Ordered Upwind Methods for Static Hamilton-Jacobi Equations

J.A. Sethian and A. Vladimirsky
Dept. of Mathematics
University of California, Berkeley 94720

Abstract

We introduce a family of fast Ordered Upwind Methods (OUMs) for approximating solutions to a wide class of static Hamilton-Jacobi equations with Dirichlet boundary conditions. Standard techniques often rely on iteration to converge to the solution of a discretized version of the partial differential equation. Our new fast methods avoid iteration through a careful use of the information about the characteristic directions of the underlying PDE. These techniques are of complexity $O(M \log M)$, where M is the total number of points in the domain. We consider anisotropic test problems in optimal control, seismology, and paths on surfaces.

1 Introduction

1.1 Equations and Discretizations

Consider the first order non-linear PDE [1]

$$H(\nabla u(\boldsymbol{x}), u(\boldsymbol{x}), \boldsymbol{x}) = 1 \qquad \boldsymbol{x} \in \Omega \subset R^2 \tag{1}$$

with the boundary condition $u(\boldsymbol{x}) = q(\boldsymbol{x})$ given on $\partial\Omega$. Smoothness in boundary data does not guarantee that a smooth solution exists. For the class of Hamilton-Jacobi equations, weak solutions can be formally introduced; a unique *viscosity solution* can be defined using conditions on smooth test functions [3, 2].

In this paper, we develop fast methods for approximating the solution to equations of this form. Start with a mesh X covering the domain Ω. Let U_i be the numerical solution value at the mesh point $\boldsymbol{x}_i \in X$. Denote the set of mesh points adjacent to $\boldsymbol{x}_i$ as $N(\boldsymbol{x}_i)$ and the set of values adjacent to U_i as $NU(\boldsymbol{x}_i) = \{U_j | \boldsymbol{x}_j \in N(\boldsymbol{x}_i)\}$. Let $\overline{H}$ be a consistent discretization of H such that one can write

$$\overline{H}(U_i, NU(\boldsymbol{x}_i), \boldsymbol{x}_i) = 1. \tag{2}$$

If M is the total number of mesh points, then one needs to solve M coupled non-linear equations simultaneously. One approach is to solve this non-linear system iteratively. For example, an iterative technique for some equations arising in control theory problems is given in [7]; an iterative technique for the Eikonal equation based on an upwind finite difference formulation is given in [14].

Our goal in this work is to introduce a set of "single-pass" algorithms. By this we mean that we recalculate each U_i at most r times, where r depends only upon the equation 1 and the mesh structure, but not upon the diameter of the mesh.

[1] For the sake of notational clarity we restrict our discussion to R^2 ; all results can be restated for R^n and for meshes on manifolds.

1.2 Characteristics and Ordering Updates

To construct single-pass algorithms with efficient update orderings, we utilize the fact that the value of $u(\boldsymbol{x})$ for the first order PDE depends only on the value of u along the characteristic(s) passing through the point $\boldsymbol{x}$. If $\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2} \in N(\boldsymbol{x}_i)$ are such that the characteristic for the mesh point $\boldsymbol{x}_i$ lies in the simplex $\boldsymbol{x}_i\boldsymbol{x}_{i_1}\boldsymbol{x}_{i_2}$ then it is useful to consider an *upwind* discretization of the PDE:

$$\overline{H}(U_i, U_{i_1}, U_{i_2}, \boldsymbol{x}_i) = 1. \tag{3}$$

This reduces the coupling in the system: U_i *depends* only upon U_{i_1} and U_{i_2} and not on all of the $NU(\boldsymbol{x}_i)$. A recursive construction allows to build the entire *dependency graph* for $\boldsymbol{x}_i$.

If two or more characteristics collide at the point $\boldsymbol{x}$, the solution loses smoothness. The entropy condition does not allow characteristics to be created at these collision points; hence, if $\boldsymbol{x}_i$ is far enough from these collision points, its dependency graph is actually a tree.

If the characteristic directions of the PDE were known in advance, then the dependency-ordering of the grid points would be known as well, leading to a fully de-coupled system. Formally, this construction would lead to a $O(M)$ method.

In general, characteristic directions are not known in advance. Nonetheless, a family of algorithms can be devised in which information about the characteristic directions is computed as the solution is constructed. We refer to this class of methods as "ordered upwind methods"; the computational complexity is $O(M \log M)$, where M is the number of mesh points, and the $\log M$ term results from the ordering process.

2 Previous Work

2.1 Dijkstra's Method

Dijkstra's method [4] is the classic single-pass algorithm for computing the minimal cost of reaching any node on a network. For simplicity, imagine a rectangular grid of size h, where the cost $C_{ij} > 0$ is given for passing through each grid point $\boldsymbol{x}_{ij} = (ih, jh)$. Given a starting point, the minimal total cost U_{ij} of arriving at the node $\boldsymbol{x}_{ij}$ can be written in terms of the minimal total cost of arriving at its neighbors:

$$U_{ij} = \min\left(U_{i-1,j}, U_{i+1,j}, U_{i,j-1}, U_{i,j+1}\right) + C_{ij}. \tag{4}$$

To find the minimal total cost, Dijkstra's method works as follows. All the mesh points are divided into three classes: *Far* (no information about the correct value of U is known), *Accepted* (the correct value of U has been computed), and *Considered* (adjacent to *Accepted*).

1. Start with all mesh points in *Far* ($U_{ij} = \infty$).
2. Move the boundary mesh points ($\boldsymbol{x}_{ij} \in \delta\Omega$) to *Accepted* ($U_{ij} = q(\boldsymbol{x}_{ij})$).

3. Move all the mesh points $\boldsymbol{x}_{ij}$ adjacent to the boundary into *Considered* and evaluate the tentative value of U_{ij} using the values at the adjacent *Accepted* mesh points according to formula 4.
4. Find the mesh point $\boldsymbol{x}_r$ with the smallest value of U among all the *Considered*.
5. Move $\boldsymbol{x}_r$ to *Accepted*.
6. Move the *Far* mesh points adjacent to $\boldsymbol{x}_r$ into *Considered*.
7. Re-evaluate the value for all the *Considered* $\boldsymbol{x}_{ij}$ adjacent to $\boldsymbol{x}_r$. If the new computed value is less than the previous tentative value for $\boldsymbol{x}_{ij}$ then update U_{ij}.
8. If *Considered* is not empty then go to 4).

The described algorithm has the computational complexity of $O(M \log(M))$; the factor of $\log(M)$ reflects the necessity of maintaining a sorted list of the *Considered* values U_i to determine the next *Accepted* mesh point. An efficient implementation of the algorithm can be built using heap-sort data structures.

Dijkstra's method can be viewed as an algorithm for computing the solution to "discrete" optimal trajectory problem on a specified network. As pointed out by Sethian in [16], the U_{ij} obtained through Dijkstra's method on the Cartesian grid is formally a first order approximation to the solution $u(x, y)$ of the differential equation

$$H(\nabla u(\boldsymbol{x}), u(\boldsymbol{x}), \boldsymbol{x}) = \max(|u_x|, |u_y|) = C(\boldsymbol{x}), \qquad (5)$$

provided the link costs are $C_{ij} = hC(\boldsymbol{x})$. This equation, however, is different from the Eikonal equation, which describes the isotropic optimal trajectory (and front propagation) problems in the continuous case.

2.2 The Eikonal Equation

Two different Dijkstra-like single-pass methods have been developed in recent years for the Eikonal equation

$$\begin{aligned} \|\nabla u(\boldsymbol{x})\| &= K(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega \\ u(\boldsymbol{x}) &= q(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega. \end{aligned} \qquad (6)$$

Equation 6 is a simple example of static Hamilton-Jacobi PDEs with convex Hamiltonian. Eikonal equations are found in a variety of application domains including robotic navigation, computational geometry, photolithography, computer vision, and seismology (see [16]). These can be interpreted as

- Isotropic front propagation problems.
- Isotropic min-time optimal trajectory problems.

The connections between these two perspectives are explored in detail in [6]. In the control-theoretic context, the characteristic lines of the equation 6 can also be interpreted as the optimal trajectories.

A key feature of Eikonal equations is that their characteristic lines coincide with the gradient lines of the viscosity solution $u(\boldsymbol{x})$; this allows the construction of single-pass algorithms. This property guarantees that if $\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2} \in N(\boldsymbol{x}_i)$ are such that the characteristic for the mesh point $\boldsymbol{x}_i$ lies in the acute simplex $\boldsymbol{x}_i\boldsymbol{x}_{i_1}\boldsymbol{x}_{i_2}$ and if that simplex is sufficiently small then $u(\boldsymbol{x}_i) \geq \max(u(\boldsymbol{x}_{i_1}), u(\boldsymbol{x}_{i_2}))$ (see Figure 1, for example).

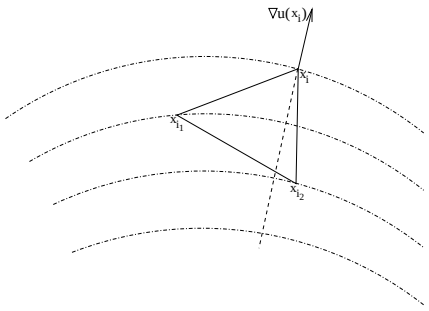

Figure 1: The characteristics and the gradient lines are the same for the Eikonal equation. If the characteristic lies in the simplex $\boldsymbol{x}_i\boldsymbol{x}_{i_1}\boldsymbol{x}_{i_2}$ then the gradient $\nabla u(\boldsymbol{x}_i)$ points from that simplex.

This observation has been used to build two different fast Dijkstra-like methods for the Eikonal equation: Tsitsiklis' Algorithm (1995) and Sethian's Fast Marching Method (1996). Using different upwinding discretizations $\overline{H}(U_i, U_{i_1}, U_{i_2}, \boldsymbol{x}_i) = 1$ of the Eikonal equation, Tsitsiklis [21] and Sethian [15] have shown that the following *causality* property holds, namely:

If $\boldsymbol{x}_i\boldsymbol{x}_{i_1}\boldsymbol{x}_{i_2}$ is an acute simplex containing the characteristic for $\boldsymbol{x}_i$ then the computed $U_i \geq \max(U_{i_1}, U_{i_2})$, regardless of the size of that simplex. (*)

2.2.1 A control-theoretic single-pass algorithm for the Eikonal equation

Tsitsiklis' algorithm evolved from studying isotropic min-time optimal trajectory problems. Tsitsiklis has shown that Property (*) holds for the first order control-theoretic upwinding discretization[2]

$$U_i = \min_{\theta \in [0,1]} \{\tau(\theta) K(\boldsymbol{x}) + \theta U_{i_1} + (1-\theta) U_{i_2}\}, \qquad (7)$$

where $\tau(\theta) = \|\theta \boldsymbol{x}_{i_1} + (1-\theta)\boldsymbol{x}_{i_2} - \boldsymbol{x}_i\|$. If θ_0 is the minimizer then the direction $(\theta_0 \boldsymbol{x}_{i_1} + (1-\theta_0)\boldsymbol{x}_{i_2} - \boldsymbol{x}_i)$ is the approximation of the characteristic direction for $\boldsymbol{x}_i$. The dynamic programming motivation and further details can be found in [21].

2.2.2 A finite-difference single-pass algorithm for the Eikonal equation

Sethian's Fast Marching Method evolved from studying isotropic front propagation problems. Sethian has shown that the Property (*) holds for the following finite-difference upwinding discretization[3]. Let $\boldsymbol{P_1} = \frac{\boldsymbol{x}_i - \boldsymbol{x}_{i_1}}{\|\boldsymbol{x}_i - \boldsymbol{x}_{i_1}\|}$ and $\boldsymbol{P_2} = \frac{\boldsymbol{x}_i - \boldsymbol{x}_{i_2}}{\|\boldsymbol{x}_i - \boldsymbol{x}_{i_2}\|}$. If P is

[2] In [21] the property is proven for the discretization on a uniform Cartesian grid only; the corresponding lemma for triangulated acute meshes is proven in [18]. An iterative approach to the corresponding system of discretized equations was earlier employed in [7] and [10].

[3] In [15] the property was proven for the first order discretization on a uniform Cartesian grid; the property for triangulated acute meshes in R^2 and R^n was proven in [9] and [19]. An iterative

the matrix with the rows $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ then the equation 6 can be rewritten as

$$\left[\begin{array}{cc} \frac{U_i - U_{i_1}}{\|\boldsymbol{x}_i - \boldsymbol{x}_{i_1}\|} & \frac{U_i - U_{i_2}}{\|\boldsymbol{x}_i - \boldsymbol{x}_{i_2}\|} \end{array} \right] \left(PP^T\right)^{-1} \left[\begin{array}{c} \frac{U_i - U_{i_1}}{\|\boldsymbol{x}_i - \boldsymbol{x}_{i_1}\|} \\ \\ \frac{U_i - U_{i_2}}{\|\boldsymbol{x}_i - \boldsymbol{x}_{i_2}\|} \end{array} \right] = K^2(\boldsymbol{x}). \tag{8}$$

This quadratic equation is solved for U_i and the additional upwinding requirement is verified (see [19] for details).

The causality property guarantees that the discretized equations can be solved one-by-one (i.e., not simultaneously) if one solves them in the order of increasing values of U; thus, each of the two presented fast Eikonal solvers can be considered as a de-coupling of a particular coupled discretization.

If the ascending ordering of the mesh points based on U were a priori known, these methods would have a computational complexity of $O(M)$. Since the ordering has to be determined in the process of computation, both methods resolve this problem in the manner similar to Dijkstra's method, and the resulting complexity is $O(M \log M)$.

Higher-order versions of the Fast Marching Method together with numerical convergence tests are presented in [16]. Extensions to triangulated manifolds are derived in [9, 19]. Some early application of the methodology include photolithography in [17], a comparison of a similar approach with volume-of-fluid techniques in [8], and a fast algorithm for image segmentation in [11].

3 Static Hamilton-Jacobi Equations

We now introduce a family of single-pass methods for a much more general class of PDEs, namely

$$\begin{aligned} H(\nabla u(\boldsymbol{x}), \boldsymbol{x}) &= 1, \quad \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) &= q(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega, \end{aligned} \tag{9}$$

where q and H are assumed to be Lipschitz-continuous and the Hamiltonian H is also assumed to be convex and homogeneous of degree one in the first argument: $H(\nabla u, \boldsymbol{x}) = \|\nabla u\| F\left(\frac{\nabla u}{\|\nabla u\|}, \boldsymbol{x}\right)$ for some function F. We will further assume that $0 < F_1 \le F(\boldsymbol{p}, \boldsymbol{x}) \le F_2 < \infty$ for all $\boldsymbol{p}$ and $\boldsymbol{x}$.

This equation can also be viewed in different frameworks:

- Anisotropic front expansion (contraction) problems, in which the speed of the front F in the normal direction depends not only upon the position, but also on the current orientation of the front, provided that the resulting Hamiltonian is convex. For example, consider the front propagation problem

$$F\left(\frac{\nabla u}{\|\nabla u\|}, \boldsymbol{x}\right) \|\nabla u\| = 1 \tag{10}$$

  where the speed $F\left(\frac{\nabla u}{\|\nabla u\|}, \boldsymbol{x}\right) = \left(\frac{\nabla u}{\|\nabla u\|} \cdot \boldsymbol{v}\right)^2 + 1$, with a given vector $\boldsymbol{v}$. This corresponds to anisotropic front motion, in which the fastest speed occurs when the normal to the front is pointing in the direction $\pm\boldsymbol{v}$.

approach to the corresponding system of discretized equations on a uniform Cartesian grid was earlier employed in [14].

- Anisotropic min-time optimal trajectory problems, in which the speed of motion depends not only on position but also on direction. The value function u for such problems is the viscosity solution of the static Hamilton-Jacobi-Bellman equation

$$\begin{aligned} &\max_{\boldsymbol{a} \in S_1} \{(\nabla u(\boldsymbol{x}) \cdot (-\boldsymbol{a})) f(\boldsymbol{a}, \boldsymbol{x})\} = 1, \quad \boldsymbol{x} \in \Omega, \\ &u(\boldsymbol{x}) = q(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega. \end{aligned} \tag{11}$$

  In this formulation, $\boldsymbol{a}$ is the unit vector determining the direction of motion, $f(\boldsymbol{a}, \boldsymbol{x})$ is the speed of motion in the direction $\boldsymbol{a}$ starting from the point $\boldsymbol{x} \in \Omega$, and $q(\boldsymbol{x})$ is the time-penalty for exiting the domain at the point $\boldsymbol{x} \in \partial\Omega$. The maximizer $\boldsymbol{a}$ corresponds to the characteristic direction for the point $\boldsymbol{x}$. If f does not depend on $\boldsymbol{a}$, Eqn. 11 reduces to the Eikonal equation, see [1]. For background discussion of control theory, see [1, 6, 7, 10] or [22].

In [22] and [18] we explore the connections between these two application domains[4].

3.1 Characteristics vs. gradients

Building a single-pass ordered upwind method for these anisotropic problems described by Eqn. 9 is considerably more challenging than it is for the Eikonal case, since the characteristics no longer coincide with the gradient lines of the viscosity solution. As a result, the characteristics and gradient lines may in fact lie in different simplexes (see Figure 2).

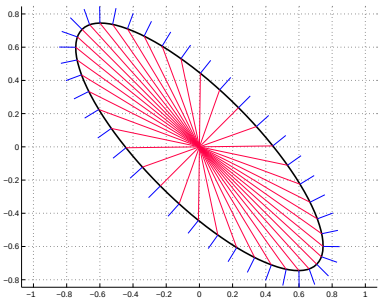

Figure 2: The characteristics and the gradient directions for the expanding ellipse.

In fact, even if the characteristic for $\boldsymbol{x}$ lies in the simplex $\boldsymbol{x}\boldsymbol{x}_1\boldsymbol{x}_2$, it is still possible that $u(\boldsymbol{x})$ is smaller than $u(\boldsymbol{x}_1)$

[4] We show that all of the PDEs of the form (9) can be produced as Hamilton-Jacobi-Bellman PDEs (11) for the min-time optimal trajectory problems. We show that the speed functions F and f are related by the homogeneous Legendre transform and that the function f generated from F will satisfy the same inequality $0 < F_1 \le f(\boldsymbol{a}, \boldsymbol{x}) \le F_2 < \infty$ for all $\boldsymbol{a}$ and $\boldsymbol{x}$. The geometric relationship between these two classes of problems is the basis for the alternative (anisotropic) Huyghens' construction using Wulff's shapes. This construction allows one to treat anisotropic front expansion/contraction problems using the methods for the Hamilton-Jacobi-Bellman PDEs, provided the Hamiltonian H is convex.

We note that aspects of these issues have been previously discussed in several specific contexts, including geometric optics [12], geophysics [13] and crystal growth [20].

or $u(\boldsymbol{x}_2)$. This is precisely why both Sethian's Fast Marching Method and Tsitsiklis' Algorithm cannot be directly applied in the anisotropic (non-Eikonal) case: it is no longer possible to de-couple the system by computing/accepting the mesh points in the ascending order. Thus, we must devise a causality relationship more carefully.

3.2 Ordered Upwind Methods

In order to build single-pass methods for this more general equation, we capitalize on the more fundamental properties of the Hamilton-Jacobi-Bellman PDEs, namely, that the viscosity solution u of the equation 9 is monotone increasing along the characteristic.

Even though the gradient lines and characteristics are not the same, our assumptions about F being bounded and bounded away from zero allow to compute a local bound on the difference between the characteristic and the gradient direction. In [22] and [18] we prove the following three lemmas about the viscosity solution u:

- **Lemma 1.** Let $\boldsymbol{a}$ be the characteristic direction for $\boldsymbol{x} \in \Omega$. If $\nabla u(\boldsymbol{x})$ is well defined and γ is the angle between the vectors $\boldsymbol{a}$ and $(-\nabla u(\boldsymbol{x}))$ then
$$\cos(\gamma) \geq \frac{F_1}{F_2}. \tag{12}$$
- **Lemma 2.** Consider the characteristic passing through a point $\bar{\boldsymbol{x}} \in \Omega$ and a level curve $u(\boldsymbol{x}) = C$, where $q_{max} < C < u(\bar{\boldsymbol{x}})$. The characteristic intersects that level set at some point $\tilde{\boldsymbol{x}}$. If $\bar{\boldsymbol{x}}$ is distance d away from the level set then
$$\|\tilde{\boldsymbol{x}} - \bar{\boldsymbol{x}}\| \leq d\frac{F_2}{F_1}. \tag{13}$$
- **Lemma 3.** Consider an unstructured mesh X of diameter h on Ω. Consider a simple closed curve Γ lying inside Ω with the property that for any point $\boldsymbol{x}$ on Γ, there exists a mesh point y inside Γ such that $\|\boldsymbol{x} - \boldsymbol{y}\| < h$. Suppose the mesh point $\bar{\boldsymbol{x}}_i$ has the smallest value $u(\bar{\boldsymbol{x}}_i)$ of all of the mesh points inside the curve. If the characteristic passing through $\bar{\boldsymbol{x}}_i$ intersects that curve at some point $\tilde{\boldsymbol{x}}_i$ then
$$\|\tilde{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_i\| \leq h\frac{F_2}{F_1}. \tag{14}$$

Building on these results we construct the following fast, ordered upwind method. As before, mesh points are divided into three classes (*Far*, *Considered*, *Accepted*). The $AcceptedFront$ is defined as a set of *Accepted* mesh points, which are adjacent to some not-yet-accepted mesh points. Define the set AF of the line segments $\boldsymbol{x}_j\boldsymbol{x}_k$, where $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ are adjacent mesh points on the $AcceptedFront$, such that there exists a *Considered* mesh point $\boldsymbol{x}_i$ adjacent to both $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$. For each *Considered* mesh point $\boldsymbol{x}_i$ we define the part of AF "relevant to $\boldsymbol{x}_i$":

$$NF(\boldsymbol{x}_i) = \left\{ (\boldsymbol{x}_j, \boldsymbol{x}_k) \in AF \mid \exists \tilde{\boldsymbol{x}} \text{ on } (\boldsymbol{x}_j, \boldsymbol{x}_k) \text{ s.t. } \|\tilde{\boldsymbol{x}} - \boldsymbol{x}_i\| \leq h\frac{F_2}{F_1} \right\}.$$

We will further assume that some consistent upwinding update formula is available: if the characteristic for $\boldsymbol{x}_i$ lies in the simplex $\boldsymbol{x}_i\boldsymbol{x}_j\boldsymbol{x}_k$ then $U_i = G(U_j, U_k, \boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$. For the sake of notational simplicity we will refer to this value as $G_{j,k}$.

1. Start with all mesh points in Far ($U_i = \infty$).
2. Move the boundary mesh points ($\boldsymbol{x}_i \in \delta\Omega$) to *Accepted* ($U_i = q(\boldsymbol{x}_i)$).
3. Move all the mesh points $\boldsymbol{x}_i$ adjacent to the boundary into *Considered* and evaluate the tentative value of $U_i = \min_{(\boldsymbol{x}_j,\boldsymbol{x}_k) NF(\boldsymbol{x}_i)} G_{j,k}$.
4. Find the mesh point $\boldsymbol{x}_r$ with the smallest value of U among all the *Considered*.
5. Move $\boldsymbol{x}_r$ to *Accepted* and update the $AcceptedFront$.
6. Move the Far mesh points adjacent to $\boldsymbol{x}_r$ into *Considered*.
7. Recompute the value for all the *Considered* $\boldsymbol{x}_i$ within the distance $h\frac{F_2}{F_1}$ from $\boldsymbol{x}_r$. If the new computed value is less than the previous tentative value for $\boldsymbol{x}_i$ then update U_i.
8. If *Considered* is not empty then goto 4).

- **Efficiency:** This results in a "single-pass" method since the maximum number of times each mesh point can be re-evaluated is bounded by the number of mesh points in the $h\frac{F_2}{F_1}$ neighborhood of that point. Thus, this method formally has the computational complexity of $O((\frac{F_2}{F_1})^2 M \log(M))$. Moreover, since the $AcceptedFront$ is approximating the level set of the viscosity solution u, as the mesh is refined, the complexity will behave as $O(\frac{F_2}{F_1} M \log(M))$.
- **Convergence:** Convergence of the method depends upon the upwinding update formula $U_i = G(U_j, U_k, \boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$. In [22] and [18] we prove the convergence to the viscosity solution of the PDE 11 using the particular update formula
$$U_i = \min_{\theta \in [0,1]} \left\{ \frac{\tau(\theta)}{f(\boldsymbol{x}, \boldsymbol{a})} + (\theta U_j + (1-\theta)U_k) \right\}, \tag{15}$$
where $\tau(\theta) = \|\theta\boldsymbol{x}_j + (1-\theta)\boldsymbol{x}_k - \boldsymbol{x}_i\|$, and $\boldsymbol{a} = \frac{\theta\boldsymbol{x}_j + (1-\theta)\boldsymbol{x}_k - \boldsymbol{x}_i}{\tau(\theta)}$. This upwinding formula is the anisotropic generalization of the update formula 7 for the Eikonal equation; it is also used as a building block for the general numerical scheme described in [7]. That scheme computes U_i based on $NU(\boldsymbol{x}_i)$, i.e., all $\boldsymbol{x}_j, \boldsymbol{x}_k$ immediately adjacent to $\boldsymbol{x}_i$ are considered. Unfortunately, that scheme also leads to the coupled system of M non-linear equations, which cannot be directly de-coupled due to the difference between the gradient and characteristic directions. Thus, our method can be considered as an indirect decoupling of the scheme described in [7].
- In [22] and [18] we also use the above method with other upwinding update formulas, obtained as the anisotropic generalizations of the discretization 8. The advantage of this approach is that it can be easily generalized for the higher-order upwinding finite difference approximations.

4 Numerical Results

In this section we consider three different test problems, each of which can be described by a non-Eikonal (anisotropic) Hamilton-Jacobi PDE. In each case, the speed function is assumed to be known from the characterization of a particular application domain. For example, in the optimal trajectory test problem, $f(\boldsymbol{x}, \boldsymbol{a})$ reflects the assumptions about the speed of the controlled vehicle, while in the seismic imaging test problem, the front expansion speed F is derived using the assumptions about the elliptical nature of the "impulse-response surface" for the anisotropic medium.

4.1 Geodesic distances on manifolds

The first test problem is to find the geodesic distance on the manifold $z = g(x, y)$. As described in [9] and [16], this can be accomplished by approximating the manifold with a triangulated mesh and then solving the distance equation $\|\nabla u\| = 1$ on that mesh. Since the latter equation is Eikonal, the Fast Marching Method can be used to solve it efficiently. However, if one desires to formulate the problem in the $x - y$ plane instead of the intrinsic manifold coordinates then the corresponding equation for u is not Eikonal. Indeed, the geodesic distance function u has to satisfy the equation 9 with the speed function F defined as:

$$F(\omega, x, y) = \sqrt{\frac{1 + g_y^2 \cos^2(\omega) + g_x^2 \sin^2(\omega) - g_x g_y \sin(2\omega)}{1 + g_x^2 + g_y^2}}, \tag{16}$$

where ω is the angle between $\nabla u(x, y)$ and the positive direction of the x-axis. The degree of anisotropy in this equation is substantial, since the dependence of F upon ω can be pronounced when ∇g is relatively large[5].

As noted in section 3, u can also be considered as a value function for the corresponding min-time optimal trajectory problem and must, therefore, satisfy the equation 11. Using the homogeneous Legendre transform on F, we obtain the speed function $f(\boldsymbol{a}, x, y)$ for the optimal trajectory problem:

$$f(\boldsymbol{a}, x, y) = \left(1 + (\nabla g(x, y) \cdot \boldsymbol{a})^2\right)^{-\frac{1}{2}}, \tag{17}$$

where $\boldsymbol{a}$ is a vector of unit length and f is the control-theoretic speed of motion in the direction $\boldsymbol{a}$ (see [22, 18] for details).

As an example, we consider the manifold $g(x, y) = .75 \sin(3\pi x) \sin(3\pi y)$ and compute the geodesic distance on it from the origin (see Figure 3). The "anisotropy coefficient" for this problem is

$$\max_{x,y} \frac{F_2(x, y)}{F_1(x, y)} = \sqrt{32 + 81\pi^2}/(4\sqrt{2}) \approx 5.1$$

We have performed the calculations using the first-order discretization 15 on a regular mesh with 292×292 mesh points.

[5] The algorithm presented in [9] on the manifold-approximating mesh is more efficient for this problem; here, it serves as a convenient test problem for the general anisotropic case: the numerical solution obtained by the Fast Marching Method on the manifold can then be compared to the solution obtained by the "general" method in the $x - y$ plane. We note, of course, that only specific anisotropic problems can be converted into Eikonal equations on manifolds; see [19] for details.

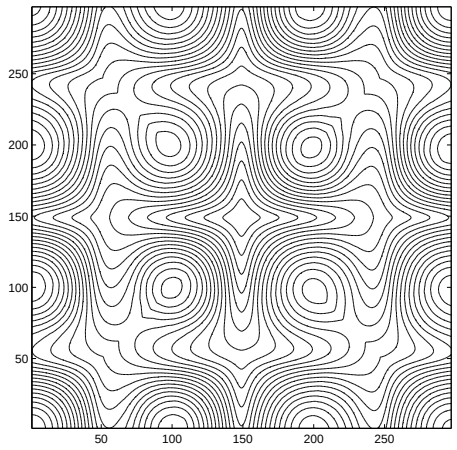

Figure 3: The geodesic distance from the origin on the manifold $z = .75 \sin(3\pi x) \sin(3\pi y)$ computed on the square $[-.5, .5] \times [-.5, .5]$ in the $x - y$ plane.

4.2 Min-time optimal trajectory

Our second example is a particular min-time optimal trajectory problem. Suppose that a vehicle's dynamics $\boldsymbol{z}(t)$ in the square $[-.5, .5] \times [-.5, .5]$ is described by

$$\begin{aligned} \frac{d\boldsymbol{z}}{dt}(t) &= \tilde{\boldsymbol{a}}(t) + \boldsymbol{b}(\boldsymbol{z}(t)), \\ \boldsymbol{z}(0) &= \begin{bmatrix} x \\ y \end{bmatrix}, \end{aligned} \tag{18}$$

where the advection velocity $\boldsymbol{b} : R^2 \to R^2$ is known and $\tilde{\boldsymbol{a}} : R_{+,0} \to S_1$ is the control. We further assume that $\|\boldsymbol{b}(x, y)\| < 1$ for all $(x, y) \in [-.5, .5] \times [-.5, .5]$. The optimal control $\tilde{\boldsymbol{a}}(\cdot)$ will minimize the time it takes for the vehicle to reach the origin from the point $\boldsymbol{x}$. The value function $u(\boldsymbol{x})$ corresponding to that minimal time is the viscosity solution of the Hamilton-Jacobi-Bellman equation 11 with the boundary condition $u(0, 0) = 0$. The speed profile for the point $\boldsymbol{x}$ is a unit circle displaced by the vector $\boldsymbol{b}(x, y)$ (see Figure 4, for example). The corresponding speed function is

$$f(\boldsymbol{a}, x, y) = \boldsymbol{a} \cdot \boldsymbol{b} + \sqrt{(\boldsymbol{a} \cdot \boldsymbol{b})^2 - \boldsymbol{b} \cdot \boldsymbol{b} + 1}, \tag{19}$$

where the direction of motion is $\boldsymbol{a} = \frac{\tilde{\boldsymbol{a}} + \boldsymbol{b}}{\|\tilde{\boldsymbol{a}} + \boldsymbol{b}\|}$.

If $\boldsymbol{b}(x, y) = \begin{bmatrix} b_1(x, y) \\ b_2(x, y) \end{bmatrix}$ then we can rewrite the Hamilton-Jacobi PDE satisfied by $u(x, y)$ as follows:

$$\begin{aligned} (1 - b_1^2)u_x^2 + (1 - b_2^2)u_y^2 - 2b_1 b_2 u_x u_y - 2b_1 u_x - 2b_2 u_y &= 1, \\ u(0, 0) &= 0. \end{aligned} \tag{20}$$

As an example, we consider the particular advection velocity $\boldsymbol{b}(x, y) = \frac{-.9 \sin(4\pi x) \sin(4\pi y)}{\sqrt{x^2 + y^2}} \begin{bmatrix} x \\ y \end{bmatrix}$ (see Figure 5). The "anisotropy coefficient" for this problem is

$$\frac{F_2(x, y)}{F_1(x, y)} = \max_{x,y} \frac{1 + \|\nabla \boldsymbol{b}(x, y)\|}{1 - \|\nabla \boldsymbol{b}(x, y)\|} = 19$$

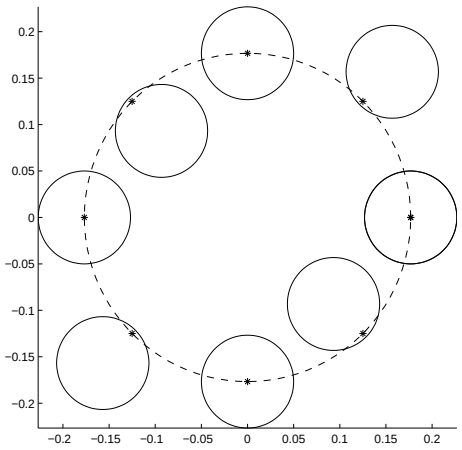

Figure 4: The vehicle's speed profiles for 8 different points on the circle.

We have performed the calculations using the first-order discretization 15 on a regular mesh with 96×96 mesh points.

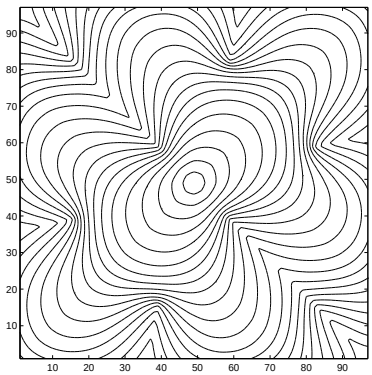

Figure 5: The value function for the min-time optimal trajectory problem. The vehicle's speed profile at every point (x, y) is the unit circle displaced by the vector $\boldsymbol{b}(x,y) = \frac{-.9\sin(4\pi x)\sin(4\pi y)}{\sqrt{x^2+y^2}}(x,y)$.

4.3 First arrivals in inhomogeneous anisotropic medium

Finally, we include an example of first arrival travel times with applications to seismic imaging. We start with a computational domain which suggests material layering under a sinusoidal profile. The computational domain is the square $[-a, a] \times [-a, a]$, with layer shapes

$$C(x) = A\sin\left(\frac{m\pi x}{a} + \beta\right) \tag{21}$$

where A is the amplitude of the sinusoidal profile, m is the number of periods, and β is the phase offset. The domain is split into n layers by the curves $y_i(x) = C(x) + b_i$, where $i = 1, \ldots, (n-1)$.

In each layer, the anisotropic speed at the point (x, y) is given by an ellipse with bigger axis (of length $2F_2$) tangential to the curve $C(x)$ and the smaller axis (of length $2F_1$) normal to the curve. F_1 and F_2 are constants in each layer. Thus, the ellipse's orientation and shape depend on the local position.

This leads to an anisotropic Hamilton-Jacobi equation of the form:

$$\|\nabla u(x,y)\| F = 1, \qquad u(0,0) = 0, \tag{22}$$

where the speed profile at any point (x, y) satisfies

$$F(x,y,u_x,u_y) = F_2\left(\frac{(1+q^2)u_x^2 + (1+p^2)u_y^2 - 2pqu_xu_y}{(1+p^2+q^2)(u_x^2+u_y^2)}\right)^{1/2}, \tag{23}$$

with

$$\begin{bmatrix} p \\ q \end{bmatrix} = \frac{\sqrt{\left(\frac{F_2}{F_1}\right)^2 - 1}}{\sqrt{1 + \left(\frac{dC}{dx}(x)\right)^2}} \begin{bmatrix} \frac{dC}{dx}(x) \\ -1 \end{bmatrix}.$$

Here, F_1 and F_2 are the ellipse semiaxes for the layer corresponding to the point (x, y).

The results are shown in Figure 6. We take $a = .5$, $A = .1225$, $m = 2$, $\beta = 0$, and layer offsets $b_i = (-.25, 0, 0.25)$. All calculations are on a 193×193 mesh. The (F_2, F_1) pair for each layer is given in the figures.

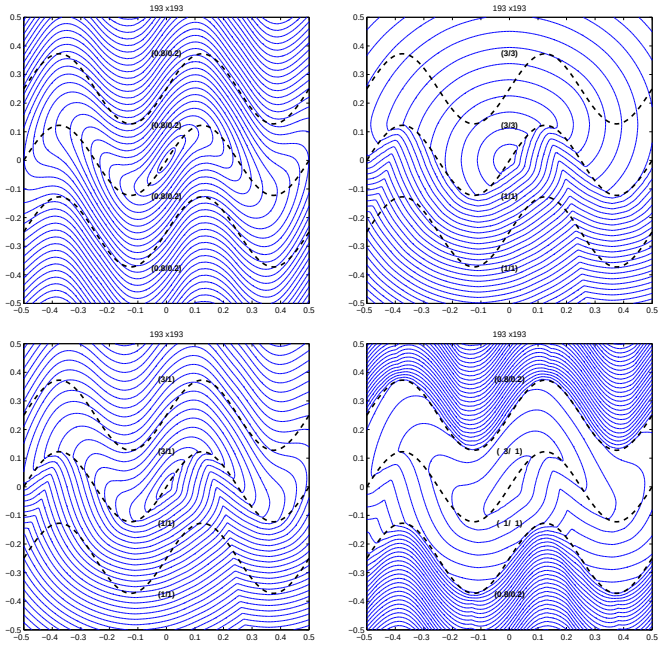

Figure 6: First arrivals in anisotropic media

Acknowledgements:

The authors thank L.C. Evans, M. Falcone and O. Hald. Supported by the Appl. Math. Sci. Office of Energy Research, U.S. Dept. Energy, # DE-AC03-76SF00098, ONR, FDN00014-96-1-0381, and the Division of Mathematical Sciences, NSF.

References

[1] Bellman, R. (1967) *Introduction to the Mathematical Theory of Control Processes* (Academic, New York),

[2] Crandall, M.G., Evans, L.C. & Lions, P-L. (1984), *Tran. AMS*, 282, 487-502.

[3] Crandall, M.G. & Lions, P-L. (1983) *Tran. AMS*, 277, 1-43.

[4] Dijkstra, E.W. (1959) *Numerische Mathematik*, 1, 269-271.

[5] Evans, L.C., *Partial Differential Equations*, American Mathematical Society, 1998.

[6] Falcone, M., Giorgi, T., Loreti, P., (1994) *SIAM J. Appl. Math.*, 54, 1335-1354.

[7] Gonzales, R. & Rofman, E. (1985) *SIAM J. Control Optim.*, 23, 2, 242-266.

[8] Helmsen, J., Puckett, E.G., Colella, P., and Dorr, M., *Two new methods for simulating photolithography development*, SPIE 1996 International Symposium on Microlithography, SPIE, v. 2726, June, 1996.

[9] Kimmel, R. & Sethian, J.A. (1998) *Proc. Nat. Acad. Sci.*, 95, 8341-8435.

[10] Kushner, H.J. & Dupuis, P.G. (1992) *Numerical Methods for Stochastic Control Problems in Continuous Time* (Springer-Verlag, New York).

[11] Malladi, R., and Sethian, J.A. (1996) *Proc. Nat. Acad. Sci.*, 93, 9389-9392.

[12] McGulagh, J. (1837) *Trans. Royal Irish Acad.*, 17, 241-263.

[13] Postma, G.W. (1955) *Geophysics*, 20, 780-806.

[14] Rouy, E. & Tourin, A. (1992) *SIAM J. Num. Anal.*, 29, 3, 867-884.

[15] Sethian, J.A. (1996) *Proc. Nat. Acad. Sci.*, 93, 4, 1591-1595.

[16] Sethian, J.A. (1999) *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*,(Cambridge University Press).

[17] Sethian, J.A., *Fast Marching Level Set Methods for Three-Dimensional Photolithography Development*, Proceedings, SPIE 1996 International Symposium on Microlithography, Santa Clara, California, June, 1996.

[18] Sethian, J.A. & Vladimirsky, A., *Fast Algorithms for Optimal Trajectory Problems*, to be submitted.

[19] Sethian, J.A. & Vladimirsky, A., (2000) *Proc. Nat. Acad. Sci.*, 97, 11, 5699-5703.

[20] Schulze, T.P. & Kohn, R.V., (1999) *Physica D*, 132, 520-542.

[21] Tsitsiklis, J.N. (1995) *IEEE Tran. Automatic Control*, 40, 1528-1538.

[22] Vladimirsky, A., *Fast Methods for Static Hamilton-Jacobi Partial Differential Equations*, Ph.D. Dissertation, Dept. of Mathematics, Univ. of California, Berkeley, 2001.