x) = g \times (y - x)\,, \tag{28}$$

for some $g \in \partial f(z)$, where $z = \lambda x + (1 - \lambda)y$.

Theorem 3 *(Clarke et al, 1998)* ***(Chain Rule.)*** *Let $\Omega_1 \subseteq \mathbb{R}^n$ and $\Omega_2 \subseteq \mathbb{R}^m$ be two open sets with $m, n \in \mathbb{N}$. Let $\mathbf{g} : \Omega_1 \to \Omega_2$ be continuously differentiable near $\mathbf{x} \in \Omega_1$, and let $F : \Omega_2 \to \mathbb{R}$ be Lipschitz near $\mathbf{g}(\mathbf{x})$. Then $f := F \circ \mathbf{g}$ is Lipschitz near $\mathbf{x}$ and*

$$\partial f(\mathbf{x}) \subseteq (\mathbf{g}'(\mathbf{x}))^* \, \partial F\left(\mathbf{g}(\mathbf{x})\right)\,, \tag{29}$$

where * denotes the adjoint.

B Theoretical Results

We now state a lemma that provides a monotonicity result related to ϕ, the viscosity solution of the Hamilton-Jacobi equation (22). According to this result, the generalized gradient of ϕ is non-positive on trajectories $\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t)$, along the direction $\left(\frac{\mathrm{d}\widetilde{\mathbf{X}}_P(\mathbf{y_s},t)}{\mathrm{d}t}, 1\right)$ for $t > 0$. This lemma is then used to prove Theorem 4, which establishes the relationship between reachable sets and the viscosity solution of a modified level set equation.

Lemma 1 *Let $\Omega \subseteq \mathbb{R}^n$ be open, $F > 0$ and let $\mathbf{V}(\mathbf{x}, t)$ satisfy assumptions (3)–(4). Let ϕ be the viscosity solution to (22). Let the trajectory $\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t)$ satisfy (1) with initial conditions $\widetilde{\mathbf{X}}_P(\mathbf{y_s}, 0) = \mathbf{y_s}$. Then,*

1.

$$p + \frac{d\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t)}{dt} \cdot \mathbf{q} \leq 0 \quad \forall (\mathbf{q}, p) \in \partial \phi(\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t), t) \tag{30}$$

2.

$$\phi^g \left(\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t), t; \left(\frac{d\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t)}{dt}, 1 \right) \right) \leq 0 \quad \forall\, t > 0\,. \tag{31}$$

The proof of this Lemma may be found in (Lolla et al, 2014c).

Theorem 4 *Let $\Omega \subseteq \mathbb{R}^n$ be an open set, $\mathbf{V}(\mathbf{x}, t) : \Omega \times [0, \infty) \to \mathbb{R}^n$ satisfy (3)–(4), and $F \geq 0$. Let $T^o(\mathbf{y}) : \Omega \to \mathbb{R}$ denote the optimal first arrival time at $\mathbf{y}$. Let the trajectory $\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t)$ satisfy (1) with initial conditions $\widetilde{\mathbf{X}}_P(\mathbf{y_s}, 0) = \mathbf{y_s}$. Let $\phi^o(\mathbf{x}, t)$ be the viscosity solution to the Hamilton-Jacobi equation (9) with initial condition (10). Then,*

1. *$\phi^o(\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t), t) \leq 0$ for all $t \geq 0$.*
2. *If ϕ^o is regular at $(\mathbf{X}^o_P(\mathbf{y_s}, t), t)$ for all $t > 0$ and $\mathbf{X}^o_P$ satisfies*

$$\frac{d\mathbf{X}^o_P}{dt} = F\frac{\mathbf{q}^o}{|\mathbf{q}^o|} + \mathbf{V}(\mathbf{X}^o_P(\mathbf{y_s}, t), t), \quad t > 0, \tag{32}$$

for some $(\mathbf{q}^o, p^o) \in \partial \phi^o(\mathbf{X}^o_P(\mathbf{y_s}, t), t)$, then

$$\phi^o(\mathbf{X}^o_P(\mathbf{y_s}, t), t) = 0 \quad \forall\, t \geq 0\,.$$

3.

$$T^o(\mathbf{y}) = \inf_{t \geq 0} \{t : \phi^o(\mathbf{y}, t) = 0\}\,, \tag{33}$$

where inf *denotes the infimum.*

4. *The optimal trajectory to $\mathbf{y_f} \in \Omega$ satisfies (11) whenever ϕ^o is differentiable at $(\mathbf{X}^\star_P(\mathbf{y_s}, t), t)$ and $|\nabla \phi^o(\mathbf{X}^\star_P, t)| \neq 0$.*
5. *If $F > \max_{\mathbf{x} \in \Omega, t \geq 0} |\mathbf{V}(\mathbf{x}, t)|$, then $T^o(\mathbf{y})$ is the viscosity solution of the modified Eikonal equation*

$$F|\nabla T^o(\mathbf{y})| + \mathbf{V}(\mathbf{y}, T^o(\mathbf{y})) \cdot \nabla T^o(\mathbf{y}) - 1 = 0\,, \mathbf{y} \in \Omega\,. \tag{34}$$

Proof (1). The viscosity solution to (9) is locally Lipschitz (see (Tonon, 2011; Bianchini and Tonon, 2012; Cannarsa and Sinestrari, 2004)). We now argue that $\phi^o_P(t) := \phi^o(\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t), t)$ is locally Lipschitz for all $t \geq 0$. Observe that $\phi^o_P(t) = \phi^o(\mathbf{g}_P(t))$ where $\mathbf{g}_P(t) := (\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t), t)$. Since $\mathbf{g}_P(t)$ is continuously differentiable in $(0, \infty)$ with $\frac{\mathrm{d}\mathbf{g}_P(t)}{\mathrm{d}t} = \left(\frac{\mathrm{d}\widetilde{\mathbf{X}}_P}{\mathrm{d}t}, 1\right)$ and ϕ^o is locally Lipschitz, $\phi^o_P(t)$ is also locally Lipschitz in $(0, \infty)$ by the chain rule stated in Theorem 3.

Let $t_1 > 0$ be fixed. Since ϕ^o_P is locally Lipschitz, there exists an open interval around t_1 in which ϕ^o_P is Lipschitz. Thus, for any $t_2 > t_1$ in this interval, Lebourg's Mean Value Theorem (Theorem 2) implies there exist $t_3 \in (t_1, t_2)$ and $s \in \partial \phi^o_P(t_3)$ such that

$$\phi^o_P(t_2) - \phi^o_P(t_1) = s \times (t_2 - t_1)\,. \tag{35}$$

Using the chain rule of Theorem 3 again (* denotes the adjoint),

$$\begin{aligned}\partial \phi^o_P(t_3) &\subseteq (\mathbf{g}'_P(t_3))^* \, \partial \phi^o\left(\mathbf{g}_P(t_3)\right) \\ &= \Bigg\{ p + \mathbf{q} \cdot \frac{\mathrm{d}\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t_3)}{\mathrm{d}t} \\ &\qquad : (\mathbf{q}, p) \in \partial \phi^o(\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t_3), t_3) \Bigg\}\,.\end{aligned} \tag{36}$$

Hence, any $s \in \partial \phi^o_P(t_3)$ can be written as

$$s = p + \mathbf{q} \cdot \frac{\mathrm{d}\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t_3)}{\mathrm{d}t} \,. \tag{37}$$

for some $(\mathbf{q}, p) \in \partial \phi^o(\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t_3), t_3)$. From (30),

$$p + \mathbf{q} \cdot \frac{\mathrm{d}\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t_3)}{\mathrm{d}t} \leq 0 \,,$$

implying that for any $s \in \partial \phi^o_P(t_3)$, $s \leq 0$. Using this result in (35) yields $\phi^o_P(t_2) \leq \phi^o_P(t_1)$ for all t_1, t_2. Since ϕ^o_P is locally Lipschitz in $(0, \infty)$, we conclude that $\phi^o_P(t)$ is non-increasing on $(0, \infty)$. Moreover, since ϕ^o_P is continuous on $[0, \infty)$, with $\phi^o_P(0) = 0$ (from (10)) and non-increasing in $(0, \infty)$, we have $\phi^o_P(t) = \phi(\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t), t) \leq 0$ for all $t \geq 0$.

(2). When the trajectory $\mathbf{X}^o_P(\mathbf{y_s}, t)$ is regular, i.e. when ϕ^o is regular at points $(\mathbf{X}^o_P(\mathbf{y_s}, t), t)$ for all $t > 0$, (21) implies $\partial_- \phi^o(\mathbf{X}^o_P(\mathbf{y_s}, t), t) = \partial \phi^o(\mathbf{X}^o_P(\mathbf{y_s}, t), t)$ for all $t > 0$. Since ϕ^o is the viscosity solution to (9), we obtain from Theorem 1 that for any $t > 0$, $p + F|\mathbf{q}| + \mathbf{V}(\mathbf{X}^o_P(\mathbf{y_s}, t), t) \cdot \mathbf{q} = 0$ for all $(\mathbf{q}, p) \in \partial_- \phi^o(\mathbf{X}^o_P(\mathbf{y_s}, t), t) = \partial \phi^o(\mathbf{X}^o_P(\mathbf{y_s}, t), t)$. Specifically, for the member $(\mathbf{q}^o, p^o)$ of $\partial \phi^o(\mathbf{X}^o_P(\mathbf{y_s}, t), t)$ that satisfies (32), the definition of generalized gradient (20) implies

$$\begin{aligned}
&\phi^{o\,g}\left(\mathbf{X}^o_P(\mathbf{y_s}, t), t; \left(\frac{\mathrm{d}\mathbf{X}^o_P(\mathbf{y_s}, t)}{\mathrm{d}t}, 1\right)\right) \\
&\quad \geq p^o + \mathbf{q}^o \cdot \frac{\mathrm{d}\mathbf{X}^o_P(\mathbf{y_s}, t)}{\mathrm{d}t} \\
&\quad = p^o + F|\mathbf{q}^o| + \mathbf{V}(\mathbf{X}^o_P(\mathbf{y_s}, t), t) \cdot \mathbf{q}^o \\
&\quad = 0 \,.
\end{aligned} \tag{38}$$

Combining this result with (31), we obtain

$$\phi^{o\,g}\left(\mathbf{X}^o_P(\mathbf{y_s}, t), t; \left(\frac{\mathrm{d}\mathbf{X}^o_P(\mathbf{y_s}, t)}{\mathrm{d}t}, 1\right)\right) = 0 \,.$$

Since ϕ^o is regular at $(\mathbf{X}^o_P(\mathbf{y_s}, t), t)$ by assumption, we then also have

$$\phi^{o\,d}\left(\mathbf{X}^o_P(\mathbf{y_s}, t), t; \left(\frac{\mathrm{d}\mathbf{X}^o_P(\mathbf{y_s}, t)}{\mathrm{d}t}, 1\right)\right) = 0 \,. \tag{39}$$

For any $h > 0$, the definition of ϕ^o_P implies

$$\begin{aligned}
&\left| \frac{\phi^o_P(t+h) - \phi^o_P(t)}{h} \right| \\
&= \left| \frac{\phi^o\left(\mathbf{X}^o_P(\mathbf{y_s}, t+h), t+h\right) - \phi^o\left(\mathbf{X}^o_P(\mathbf{y_s}, t), t\right)}{h} \right| \,.
\end{aligned} \tag{40}$$

Since ϕ^o is locally Lipschitz, $\exists C > 0$ such that for $h > 0$ small enough,

$$\begin{aligned}
&\left| \phi\left(\mathbf{X}^o_P(\mathbf{y_s}, t+h), t+h\right) - \phi^o\left(\mathbf{X}^o_P(\mathbf{y_s}, t) + h \frac{\mathrm{d}\mathbf{X}^o_P}{\mathrm{d}t}, t+h\right) \right| \\
&\leq C \left| \mathbf{X}^o_P(\mathbf{y_s}, t+h) - \mathbf{X}^o_P(\mathbf{y_s}, t) - h \frac{\mathrm{d}\mathbf{X}^o_P}{\mathrm{d}t} \right| \\
&= C\, |\mathbf{o}(h)| \,,
\end{aligned} \tag{41}$$

where $\mathbf{o}(h) \in \mathbb{R}^n$ denotes a vector whose individual terms are $o(h)$. Adding and subtracting $\phi^o\left(\mathbf{X}^o_P + h\frac{\mathrm{d}\mathbf{X}^o_P}{\mathrm{d}t}, t+h\right)$ from the numerator of (40) and using the triangle inequality, we obtain

$$\begin{aligned}
&\left| \frac{\phi^o_P(t+h) - \phi^o_P(t)}{h} \right| \\
&\leq \left| \frac{\phi^o\left(\mathbf{X}^o_P(\mathbf{y_s}, t) + h\frac{\mathrm{d}\mathbf{X}^o_P(\mathbf{y_s}, t)}{\mathrm{d}t}, t+h\right) - \phi^o\left(\mathbf{X}^o_P(\mathbf{y_s}, t), t\right)}{h} \right| \\
&\qquad + C \left| \frac{\mathbf{o}(h)}{h} \right| .
\end{aligned}$$

The first term on the right converges to: $\phi^{o\,d}\left(\mathbf{X}^o_P, t; \left(\frac{\mathrm{d}\mathbf{X}^o_P}{\mathrm{d}t}, 1\right)\right)$ as $h \downarrow 0$ and by (39), its value is zero. The second term uniformly converges to zero as $h \downarrow 0$, by definition. This implies

$$\lim_{h \downarrow 0} \left| \frac{\phi^o_P(t+h) - \phi^o_P(t)}{h} \right| = 0 \,,$$

and consequently that

$$\lim_{h \downarrow 0} \frac{\phi^o_P(t+h) - \phi^o_P(t)}{h} = 0 \,. \tag{42}$$

Since (42) holds for all $t > 0$, ϕ^o_P is right differentiable in $(0, \infty)$ and the value of the right-derivative is zero for all $t > 0$. This implies that ϕ^o_P is constant in $(0, \infty)$. Since $\phi^o_P(0) = 0$, we obtain $\phi^o_P(t) = \phi^o(\mathbf{X}^o_P(\mathbf{y_s}, t), t) = 0$ for all $t \geq 0$. Therefore, trajectories $\mathbf{X}^o_P(\mathbf{y_s}, t)$ that are regular and satisfy (32) always remain on the zero-level set of ϕ^o.

(3). It has been shown in part (1) that $\phi^o(\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t), t) \leq 0$ for all $t \geq 0$ for any trajectory $\widetilde{\mathbf{X}}_P(\mathbf{y_s}, t)$ that satisfies (1) and the initial conditions $\widetilde{\mathbf{X}}_P(\mathbf{y_s}, 0) = \mathbf{y_s}$. Therefore, for a trajectory $\mathbf{X}_P(\mathbf{y_s}, t)$ that reaches a given end point $\mathbf{y} \in \Omega$ at time $\widetilde{T}(\mathbf{y})$ (not necessarily optimal),

$$\phi^o(\mathbf{y}, \widetilde{T}(\mathbf{y})) = \phi^o(\widetilde{\mathbf{X}}_P(\mathbf{y_s}, \widetilde{T}(\mathbf{y})), \widetilde{T}(\mathbf{y})) \leq 0 \,. \tag{43}$$

Since this inequality holds for any arbitrary arrival time $\widetilde{T}(\mathbf{y})$, it will also hold for the optimal arrival time $T^o(\mathbf{y})$, implying

$$\phi^o(\mathbf{y}, T^o(\mathbf{y})) \leq 0 \ \text{ for all } \mathbf{y} \in \Omega \,. \tag{44}$$

For $\mathbf{y} = \mathbf{y_s}$, (33) holds trivially. For any $\mathbf{y} \neq \mathbf{y_s}$, $\phi^o(\mathbf{y}, 0) > 0$ by (10). The continuity of ϕ^o and (44) together then yield

$$T^o(\mathbf{y}) \geq \inf_{t \geq 0} \{t : \phi^o(\mathbf{y}, t) = 0\} \,. \tag{45}$$

In part (2), we showed the existence of trajectories that always remain on the zero level set of ϕ^o. Furthermore, any point on the zero level set of ϕ^o belongs to a characteristics of (9) emanating from $\mathbf{y_s}$, since $\mathbf{y_s}$ is the only point in Ω where ϕ^o is initially zero. Therefore, when the zero level set reaches $\mathbf{y}$ for the first time, it implies the existence of a trajectory $\mathbf{X}^o_P(\mathbf{y_s}, t)$ with $\mathbf{X}^o_P(\mathbf{y_s}, 0) = \mathbf{y_s}$ that satisfies (1). For this trajectory, (45) holds with an equality, thereby establishing (33). Physically, this means that fastest arrival time at any end point $\mathbf{y} \in \Omega$ is when the zero level set of ϕ^o reaches $\mathbf{y}$ for the first time, and equivalently that the reachability front $\partial\mathcal{R}(\mathbf{y_s}, t)$ coincides with the zero level set of ϕ^o at time t.

(4). Let $\mathbf{y_f} \in \Omega$ be fixed. From part (3), the optimal trajectory to $\mathbf{y_f}$ satisfies $\phi^o(\mathbf{X}^\star_P(\mathbf{y_s}, t), t) = 0$ for all $t \geq 0$. Hence, $\phi^o_P(t) := \phi^o(\mathbf{X}^\star_P(\mathbf{y_s}, t), t)$ equals zero for all $0 \leq t \leq T^\star(\mathbf{y_f})$. Let us fix a time $0 < t < T^\star(\mathbf{y_f})$ such that ϕ^o is differentiable at $(\mathbf{X}^\star_P(\mathbf{y_s}, t), t)$. The usual chain rule then yields

$$0 = \frac{\mathrm{d}\phi^o_P(t)}{\mathrm{d}t} = \frac{\partial \phi^o}{\partial t} + \nabla \phi^o \cdot \frac{\mathrm{d}\mathbf{X}^\star_P(\mathbf{y_s}, t)}{\mathrm{d}t} \,, \tag{46}$$

where the derivatives of ϕ^o are evaluated at $(\mathbf{X}^\star_P(\mathbf{y_s},t),t)$. Since ϕ^o is assumed to be differentiable at this point, (9) holds in the classical sense and $\frac{\partial \phi^o}{\partial t} = -F|\nabla\phi^o(\mathbf{X}^\star_P,t)| - \mathbf{V}(\mathbf{X}^\star_P,t)\cdot\nabla\phi^o(\mathbf{X}^\star_P,t)$. Substituting this in (46) gives

$$\frac{\mathrm{d}\mathbf{X}^\star_P}{\mathrm{d}t}\cdot\nabla\phi^o(\mathbf{X}^\star_P,t) = F|\nabla\phi^o(\mathbf{X}^\star_P,t)| + \mathbf{V}(\mathbf{X}^\star_P,t)\cdot\nabla\phi^o(\mathbf{X}^\star_P,t)\,. \tag{47}$$

Using (1),

$$\begin{aligned}&\frac{\mathrm{d}\mathbf{X}^\star_P}{\mathrm{d}t}\cdot\nabla\phi^o(\mathbf{X}^\star_P,t)\\ &= F^\star_P(t)\,\hat{\mathbf{h}}^\star(t)\cdot\nabla\phi^o(\mathbf{X}^\star_P,t) + \mathbf{V}(\mathbf{X}^\star_P,t)\cdot\nabla\phi^o(\mathbf{X}^\star_P,t)\\ &\le F|\nabla\phi^o(\mathbf{X}^\star_P,t)| + \mathbf{V}(\mathbf{X}^\star_P,t)\cdot\nabla\phi^o(\mathbf{X}^\star_P,t)\,,\end{aligned}$$

equality holding iff $F^\star_P(t) = F$ and $\hat{\mathbf{h}}^\star(t) = \frac{\nabla\phi^o(\mathbf{X}^\star_P,t)}{|\nabla\phi^o(\mathbf{X}^\star_P,t)|}$, for $|\nabla\phi^o(\mathbf{X}^\star_P,t)| \neq 0$. Using this result in (47) yields

$$\frac{\mathrm{d}\mathbf{X}^\star_P}{\mathrm{d}t} = F\,\frac{\nabla\phi^o(\mathbf{X}^\star_P,t)}{|\nabla\phi^o(\mathbf{X}^\star_P,t)|} + \mathbf{V}(\mathbf{X}^\star_P,t)\,.$$

(5). Under the assumption $F > \sup_{\mathbf{x}\in\Omega,t\ge 0}\{|\mathbf{V}(\mathbf{x},t)|\}$, the start point $\mathbf{y_s}$ belongs to the interior of the reachable set $\mathcal{R}(\mathbf{y_s},t)$ for all $t>0$, i.e. for any $t>0$, there exists $\epsilon_t > 0$ such that all points $\mathbf{x}'$ satisfying $|\mathbf{y_s}-\mathbf{x}'| < \epsilon_t$ are members of $\mathcal{R}(\mathbf{y_s},t)$. This condition is equivalent to the 'Small Time Local Controllability' condition discussed in (Bardi and Capuzzo-Dolcetta, 2008) as a result of which, T^o is continuous in Ω. See (Bardi and Capuzzo-Dolcetta, 2008) for a formal proof of this statement. Let us fix $\mathbf{y}\in\Omega$. By definition, $T^o(\mathbf{y})$ satisfies

$$T^o(\mathbf{y}) = \inf_{h>0}\{T^o(\widetilde{\mathbf{y}}) + h\}\,, \tag{48}$$

where $\widetilde{\mathbf{y}}\in\Omega$ is a point such that there exists a trajectory $\widetilde{\mathbf{X}}_P(\mathbf{y_s},t)$ satisfying (1) and the limiting conditions

$$\widetilde{\mathbf{X}}_P(\mathbf{y_s},T^o(\widetilde{\mathbf{y}})) = \widetilde{\mathbf{y}},\quad \widetilde{\mathbf{X}}_P(\mathbf{y_s},T^o(\widetilde{\mathbf{y}})+h) = \mathbf{y}\,. \tag{49}$$

In order to show that T^o is a viscosity solution to (34), we show that it is both a viscosity subsolution and a supersolution to (34).

Viscosity Subsolution: From Definition 3 and Remark 1, $T^o\in\mathcal{C}(\Omega)$ is a viscosity subsolution to (34) if at every $\mathbf{y}\in\Omega$ and for every $\mathcal{C}^1$ function $\tau_s:\Omega\to\mathbb{R}$ such that $\tau_s(\mathbf{y}) = T^o(\mathbf{y})$ and $\tau_s - T^o$ has a local minima at $\mathbf{y}$,

$$F\,|\nabla\tau_s(\mathbf{y})| + \mathbf{V}(\mathbf{y},T^o(\mathbf{y}))\cdot\nabla\tau_s(\mathbf{y}) - 1 \le 0\,. \tag{50}$$

Since $\tau_s \ge T^o$ in a neighborhood of $\mathbf{y}$, we obtain for $h>0$ small enough,

$$\tau_s(\mathbf{y}) - \tau_s(\widetilde{\mathbf{y}}) \le T^o(\mathbf{y}) - T^o(\widetilde{\mathbf{y}})\,.$$

Moreover, for this choice of h and the resulting $\widetilde{\mathbf{y}}$, (48) implies

$$T^o(\mathbf{y}) \le T^o(\widetilde{\mathbf{y}}) + h\,.$$

Combining the above two inequalities yields

$$\tau_s(\mathbf{y}) - \tau_s(\widetilde{\mathbf{y}}) \le T^o(\mathbf{y}) - T^o(\widetilde{\mathbf{y}}) \le h\,. \tag{51}$$

Since τ_s is differentiable at $\mathbf{y}$, Taylor's theorem may be used to expand $\tau_s(\widetilde{\mathbf{y}})$ near $\mathbf{y}$.

$$\begin{aligned}\tau_s(\widetilde{\mathbf{y}}) &= \tau_s(\mathbf{y}) + \nabla\tau_s(\mathbf{y})\cdot(\widetilde{\mathbf{y}}-\mathbf{y}) + \mathbf{o}\left(|\widetilde{\mathbf{y}}-\mathbf{y}|\right)\\ &= \tau_s(\mathbf{y}) - \int_{T^o(\widetilde{\mathbf{y}})}^{T^o(\widetilde{\mathbf{y}})+h}\nabla\tau_s(\mathbf{y})\cdot\frac{\mathrm{d}\widetilde{\mathbf{X}}_P}{\mathrm{d}t}\,\mathrm{d}t + \mathbf{o}\left(|\widetilde{\mathbf{y}}-\mathbf{y}|\right)\,.\end{aligned} \tag{52}$$

Inserting (52) in (51) and dividing by h,

$$\frac{1}{h}\int_{T^o(\widetilde{\mathbf{y}})}^{T^o(\widetilde{\mathbf{y}})+h}\nabla\tau_s(\mathbf{y})\cdot\frac{\mathrm{d}\widetilde{\mathbf{X}}_P}{\mathrm{d}t}\,\mathrm{d}t + \frac{\mathbf{o}\left(|\widetilde{\mathbf{y}}-\mathbf{y}|\right)}{h} \le 1\,.$$

As $h\downarrow 0$, and after noting that the second term on the left vanishes under this limit, we obtain

$$\nabla\tau_s(\mathbf{y})\cdot\frac{\mathrm{d}\widetilde{\mathbf{X}}_P}{\mathrm{d}t}(\mathbf{y_s},T^o(\mathbf{y})) \le 1\,. \tag{53}$$

One can see that (50) is satisfied trivially when $|\nabla\tau_s(\mathbf{y})| = 0$. Thus, we may assume $|\nabla\tau_s(\mathbf{y})| \neq 0$. Since (53) holds for any valid choice of $\frac{\mathrm{d}\widetilde{\mathbf{X}}_P}{\mathrm{d}t}$, we may choose $\frac{\mathrm{d}\widetilde{\mathbf{X}}_P}{\mathrm{d}t}(\mathbf{y_s},T^o(\mathbf{y})) = F\frac{\nabla\tau_s(\mathbf{y})}{|\nabla\tau_s(\mathbf{y})|} + \mathbf{V}(\mathbf{y},T^o(\mathbf{y}))$ to obtain

$$\begin{aligned}&\nabla\tau_s(\mathbf{y})\cdot\left(F\frac{\nabla\tau_s(\mathbf{y})}{|\nabla\tau_s(\mathbf{y})|} + \mathbf{V}(\mathbf{y},T^o(\mathbf{y}))\right)\\ &= F|\nabla\tau_s(\mathbf{y})| + \mathbf{V}(\mathbf{y},T^o(\mathbf{y}))\cdot\nabla\tau_s(\mathbf{y}) \le 1\,,\end{aligned}$$

thereby establishing (50). Therefore, T^o is a viscosity subsolution to (34).

Viscosity Supersolution: T^o is a viscosity supersolution to (34) if at any $\mathbf{y}\in\Omega$ and for every $\mathcal{C}^1$ function $\tau^s:\Omega\to\mathbb{R}$ such that $\tau^s(\mathbf{y}) = T^o(\mathbf{y})$ and $\tau^s - T^o$ has a local maxima at $\mathbf{y}$,

$$F|\nabla\tau^s(\mathbf{y})| + \mathbf{V}(\mathbf{y},T^o(\mathbf{y}))\cdot\nabla\tau^s(\mathbf{y}) - 1 \ge 0\,. \tag{54}$$

For any $0 < h < T^o(\mathbf{y})$, there exists $\widehat{\mathbf{y}}\in\Omega$ satisfying $T^o(\widehat{\mathbf{y}}) + h = T^o(\mathbf{y})$ and a trajectory $\widehat{\mathbf{X}}_P(\mathbf{y_s},t)$ satisfying (1) and the limiting conditions

$$\widehat{\mathbf{X}}_P(\mathbf{y_s},T^o(\widehat{\mathbf{y}})) = \widehat{\mathbf{y}},\quad \widehat{\mathbf{X}}_P(\mathbf{y_s},T^o(\mathbf{y})) = \mathbf{y}\,. \tag{55}$$

Of course, the optimal trajectory leading to $\mathbf{y}$ is a valid choice for $\widehat{\mathbf{X}}_P(\mathbf{y_s},t)$. For $h>0$ small enough,

$$h = T^o(\mathbf{y}) - T^o(\widehat{\mathbf{y}}) \le \tau^s(\mathbf{y}) - \tau^s(\widehat{\mathbf{y}})\,. \tag{56}$$

As in the earlier sub-section, we may use Taylor's theorem to expand $\tau^s(\widehat{\mathbf{y}})$ near $\mathbf{y}$ to obtain

$$\begin{aligned}\tau^s(\widehat{\mathbf{y}}) &= \tau^s(\mathbf{y}) + \nabla\tau^s(\mathbf{y})\cdot(\widehat{\mathbf{y}}-\mathbf{y}) + \mathbf{o}\left(|\widehat{\mathbf{y}}-\mathbf{y}|\right)\\ &= \tau^s(\mathbf{y}) - \int_{T^o(\widehat{\mathbf{y}})}^{T^o(\widehat{\mathbf{y}})+h}\nabla\tau^s(\mathbf{y})\cdot\frac{\mathrm{d}\widehat{\mathbf{X}}_P}{\mathrm{d}t}\,\mathrm{d}t + \mathbf{o}\left(|\widehat{\mathbf{y}}-\mathbf{y}|\right)\,.\end{aligned} \tag{57}$$

Inserting (57) in (56) and dividing by h,

$$\frac{1}{h}\int_{T^o(\widehat{\mathbf{y}})}^{T^o(\widehat{\mathbf{y}})+h}\nabla\tau^s(\mathbf{y})\cdot\frac{\mathrm{d}\widehat{\mathbf{X}}_P}{\mathrm{d}t}\,\mathrm{d}t + \frac{\mathbf{o}\left(|\widehat{\mathbf{y}}-\mathbf{y}|\right)}{h} \ge 1\,. \tag{58}$$

Observe that from (1),

$$\nabla\tau^s(\mathbf{y})\cdot\frac{\mathrm{d}\widehat{\mathbf{X}}_P(\mathbf{y_s},t)}{\mathrm{d}t} \le F|\nabla\tau^s(\mathbf{y})| + \mathbf{V}(\widehat{\mathbf{X}}_P(\mathbf{y_s},t),t)\cdot\nabla\tau^s(\mathbf{y})\,.$$

Taking limits of (58) as $h\downarrow 0$ gives

$$1 \le \nabla\tau^s(\mathbf{y})\cdot\frac{\mathrm{d}\widehat{\mathbf{X}}_P}{\mathrm{d}t}(\mathbf{y_s},T^o(\mathbf{y})) \le F|\nabla\tau^s(\mathbf{y})| + \mathbf{V}(\mathbf{y},T^o(\mathbf{y}))\cdot\nabla\tau^s(\mathbf{y})\,,$$

which proves that T^o is a viscosity supersolution of (34). Therefore, T^o is a viscosity solution to (34).

C Numerical Schemes

We now summarize the numerical schemes utilized to discretize and solve (9) and (12). These equations are solved using a Finite Volume framework implemented in MATLAB. The term $|\nabla\phi^o|$ in (9) is discretized using either a first order (Sethian, 1999b; Lolla, 2012) or a higher order (Yigit, 2011) upwind scheme and $\mathbf{V}(\mathbf{x},t)\cdot\nabla\phi^o$ is discretized using a second order TVD scheme on a staggered C-grid (Ueckermann and Lermusiaux, 2011).

C.1 Forward Level Set Evolution

We discretize (9) in time using a fractional step method as follows:

$$\frac{\bar{\phi}-\phi^o(\mathbf{x},t)}{\Delta t/2} = -F|\nabla\phi^o(\mathbf{x},t)| \tag{59}$$

$$\frac{\bar{\bar{\phi}}-\bar{\phi}}{\Delta t} = -\mathbf{V}\left(\mathbf{x},t+\frac{\Delta t}{2}\right)\cdot\nabla\bar{\phi} \tag{60}$$

$$\frac{\phi^o(\mathbf{x},t+\Delta t)-\bar{\bar{\phi}}}{\Delta t/2} = -F|\nabla\bar{\bar{\phi}}| \tag{61}$$

(59)–(61) are solved only in the interior nodes of the discretized system. For boundaries that are open inlets/outlets or side walls (i.e. not interior obstacles nor forbidden regions), open boundary conditions are used on ϕ^o, and on the intermediate variables $\bar{\phi}$ and $\bar{\bar{\phi}}$ at each time step. Specifically, a radiation boundary condition with infinite wave speed is assumed, which amounts to an internal zero normal gradient (Neumann) condition: so the boundary values are updated by replacing them with the value of the variable one cell interior to the boundary. Obstacles and forbidden regions in the domain are masked, i.e., (9) is solved only at interior nodes not lying under these regions. For points adjacent to the mask, open boundary conditions are implemented and necessary spatial gradients are evaluated using neighboring nodes that do not lie under the mask. As a result, the value of ϕ^o under the mask is never used in the computation. We note that in some situations, more complex open boundary conditions could be used as done in regional ocean modeling (Lermusiaux, 1997; Haley Jr. and Lermusiaux, 2010). We have implemented the narrow-band scheme of (Adalsteinsson and Sethian, 1995) to solve (59)–(61).

The reachability front $\partial\mathcal{R}_{\phi^o}(t)$ is extracted from the ϕ^o field at every time step using a contour algorithm. In a 2-D problem, the amount of storage required for this is not significant, because $\partial\mathcal{R}_{\phi^o}(t)$ is a 1-D curve which is numerically represented by a finite number of points. We also note that this contour extraction is not needed: we could simply store the times when the zero contour of ϕ^o crosses each grid point in order to compute the normals for the backtracking (Yigit, 2011).

C.2 Backtracking

(12) is discretized using first order (Lolla, 2012) or higher-order (Yigit, 2011) time integration schemes. Ideally, it suffices to solve (9) until the level set front first reaches $\mathbf{y_f}$. However, due to the discrete time steps, a more convenient stopping criterion is the first time, T, when $\phi^o(\mathbf{y_f},T)\le 0$. Due to this, $\mathbf{y_f}$ does not lie on the final contour $\partial\mathcal{R}_{\phi^o}(T)$ exactly. Thus, we first project $\mathbf{y_f}$ onto $\partial\mathcal{R}_{\phi^o}(T)$. The projected $\hat{\mathbf{n}}_p$ is computed as the unit normal to $\partial\mathcal{R}_{\phi^o}(T)$ at the projected point. The discretized form of (12),

$$\frac{\mathbf{X}^\star_P(\mathbf{y_s},t-\Delta t)-\mathbf{X}^\star_P(\mathbf{y_s},t)}{\Delta t} = -\mathbf{V}(\mathbf{X}^\star_P,t)-F\underbrace{\frac{\nabla\phi^o(\mathbf{X}^\star_P,t)}{|\nabla\phi^o(\mathbf{X}^\star_P,t)|}}_{\hat{\mathbf{n}}_p(\mathbf{x},t)}, \tag{62}$$

is marched back in time until we reach a point on the first saved contour and this generates a discrete representation of $\mathbf{X}^\star_P(\mathbf{y_s},t)$. Along the way, we project each newly computed trajectory point, $\mathbf{X}^\star_P(\mathbf{y_s},t-\Delta t)$ onto the corresponding intermediate level set contour (see (Lolla, 2012)). Instead of performing these projections, one can use the two intermediate discrete level set contours between which an unprojected trajectory point lies, to interpolate either the normal $\hat{\mathbf{n}}_p$ at the trajectory point, or a contour passing through the trajectory point, from which $\hat{\mathbf{n}}_p$ can be computed. This interpolation should be of sufficiently high order to prevent potential biases that may occur. One can also use a predictor-corrector scheme to compute $\mathbf{X}^\star_P(\mathbf{y_s},t-\Delta t)$ using the normals both at $t-\Delta t$ and t.

As discussed in §4.1 and the *Uniqueness* remark of §3.3, multiple optimal paths exist to end points $\mathbf{y_f}$ which lie on shock lines. However, as (9) is solved numerically, $\mathbf{y_f}$ does not lie on shock lines exactly due to discretization errors. In fact, $\mathbf{y_f}$ does not even lie exactly on the final level set contour $\partial\mathcal{R}_{\phi^o}(T)$, as mentioned above. Consequently, solving (12) in such cases yields only one of the optimal trajectories, depending on the numerical errors.

References

Adalsteinsson D, Sethian JA (1995) A fast level set method for propagating interfaces. Journal of Computational Physics 118(2):269 – 277

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

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

Athans M, Falb P (2006) Optimal Control: An Introduction to the Theory And Its Applications. Dover Books on Engineering Series, Dover Publications

Bahr A, Leonard JJ, Fallon MF (2009) Cooperative localization for autonomous underwater vehicles. The International Journal of Robotics Research 28(6):714–728

Bakolas E, Tsiotras P (2010) The Zermelo - Voronoi diagram: A dynamic partition problem. Automatica 46(12):2059 – 2067

Bardi M, Capuzzo-Dolcetta I (2008) Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations. Modern Birkhäuser Classics, Birkhäuser Boston

Barraquand J, Langlois B, Latombe JC (1992) Numerical potential field techniques for robot path planning. IEEE Transactions on Systems, Man, and Cybernetics 22(2):224–241

Bhatta P, Fiorelli E, Lekien F, Leonard NE, Paley DA, Zhang F, Bachmayer R, Davis RE, Fratantoni DM, Sepulchre R (2005) Coordination of an underwater glider fleet for adaptive sampling. In: Proceedings of International Workshop on Underwater Robotics, pp 61–69

Bianchini S, Tonon D (2012) SBV Regularity for Hamilton–Jacobi equations with Hamiltonian depending on (t, x). SIAM Journal on Mathematical Analysis 44(3):2179–2203

Binney J, Krause A, Sukhatme GS (2010) Informative path planning for an autonomous underwater vehicle. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp 4791–4796

Bokanowski O, Forcadel N, Zidani H (2010) Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. SIAM Journal of Optimal Control 48(7):4292–4316

Bressan A (2011) Viscosity solutions of Hamilton-Jacobi Equations and Optimal Control Problems. An Illustrated Tutorial, Penn State University

Bruce J, Veloso M (2002) Real-time randomized path planning for robot navigation. In: Proceedings of IROS-2002, Switzerland

Bryson AE, Ho YC (1975) Applied Optimal Control: Optimization, Estimation and Control. John Wiley and Sons

Cannarsa P, Sinestrari C (2004) Semiconcave Functions, Hamilton-Jacobi Equations, and Optimal Control. Progress in Nonlinear Differential Equations and Their Applications, Birkhäuser Boston

Canny JF (1988) The complexity of robot motion planning. MIT Press, Cambridge, MA, USA

Carroll K, McClaran S, Nelson E, Barnett D, Friesen D, William G (1992) AUV path planning: An A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones. In: Proceedings of the Symposium on Autonomous Underwater Vehicle Technology, pp 79–84

Cheung MY, Hover FS (2013) A multi-armed bandit formulation with switching costs for autonomous adaptive acoustic relay positioning. In: International Symposium on Untethered Unmanned Submersible Technology (UUST), Portsmouth, New Hampshire, USA

Cheung MY, Leighton J, Hover FS (2013) Autonomous mobile acoustic relay positioning as a multi-armed bandit with switching costs. In: International Conference on Intelligent Robotic Systems (IROS) (to appear), Tokyo, Japan

Choi HL, How JP (2010) Continuous trajectory planning of mobile sensors for informative forecasting. Automatica 46(8):1266–1275

Chopp DL (1993) Computing minimal surfaces via level set curvature flow. Journal of Computational Physics 106(1):77–91

Chopp DL (2009) Another look at velocity extensions in the level set method. SIAM Journal of Scientific Computing 31(5):3255–3273

Clarke FH, Ledyaev YS, Stern RJ, Wolenski PR (1998) Nonsmooth analysis and control theory. Springer-Verlag New York, Inc., Secaucus, NJ, USA

Cushman-Roisin B, Beckers J (2010) Introduction to Geophysical Fluid Dynamics. Physical and Numerical aspects. Academic Press

Davis RE, Leonard NE, Fratantoni DM (2009) Routing strategies for underwater gliders. Deep Sea Research Part II: Topical Studies in Oceanography 56(35):173 – 187

Dijkstra H, Katsman C (1997) Temporal variability of the wind-driven quasi-geostrophic double gyre ocean circulation: Basic bifurcation diagrams. Geophysics Astrophysics Fluid Dynamics 85:195–232

Fiorelli E, Leonard NE, Bhatta P, Paley DA, Bachmayer R, Fratantoni DM (2004) Multi-AUV control and adaptive sampling in Monterey Bay. IEEE Journal of Oceanic Engineering 34:935–948

Frankowska H (1989) Hamilton-Jacobi Equations: Viscosity Solutions and Generalized Gradients. Journal of Mathematical Analysis and Applications 141(1):21 – 26

Garau B, Bonet M, Alvarez A, Ruiz S, Pascual A (2009) 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

Garrido S, Moreno L, Blanco D (2006) Voronoi diagram and fast marching applied to path planning. In: Robotics and Automation, Proceedings 2006 IEEE International Conference on, pp 3049 –3054

Haley PJ, Lermusiaux P, Robinson A, Leslie W, Logoutov O, Cossarini G, Liang X, Moreno P, Ramp S, Doyle J, Bellingham J, Chavez F, Johnston S (2009) 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(35):127 – 148

Haley Jr PJ, Lermusiaux PFJ (2010) Multiscale two-way embedding schemes for free-surface primitive equations in the multidisciplinary simulation, estimation and assimilation system. Ocean Dynamics 60(6):1497–1537

Heaney KD, Gawarkiewicz G, Duda TF, Lermusiaux PFJ (2007) Nonlinear optimization of autonomous undersea vehicle sampling strategies for oceanographic data-assimilation. Journal of Field Robotics 24(6):437–448

Hollinger G, Choudhary S, Qarabaqi P, Murphy C, Mitra U, Sukhatme G, Stojanovic M, Singh H, Hover F (2012) Underwater data collection using robotic sensor networks. Selected Areas in Communications, IEEE Journal on 30(5):899–911

Isern-Gonzalez J, Hernandez-Sosa D, Fernasndez-Perdomo E, Cabrera-Gomez J, Dominguez-Brito AC, Prieto-Maran V (2012) Obstacle avoidance in underwater glider path planning. Journal of Physical Agents 6(1):11–20

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

Kuffner JJ, LaValle SM (2000) RRT-connect: An efficient approach to single-query path planning. In: Proceedings of IEEE International Confernce on Robotics and Automation, pp 995–1001

Latombe JC (1991) Robot Motion Planning. Kluwer Academic Publishers, Boston, MA

Lavalle SM (1998) Rapidly-exploring Random Trees: A new tool for path planning. Tech. rep., Iowa State University

LaValle SM (2006) Planning Algorithms. Cambridge University Press, Cambridge, U.K.

Leonard N, Fiorelli E (2001) Virtual leaders, artificial potentials and coordinated control of groups. In: Decision and Control, 2001. Proceedings of the 40th IEEE Conference on, vol 3, pp 2968–2973

Leonard NE, Paley D, Lekien F, Sepulchre R, Fratantoni D, Davis R (2007) Collective motion, sensor networks, and ocean sampling. Proceedings of the IEEE, special issue on the emerging technology of networked control systems 95:48–74

Lermusiaux PFJ (1997) Error Subspace Data Assimilation Methods for Ocean Field Estimation: Theory, Validation and Applications. PhD thesis, Harvard University

Lermusiaux PFJ (2006) Uncertainty estimation and prediction for interdisciplinary ocean dynamics. Journal of Computational Physics 217:176–199

Lermusiaux PFJ (2007) Adaptive modeling, adaptive data assimilation and adaptive sampling. Physica D Nonlinear Phenomena 230:172–196

Lermusiaux PFJ, Chiu CS, Gawarkiewicz GG, Abbot P, Robinson AR, Miller RN, Haley PJ, Leslie WG, Majumdar SJ, Pang A, Lekien F (2006) Quantifying uncertainties in ocean predictions. Oceanography 19(1):90–103

Lermusiaux PFJ, Lolla T, Haley Jr PJ, Yigit K, Ueckermann MP, Sondergaard T, Leslie WG (2014) Science of Autonomy: Time-Optimal Path Planning and Adaptive Sampling for Swarms of Ocean Vehicles. Chapter 11, Springer Handbook of Ocean Engineering: Autonomous Ocean Vehicles, Subsystems and Control, Tom Curtin (Ed.), in press.

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

Lolla T, Ueckermann MP, Yigit K, Haley PJ, Lermusiaux PFJ (2012) Path planning in time dependent flow fields using level set methods. In: Proceedings of IEEE International Conference on Robotics and Automation, pp 166–173

Lolla T, Haley PJ, Lermusiaux PFJ (2014a) Path planning in multiscale ocean flows: coordination, pattern formation and dynamic obstacles. Ocean Modeling (to be submitted)

Lolla T, Haley Jr PJ, Lermusiaux PFJ (2014b) Time-optimal path planning in dynamic flows using level set equations: Realistic applications. Ocean Dynamics Under review

Lolla T, Lermusiaux PFJ, Ueckermann MP, Haley PJ (2014c) Modified level set approaches for the planning of time-optimal paths for swarms of ocean vehicles, MSEAS report-14. Tech. rep., Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA USA

Melchior NA, Simmons R (2007) Particle RRT for path planning with uncertainty. In: Robotics and Automation, 2007 IEEE International Conference on, pp 1617–1624

Mitchell IM, Bayen AM, Tomlin CJ (2005) A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. IEEE Transactions on Automatic Control 50(7):947 – 957

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

Osher S, Fedkiw R (2003) Level Set Methods and Dynamic Implicit Surfaces. Springer Verlag

Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79(1):12 – 49

Paley DA, Zhang F, Leonard NE (2008) Cooperative control for ocean sampling: The glider coordinated control system. IEEE Transactions on Control Systems Technology 16(4):735–744

Pedlosky J (1998) Ocean Circulation Theory. Springer-Verlag

Pereira A, Binney J, Hollinger GA, Sukhatme GS (2013) Risk-aware path planning for autonomous underwater vehicles using predictive ocean models. Journal of Field Robotics 30(5):741–762, DOI 10.1002/rob.21472, URL http://robotics.usc.edu/publications/801/

Petres C, Pailhas Y, Patron P, Petillot Y, Evans J, Lane D (2007) Path planning for autonomous underwater vehicles. Robotics, IEEE Transactions on 23(2):331 –341

Ramp S, Davis R, Leonard N, Shulman I, Chao Y, Robinson A, Marsden J, Lermusiaux P, Fratantoni D, Paduan J, Chavez F, Bahr F, Liang S, Leslie W, Li Z (2009) 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(35):68–86

Rao D, Williams SB (2009) Large-scale path planning for underwater gliders in ocean currents. In: Proceedings of Australasian Conference on Robotics and Automation

Rhoads B, Mezic I, Poje A (2010) Minimum time feedback control of autonomous underwater vehicles. In: Decision and Control (CDC), 2010 49^{th} IEEE Conference on, pp 5828 –5834

Russo G, Smereka P (2000) A Remark on Computing Distance Functions. Journal of Computational Physics 163:51–67

Sapsis TP, Lermusiaux PFJ (2009) Dynamically orthogonal field equations for continuous stochastic dynamical systems. Physica D: Nonlinear Phenomena 238(23–24):2347–2360

Schneider T, Schmidt H (2010) Unified command and control for heterogeneous marine sensing networks. Journal of Field Robotics 27(6):876–889

Schofield O, Glenn S, Orcutt J, Arrott M, Meisinger M, Gangopadhyay A, Brown W, Signell R, Moline M, Chao Y, Chien S, Thompson D, Balasuriya A, Lermusiaux P, Oliver M (2010) Automated sensor network to advance ocean science. Eos, Transactions American Geophysical Union 91(39):345–346

Sethian JA (1999a) Fast marching methods. SIAM Rev 41(2):199–235

Sethian JA (1999b) Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press: Cambridge, U.K.

Simmonet E, Dijkstra H, Ghil M (2009) Bifurcation Analysis of Ocean, Atmosphere, and Climate Models. In 'Computational Methods for the Atmosphere and the Oceans', vol XIV, p. 187-229. Handbook of Numerical Analysis

Smith RN, Chao Y, Li PP, Caron DA, Jones BH, Sukhatme GS (2010) Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model. International Journal of Robotic Research 29(12):1475–1497

Soulignac M, Taillibert P, Rueher M (2009) Time-minimal path planning in dynamic current fields. In: Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, pp 2473 –2479

Subramani D (2014) Energy optimal path planning in time-dependent flow fields using dynamically orthogonal level-set equations. Master's thesis, Computation for Design and Optimization (CDO) Program, Massachusetts Institute of Technology

Sussman M, Smereka P, Osher S (1994) A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. Journal of Computational Physics 114:146–159

Thompson DR, Chien SA, Chao Y, Li P, Arrott M, Meisinger M, Balasuriya AP, Petillo S, Schofield O (2009) Glider Mission Planning in a Dynamic Ocean Sensorweb. In: SPARK Workshop on Scheduling and Planning Applications, International Conference on Automated Planning and Scheduling

Thompson DR, Chien SA, Chao Y, Li P, Cahill B, Levin J, Schofield O, Balasuriya AP, Petillo S, Arrott M, Meisinger M (2010) Spatiotemporal path planning in strong, dynamic, uncertain currents. In: Proceedings of IEEE International Conference on Robotics and Automation, pp

4778–4783
Tonon D (2011) Regularity Results for Hamilton-Jacobi Equations. PhD thesis, Scuola Internazionale Superiore di Studi Avanzati
Ueckermann M, Lermusiaux P, Sapsis T (2013) Numerical schemes for dynamically orthogonal equations of stochastic fluid and ocean flows. Journal of Computational Physics 233:272 – 294
Ueckermann MP, Lermusiaux PFJ (2011) 2.29 Finite Volume MATLAB Framework Documentation. Tech. rep., MSEAS Group, Massachusetts Institute of Technology, Cambridge, MA USA
Van Leer B (1977) Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. Journal of Computational Physics 23(3):276 – 299
Vasudevan C, Ganesan K (1996) Case-based path planning for autonomous underwater vehicles. Autonomous Robots 3:79–89
Wang D, Lermusiaux PFJ, Haley PJ, Eickstedt D, Leslie WG, Schmidt H (2009) Acoustically focused adaptive sampling and on-board routing for marine rapid environmental assessment. Journal of Marine Systems 78:S393–S407
Warren CW (1990) A technique for autonomous underwater vehicle route planning. In: Autonomous Underwater Vehicle Technology, Proceedings of the Symposium on, pp 201–205
Witt J, Dunbabin M (2008) Go with the flow: Optimal auv path planning in coastal environments. In: Proceedings of Australasian Conference on Robotics and Automation
Yang K, Gan S, Sukkarieh S (2010) An efficient path planning and control algorithm for ruavs in unknown and cluttered environments. Journal of Intelligent and Robotic Systems 57:101–122
Yigit K (2011) Path planning methods for autonomous underwater vehicles. Master's thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology
Yilmaz NK, Evangelinos C, Lermusiaux P, Patrikalakis NM (2008) Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. IEEE Journal of Oceanic Engineering 33(4):522–537
Zhang F, Fratantoni DM, Paley D, Lund J, Leonard NE (2007) Control of coordinated patterns for ocean sampling. International Journal of Control 80(7):1186–1199
Zhang W, Inane T, Ober-Blobaum S, Marsden JE (2008) Optimal trajectory generation for a glider in time-varying 2D ocean flows B-spline model. In: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pp 1083 –1088