FAST MARCHING METHODS FOR A CLASS OF ANISOTROPIC STATIONARY HAMILTON-JACOBI EQUATIONS

KEN ALTON AND IAN M. MITCHELL*

Date. January 16, 2007

Technical Report Number. TR-2006-27

Submitted to. SIAM Journal on Numerical Analysis. Please do not redistribute.

Abstract. The Fast Marching Method (FMM) has proved to be a very efficient algorithm for solving the isotropic Eikonal equation. Because it is a minor modification of Dijkstra's algorithm for finding the shortest path through a discrete graph, FMM is also easy to implement. In this paper we describe a new class of Hamilton-Jacobi (HJ) PDEs with axis-aligned anisotropy which satisfy a causality condition for standard finite difference schemes on orthogonal grids and can hence be solved using the FMM; the only modification required to the algorithm is in the local update equation for a node. Since our class of HJ PDEs and grids permit asymmetries, we also examine some methods of improving the efficiency of the local update that do not require symmetric grids and PDEs. This class of HJ PDEs has applications in robotic path planning, and a brief example is included. In support of this and similar applications, we also include explicit update formulas for variations of the Eikonal equation that use the Manhattan, Euclidean and infinity norms on orthogonal grids of arbitrary dimension and with variable node spacing.

Key words. fast marching method, anisotropic optimal control, Hamilton-Jacobi equation, viscosity solution

AMS subject classifications. 35F30, 49L20, 49L25, 49N90, 65N06, 65N12

1. Introduction.

The Fast Marching Method (FMM) [23, 19] has become a popular algorithm to use when solving the Dirichlet problem for an isotropic static Hamilton-Jacobi Partial Differential Equation (HJ PDE), such as the Eikonal equation $\|Du(x)\|_2 = c(x)$. FMM has proven to be particularly efficient in practice because it can approximately solve this problem in a single pass through the nodes of a grid. It is also straightforward to implement, requiring only a small modification of Dijkstra's algorithm [8], which is a popular method for finding the shortest path through a discrete grid.

While the isotropic case is the most common, there are applications which require solution of anisotropic HJ PDEs. Unfortunately, FMM produces a correct approximation only under certain causality conditions on the values of nodes and their neighbors. This limitation has motivated the development of a more generally applicable version of FMM called the Ordered Upwind Method (OUM) [21] and also several recent works such as [25, 13, 18] on sweeping methods. However, OUM is much more complex to implement than FMM, and sweeping methods can be much less efficient for problems with curved characteristics [12, 10].

Consequently, we have motivation to seek classes of anisotropic problems to which FMM might still be applied. One such class of problems was identified in [20] and includes the Eikonal equation where an energy norm replaces the standard Euclidean norm. In [1] we identified another such class of problems. Because its characteristics are minimum time paths to the boundary, the Eikonal equation has often been

*Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada ({kalton, mitchell}@cs.ubc.ca, http://www.cs.ubc.ca/∼mitchell). This work was supported by a grant from the National Science and Engineering Research Council of Canada.

proposed for robotic path planning; for example, see [14]. However, for some robots using the Euclidean norm in this equation is inappropriate. Consider a robot arm, where each joint has its own motor. If each motor can rotate at some maximum speed independent of the action of the other motors, then the action of the whole arm is best bounded in an infinity norm. The corresponding Eikonal equation should use the dual Manhattan norm and is thus anisotropic. Other scenarios where such problems arise were considered in [1]—such as planning collision free optimal paths for multiple robots—and experimental evidence suggested that FMM would be successful on these problems.

As a group, the anisotropy in these problems is axis-aligned. In this paper we describe a broader class of such axis-aligned problems (Section 2), and demonstrate that FMM can be applied to approximate their solution on orthogonal grids without modification of the algorithm beyond the local update function for a single node (Section 3). We use a result from [3] to prove that the approximate solution converges to the solution of the HJ PDE (Section 4). We also demonstrate some methods by which the local update's efficiency can be improved even if the grid and/or PDE lack symmetry (Section 5). The examples (Section 6) include a new multirobot scenario, and we derive analytic update formulas for the Eikonal equation with the $p = 1$, 2 and ∞ norms on variably-spaced orthogonal grids in any dimension (Appendix B).

1.1. The Problem. The Dirichlet problem of a static HJ PDE is to find a function u, such that

$$H(x, Du(x)) = 0, \qquad x \in \Omega \tag{1.1a}$$

$$u(x) = g(x), \quad x \in \partial\Omega, \tag{1.1b}$$

where $Du(x)$ is the gradient of u at x, $\Omega \subset \mathbb{R}^d$ is a bounded Lipschitz domain, and $\partial\Omega$ is the domain's boundary. In general, it is not possible to find a classical solution to the Dirichlet problem (1.1) where u is differentiable for all x, so we seek instead the viscosity solution [7], a unique weak solution which is continuous and almost everywhere differentiable.

To appreciate the difference between an isotropic and anisotropic problem it is useful to consider a control-theoretic formulation of the Hamiltonian,

$$H(x, Du(x)) = \max_{a \in \mathcal{A}(x)} (-Du(x) \cdot a) - 1, \tag{1.2}$$

where a is an action and $\mathcal{A}(x) \subset \mathbb{R}^d$ is a compact, convex action set. In an isotropic problem $\mathcal{A}(x)$ is a hypersphere for all x, although its radius may depend on x. In such a problem (1.2) reduces to

$$H(x, Du(x)) = \|Du(x)\|_2 - c(x), \tag{1.3}$$

where $c(x) = 1/r(x)$ and $r(x)$ is the radius of the hyperspherical $\mathcal{A}(x)$. In this case (1.1a) becomes the Eikonal equation. In an anisotropic problem $\mathcal{A}(x)$ is not always a hypersphere.

1.2. The Fast Marching Method. Since we typically cannot solve for the viscosity solution exactly, we compute an approximate solution $\underline{u}$ on an orthogonal grid with nodes forming both a discretization $\underline{\Omega}$ of Ω, and a discretization $\underline{\partial\Omega}$ of $\partial\Omega$; for example, see Figure 1.2. We take $\underline{\Omega}$ and $\underline{\partial\Omega}$ to be disjoint sets. We allow any orthogonal grid, including those with node spacing that varies between dimensions

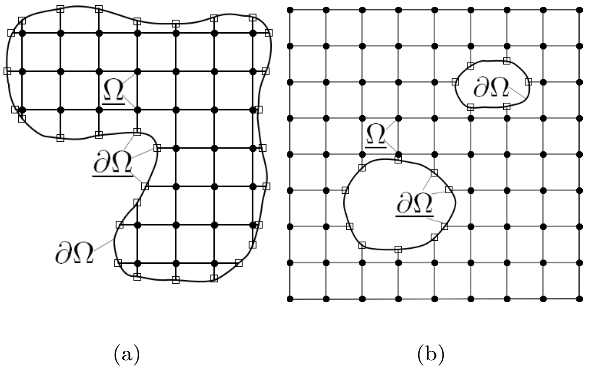


FIG. 1.1. *Orthogonal grids combining discretizations $\underline{\Omega}$ and $\underline{\partial\Omega}$. (a) boundary conditions are given around the outside of Ω. (b) boundary conditions are given on the inside of Ω.*

and within a single dimension; the latter capability makes it easier to more accurately manage an irregular boundary [10]. Informally, we refer to $\underline{u}(\underline{x})$ as the value of node $\underline{x}$. Let $\mathcal{N}(\underline{x})$ be the set of neighbors of node $\underline{x} \in \underline{\Omega}$.

Algorithm 1 outlines a simple dynamic programming algorithm. The algorithm can become either Dijkstra's algorithm or FMM depending on the choice of the `Update` function. For Dijkstra's algorithm, `Update` computes $\underline{u}(\underline{x}_0)$ as a simple minimization over the neighboring nodes of $\underline{x}_0$ of the path costs to $\underline{x}_0$ via each node. For FMM, the `Update` function computes $\underline{u}(\underline{x}_0)$ as a minimization over the neighboring simplices of $\underline{x}_0$ of the minimum path costs to $\underline{x}_0$ via each simplex. In both cases the `Update` function must satisfy a *causality* property in order for the algorithm to terminate with a correct solution: `Update` must compute a node value $\underline{u}(\underline{x})$ based only on information from neighboring nodes with smaller values, so that $\underline{u}$ is computed in increasing order of $\underline{u}(\underline{x})$. In Dijkstra's algorithm and FMM for a standard Euclidean norm Eikonal equation, this property is automatic. A major contribution of this paper is to demonstrate than an appropriate `Update` function can be defined for a class of static HJ PDEs with axis-aligned anisotropy such that the causality condition is fulfilled and FMM can be used.

```
foreach $\underline{x} \in \underline{\Omega}$ do $\underline{u}(\underline{x}) \leftarrow \infty$
foreach $\underline{x} \in \underline{\partial\Omega}$ do
    $\underline{u}(\underline{x}) \leftarrow g(\underline{x})$
    foreach $\underline{x}_0 \in \mathcal{N}(\underline{x}) \setminus \underline{\partial\Omega}$ do  $\underline{u}(\underline{x}_0) \leftarrow$ Update($\underline{x}_0$, $\underline{u}$)
end
$\mathcal{Q} \leftarrow \underline{\Omega}$
while $\mathcal{Q} \neq \emptyset$ do
    $\underline{x} \leftarrow$ ExtractMin($\mathcal{Q}$)
    foreach $\underline{x}_0 \in \mathcal{N}(\underline{x}) \cap \mathcal{Q}$ do  $\underline{u}(\underline{x}_0) \leftarrow$ Update($\underline{x}_0$, $\underline{u}$)
end
```

Algorithm 1: Dynamic Programming Algorithm.

While the `Update` function in Algorithm 1 is determined by the underlying equation which we seek to solve, it is assumed that its execution time is independent of grid resolution and hence it does not affect the algorithm's asymptotic complexity. The `Update` functions in this paper maintain this property. FMM is usually described as being $\mathcal{O}(n \log n)$, where $n = |\underline{\Omega}|$ is the number of grid points in the discretized domain. This complexity is derived by noting that each node is removed from $\mathcal{Q}$ once by `ExtractMin`; and, in the usual binary heap implementation of $\mathcal{Q}$, extraction of the minimum value node costs $\mathcal{O}(\log |\mathcal{Q}|) \leq \mathcal{O}(\log n)$. Because we restrict our modifications of Algorithm 1 to the `Update` function, all the results here can be used with other versions of FMM; for example, the $\mathcal{O}(n)$ algorithm described in [24], which uses an untidy priority queue for $\mathcal{Q}$ to reduce the cost of `ExtractMin` and hence the whole algorithm. However, for implementation simplicity we have used the standard binary heap version of $\mathcal{Q}$ in our experiments.

1.3. Related Work. The dynamic programming algorithm now called the Fast Marching Method was initially developed to solve isotropic Eikonal equations on orthogonal grids [23, 19]. By solving an isotropic problem on a manifold and then projecting the solution into a subspace, FMM can solve certain anisotropic problems [20]; for example, (1.2) with a constant elliptic $\mathcal{A}(x) = \mathcal{A}$ can be solved by running isotropic FMM on an appropriately tilted planar manifold and then projecting away one or more dimensions.

OUMs [21, 22] can solve general convex anisotropic problems on unstructured grids with an asymptotic complexity only a constant factor (related to the degree of anisotropy) worse than FMM. FMM fails for these general problems because the neighboring simplex from which the characteristic approaches a node $\underline{x}_0$ may contain another node $\underline{x}$ such that causality does not hold: $\underline{u}(\underline{x}_0) < \underline{u}(\underline{x})$. OUM avoids this difficulty by searching along the active front to find a set of neighboring nodes (which may not be direct neighbors of $\underline{x}_0$) whose values have been accepted, and then constructing a simplex with these nodes from which to update $\underline{u}(\underline{x}_0)$. However, there are some nontrivial additional data structures which must be implemented for OUM beyond those required by FMM in order for this search along the active front to not degrade the asymptotic complexity.

An alternative to these single pass (or label setting) algorithms are the sweeping (or label correcting) algorithms, which are often even simpler to implement than FMM. Sweeping algorithms are also capable of handling anisotropic and even nonconvex problems. The simplest sweeping algorithm is to just iterate through the grid updating each node in a Gauss-Seidel (GS) fashion (so a new value for a node is used immediately in subsequent updates) until $\underline{u}$ converges. GS converges quickly if the node update order is aligned with the characteristics of the solution, so better sweeping algorithms [15, 25, 13, 18] alternate among a collection of static node orderings so that all possible characteristic directions will align with at least one ordering. It is argued in [25] that these methods achieve $\mathcal{O}(n)$ asymptotic complexity (assuming that the node orderings are already determined); however, unlike FMM and OUM the constant depends on the problem. For practical grid resolutions on problems with curved characteristics FMM does better despite the difference in asymptotic complexity [12, 10].

There are also a number of sweeping algorithms which use dynamic node orderings; for example [17, 4]. These algorithms attempt to approximate (with varying degrees of accuracy) the optimal ordering generated by FMM without the overhead associated with managing an accurate queue. The dynamic ordering imposes slightly more implementation complexity than the static sweeping algorithms, but current indications are that they achieve results comparable to or better than FMM on isotropic problems. Because they still examine only neighboring simplices when computing an update, these schemes will likely need to revisit nodes multiple times for general anisotropic problems.

Accurate robotic path planning is only used in cluttered environments where optimal paths—and hence the characteristics of the HJ PDE—are not straight. No other algorithm proposed approaches the simple implementation and guaranteed speed of FMM for these types of problems. Consequently, we set out in this paper to characterize another class of anisotropic HJ PDEs for which FMM will work, and also to explore their efficient implementation. It should be noted that the update procedures discussed in this paper can be applied to any of the sweeping algorithms without modification; in fact, sweeping algorithms are likely to run faster on this restricted axis-aligned class of anisotropic problems than they might on other more general problems.

2. Class of Hamiltonians. FMM can be extended to handle a class of axis-aligned anisotropic problems, defined by a restriction of the Hamiltonian H to the form

$$H(x, Du(x)) = G(x, Du(x)) - c(x), \tag{2.1}$$

where $c : \mathbb{R}^d \to \mathbb{R}$ is a strictly positive *state-cost function* and $G : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a *gradient-size function*, satisfying Properties 1 to 4. We let $q, \tilde{q} \in \mathbb{R}^d$ and make the definitions:

Definition 2.1. *Write $q \trianglerighteq \tilde{q}$ if $q_j \tilde{q}_j \geq 0$ and $|q_j| \geq |\tilde{q}_j|, \forall j$ such that $1 \leq j \leq d$.*

Definition 2.2. *Write $q \triangleright \tilde{q}$ if (i) $q \neq 0$ and (ii) $q_j \tilde{q}_j \geq 0$ and $|q_j| > |\tilde{q}_j|$ or $q_j = \tilde{q}_j = 0$, $\forall j$, such that $1 \leq j \leq d$.*

The following properties in the second argument are satisfied by G.

Property 1. *G is nonnegative definite: $G(x, q) \geq 0$, for all $q \in \mathbb{R}^d$ and $G(x, q) = 0$ if and only if $q = 0$.*

Property 2. *G satisfies the triangle inequality: $G(x, q + \tilde{q}) \leq G(x, q) + G(x, \tilde{q})$, for all $q, \tilde{q} \in \mathbb{R}^d$.*

Property 3. *G is one-sided homogeneous: $G(x, tq) = tG(x, q)$, for all $t \geq 0$ and $q \in \mathbb{R}^d$.*

Property 4. *G is one-sided monotone: If $q \trianglerighteq \tilde{q}$, then $G(x, q) \geq G(x, \tilde{q})$.*

In this paper, we typically deal only with the `Update` function. For this reason, we usually consider a fixed $x \in \Omega$ in (2.1) and may write $G(q) = G(x, q)$, $c = c(x)$ and $\mathcal{A} = \mathcal{A}(x)$ wherever no ambiguity results. When discussing properties of G these are in reference to the q parameter. Properties 1 to 4 of G are closely related to those required of a norm [5, p. 634]. A norm $\| \cdot \|$ is nonnegative definite and satisfies the triangle inequality. Furthermore, a norm $\| \cdot \|$ is homogeneous , i.e., $\|tq\| = |t| \|q\|$, for all $t \in \mathbb{R}$ and $q \in \mathbb{R}^d$. This form of norm homogeneity is distinct from state homogeneity of the Hamiltonian, which occurs when $H(x, q) = H(q)$ for all $x \in \Omega$. Norm homogeneity is a more restrictive property than one-sided homogeneity. However, one-sided monotonicity is not required of a norm (see Section 2.1 for an example). The source of the *axis-aligned* description of the problem class is the one-sided monotonicity property of G.

In the following propositions, we show that Properties 1 to 4 of G imply other properties, which are used, for example, in Section 3.1.

Proposition 2.3. *If G satisfies Properties 1, 2, and 3, then G is convex and continuous.*

Proof. By the triangle inequality and one-sided homogeneity of G,

$$G(\kappa q + (1-\kappa)\tilde{q}) \leq G(\kappa q) + G((1-\kappa)\tilde{q}) = \kappa G(q) + (1-\kappa)G(\tilde{q}),$$

where $0 \leq \kappa \leq 1$, so G is convex. Moreover, G is continuous, since a convex function on an open convex subset of $\mathbb{R}^d$ is continuous [6]. □

Proposition 2.4. *If G satisfies Properties 1, 3, and 4, then G also satisfies* strict one-sided monotonicity*: If $q \rhd \tilde{q}$ then $G(q) > G(\tilde{q})$.*

Proof. Let $q \rhd \tilde{q}$. If $\tilde{q} = 0$, then $G(q) > G(\tilde{q}) = 0$, by the nonnegative definiteness of G. So, consider only the case where $\tilde{q} \neq 0$. Let $\mathcal{J} = \{j \mid |q_j| > |\tilde{q}_j|\}$. Note that by Definition 2.2 since $\tilde{q} \neq 0$, we have $\mathcal{J} \neq \emptyset$ and $\exists j \in \mathcal{J}$ such that $\tilde{q}_j \neq 0$. Define a scalar multiple of q:

$$\check{q} = tq = \left(\max_{j \in \mathcal{J}} \frac{|\tilde{q}_j|}{|q_j|}\right) q.$$

Since $|q_j| > |\tilde{q}_j|, \forall j \in \mathcal{J}$, we have $0 < t < 1$. Furthermore, for $j \in \mathcal{J}$,

$$|\check{q}_j| = \left(\max_{j \in \mathcal{J}} \frac{|\tilde{q}_j|}{|q_j|}\right) |q_j| \geq |\tilde{q}_j|,$$

while for $j \notin \mathcal{J}$,

$$\check{q}_j = tq_j = 0 = \tilde{q}_j.$$

Consequently, $|\check{q}_j| \geq |\tilde{q}_j|, \forall j$ such that $1 \leq j \leq d$. Also, since $t > 0$, we have $\check{q}_j \tilde{q}_j = tq_j \tilde{q}_j \geq 0, \forall j$ such that $1 \leq j \leq d$. This implies, by one-sided monotonicity of G, that $G(\check{q}) \geq G(\tilde{q})$. Moreover, by one-sided homogeneity of G, $G(\check{q}) = G(tq) = tG(q)$. It follows by nonnegativity of G that $G(q) = G(\check{q})/t > G(\tilde{q})$, since $0 < t < 1$. □

In understanding (2.1), it is useful to relate it to the control-theoretic (1.2). First, we define the dual of G:

$$G_*(q) = \max_{G(a) \leq 1} (q \cdot a). \tag{2.2}$$

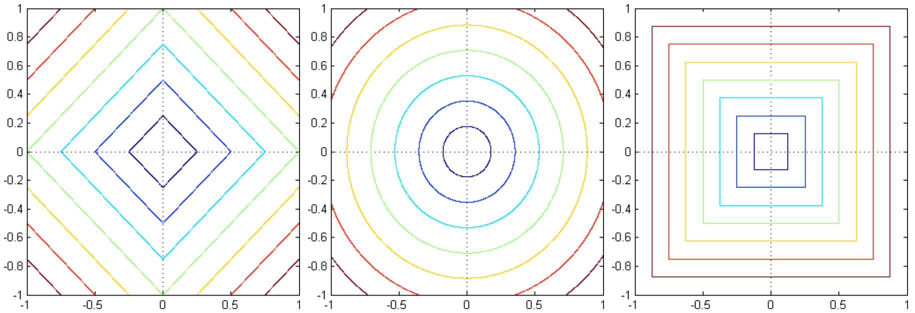
This definition of dual is similar to that of a dual norm (see Appendix A and [5, p. 637]). The following two propositions are proved in Appendix A.

Proposition 2.5. *If G satisfies Properties 1 to 4, then G_* also satisfies Properties 1 to 4.*

Proposition 2.6. *If G satisfies Properties 1 to 4, then $G = G_{**}$.*

We now demonstrate how the action set $\mathcal{A}$ can be defined to equate two formulations of (1.1), one with H defined by (2.1) and the other with H defined by (1.2). We note that [20] demonstrated a closely-related equivalence of two Hamiltonian formulations, a control-theoretic formulation using a directional speed function and a wavefront propagation formulation using a function that defines the speed of the wavefront in the normal direction. The two speed functions in that case are related by a homogeneous Legendre transform. However, we use the formulations (1.2) and (2.1) here, because they are particularly suited to understanding axis-aligned anisotropic problems and constructing numerical algorithms to solve them.

Proposition 2.7. *Let a function F satisfy Properties 1 to 4 and let $\mathcal{A} = \{a \mid F(a) \leq 1/c\}$. Furthermore, let $G(q) = F_*(-q)$, for all $q \in \mathbb{R}^d$. Then (1.1) with H defined by (2.1) and (1.1) with H defined by (1.2) are equivalent.*



(a) $p = 1$ (b) $p = 2$ (c) $p = \infty$

FIG. 2.1. *Contour plots of* $\|q\|_p$.

Proof. Let G be as defined above. Let $q = Du(x)$. We have

$$\begin{aligned} G(q) - c &= F_*(-q) - c \\ &= \max_{F(a)\leq 1} (-q \cdot a) - c \\ &= \max_{F(a/c)\leq 1/c} (-q \cdot a) - c \\ &= \max_{F(a)\leq 1/c} (-q \cdot ca) - c \\ &= c \max_{a\in\mathcal{A}} (-q \cdot a) - c \end{aligned} \tag{2.3}$$

where the third equality holds because of the one-sided homogeneity of F and the fourth equality makes the replacement $a \leftarrow ca$. Since $H = 0$ in (1.1a) and $c > 0$, the two forms of (1.1) are equivalent. □

In practice, one may be faced with the task of constructing a Hamiltonian of the form (2.1) given a control-theoretic formulation, such as (1.2) with action set $\mathcal{A}(x)$ or, equivalently, a directional speed function [22]. The proof of Proposition 2.7 explains this transformation in detail. Furthermore, the results in Appendix A show how G can be formed from F by decomposing F into orthant-associated norms, finding their duals, and reconstructing G orthant by orthant using these duals.

2.1. Example G Functions.

The Hamiltonian (2.1) encompasses a fairly broad range of anisotropic problems. We consider examples of G that satisfy Properties 1 to 4. In particular, we look at the case where G is a p-norm or some variant. We must ensure that G is one-sided monotone, which is not true of all norms.

The p-norm is a useful category of one-sided monotone norms. Let a p-norm, $\|\cdot\|_p$, be defined by

$$\|q\|_p = \left(\sum_{j=1}^{d} |q_j|^p \right)^{1/p},$$

where $p \geq 1$. Commonly used p-norms, illustrated in Figure 2.1, are the Manhattan norm ($p = 1$), the Euclidean norm ($p = 2$), and the maximum norm ($p = \infty$).

Proposition 2.8. *$\|\cdot\|_p$ is one-sided monotone.*

Proof. Let $q, \tilde{q} \in \mathbb{R}^d$. Let $|q_j| \geq |\tilde{q}_j|, \forall j$ such that $1 \leq j \leq d$. First, consider finite p. This implies $|q_j|^p \geq |\tilde{q}_j|^p \geq 0$, since x^p is nonnegative and nondecreasing on $x \geq 0$, when $p \geq 0$. This, in turn, implies

$$\sum_{j=1}^{d} |q_j|^p \geq \sum_{j=1}^{d} |\tilde{q}_j|^p \geq 0.$$

It follows that

$$\|q\|_p = \left(\sum_{j=1}^{d} |q_j|^p\right)^{1/p} \geq \left(\sum_{j=1}^{d} |\tilde{q}_j|^p\right)^{1/p} = \|\tilde{q}\|_p,$$

since $1/p \geq 0$.

Consider the case $p = \infty$ separately:

$$\|q\|_\infty = \lim_{p\to\infty} \left(\sum_{j=1}^{d} |q_j|^p\right)^{1/p} = \max_{1\leq j\leq d} |q_j|.$$

Since $|q_j| \geq |\tilde{q}_j|, \forall j$ such that $1 \leq j \leq d$, we have $\max_{1\leq j\leq d} |q_j| \geq \max_{1\leq j\leq d} |\tilde{q}_j|$.

Therefore, for both the case where p is finite and $p = \infty$, we have that $|q_j| \geq |\tilde{q}_j|, \forall j$ such that $1 \leq j \leq d$ implies $\|q\|_p \geq \|\tilde{q}\|_p$, a stronger condition than one-sided monotonicity. □

A linearly-transformed p-norm transforms its argument before applying a p-norm. Such a norm is not one-sided monotone in general. Let B be an nonsingular $d \times d$ matrix. Define a linearly-transformed p-norm, $\|\cdot\|_{B,p}$, to be

$$\|q\|_{B,p} = \|Bq\|_p,$$

where $p \geq 1$. Note B must be nonsingular so that $\|\cdot\|_{B,p}$ satisfies properties of a norm such as definiteness and homogeneity. As a simple example (Figure 2.2(a)), take the Euclidean norm of a vector that is rotated by $-\pi/4$ then scaled by 3 in the q_2-axis, i.e.,

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} \cos(-\pi/4) & -\sin(-\pi/4) \\ \sin(-\pi/4) & \cos(-\pi/4) \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -3/\sqrt{2} & 3/\sqrt{2} \end{bmatrix}. \tag{2.4}$$

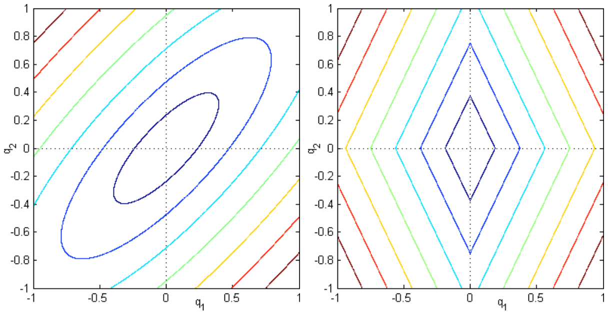
Let $q = (2, 2)^T$ and $\tilde{q} = (\sqrt{2}, 0)^T$. We have $q \trianglerighteq \tilde{q}$, but,

$$\|Bq\|_2 = \|(2\sqrt{2}, 0)^T\|_2 = \sqrt{8} \leq \sqrt{10} = \|(1, -3)^T\|_2 = \|B\tilde{q}\|_2.$$

Consequently, this particular linearly-transformed p-norm is not one-sided monotone.

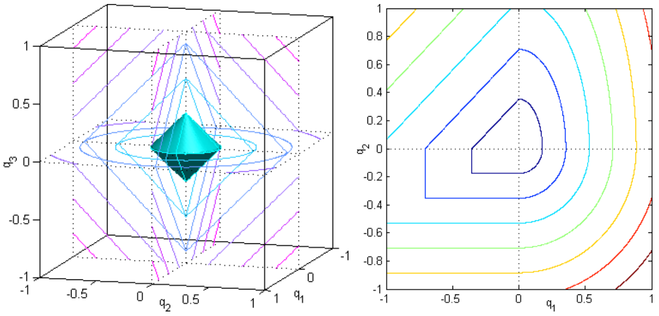
A scaled p-norm (Figure 2.2(b)) is a special case of a linearly-transformed p-norm. Such a norm scales the components of its argument before applying a p-norm, by restricting B to be a nonsingular diagonal matrix. It is simple to show that a scaled p-norm is one-sided monotone, considering Proposition 2.8.

A mixed p-norm is a recursive composition of p-norms and it is one-sided monotone. The following is an example (Figure 2.3(a)) of a mixed p-norm that takes the



(a) (b)

Fig. 2.2. *Contour plots of $\|Bq\|_p$. (a) is not one-sided monotone: $p = 2$ and B is defined by (2.4). (b) is one-sided monotone: $p = 1$ and B scales by 2 in the q_1-axis.*



(a) (b)

Fig. 2.3. *Contour plots of $G(q)$. (a) mixed p-norm: G is defined by (2.5). (b) asymmetric norm-like function: G is defined by (2.6).*

Euclidean norm of the first 2 components and then takes the Manhattan norm of the result and the last component:

$$\begin{aligned} \|q\| &= \| \left(\| \left(q_1, q_2 \right) \|_2, q_3 \right) \|_1 \\ &= \sqrt{(q_1)^2 + (q_2)^2} + |q_3|. \end{aligned} \tag{2.5}$$

where $q = (q_1, q_2, q_3)$. This particular norm was used as a G function in [1] for a simple 2-robot coordinated optimal control problem.

Finally, the one-sidedness of Properties 3 and 4 allows G to be asymmetric, which is not permitted for a norm. An example of such an asymmetric norm-like function is shown in Figure 2.3(b) and is given by

$$G(q) = \begin{cases} \|B_a q\|_\infty, & \text{if } q_1 \leq 0 \text{ and } q_2 \leq 0, \\ \|B_b q\|_1, & \text{if } q_1 \leq 0 \text{ and } q_2 > 0, \\ \|B_c q\|_2, & \text{if } q_1 > 0 \text{ and } q_2 \leq 0, \\ \|B_d q\|_2, & \text{if } q_1 > 0 \text{ and } q_2 > 0, \end{cases} \tag{2.6}$$

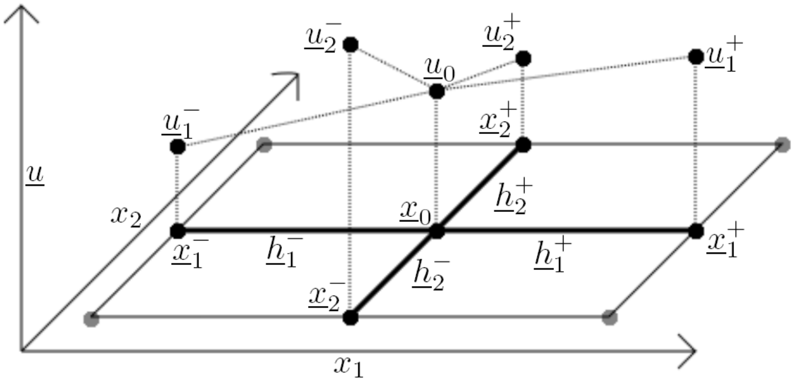


FIG. 3.1. *Neighborhood of $\underline{x}_0$ with $d = 2$.*

where

$$B_a = \begin{bmatrix} 1/2 & 0 \\ 0 & 1 \end{bmatrix} \quad B_b = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \quad B_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad B_d = \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix}.$$

We numerically solve examples of axis-aligned anisotropic problems, both symmetric and asymmetric, in Section 6.

3. FMM and the Discretized Problem.

We define a discretized analogue of the Dirichlet problem (1.1). By describing the `Update` function in Algorithm 1, we also formalize the FMM algorithm. Finally, we show that FMM solves the discretized problem.

Let $\underline{x}_0 \in \underline{\Omega}$. The neighborhood of $\underline{x}_0$ is shown in Figure 3.1. Let $\underline{x}_j^\pm$ be the neighbors of $\underline{x}_0$ in the $\pm e_j$ directions, e_j being the jth vector in the standard basis. The set of neighbors is

$$\mathcal{N}(\underline{x}_0) = \{\underline{x}_1^\pm, \underline{x}_2^\pm, \ldots, \underline{x}_d^\pm\}$$

and the neighborhood vector is

$$N(\underline{x}_0) = (\underline{x}_0, \underline{x}_1^\pm, \underline{x}_2^\pm, \ldots, \underline{x}_d^\pm).$$

Let $h_j^\pm = \pm\|\underline{x}_0 - \underline{x}_j^\pm\|$ be signed distances to the neighbors in the $\pm e_j$ directions. Let

$$\mathcal{S} = \{(s_1, s_2, \ldots, s_d) \mid s_j \in \{-1, +1\}, 1 \leq j \leq d\},$$

such that $s \in \mathcal{S}$ represents one of the 2^d neighboring simplices of $\underline{x}_0$. Note that we abuse notation by using $s_j \in \{-1, +1\}$ as a superscript indexing $\underline{x}_j^\pm$ or $h_j^\pm$.

Let $B(\Omega)$ be the set of bounded functions on domain Ω. We define the numerical Hamiltonian $\underline{H} : \Omega^{1+2d} \times B(\Omega) \times \mathbb{R} \to \mathbb{R}$ as follows:

$$\underline{H}(N, \phi, \mu) = \max_{s \in \mathcal{S}} [G(\underline{x}_0, \underline{D}^s(N, \phi, \mu))] - c(\underline{x}_0), \tag{3.1}$$

where G is as defined in Section 2 and

$$\underline{D}^s(N, \phi, \mu) = (\underline{D}_1^s(N, \phi, \mu), \underline{D}_2^s(N, \phi, \mu), \ldots, \underline{D}_d^s(N, \phi, \mu))$$

is a first-order, upwind, finite-difference gradient approximation from the simplex represented by s; that is,

$$\underline{D}_j^s(N, \phi, \mu) = \frac{\max(0, \mu - \phi(\underline{x}_j^{s_j}))}{-h_j^{s_j}}, \tag{3.2}$$

for $1 \leq j \leq d$. Although $\underline{H}$ is defined on domain $\Omega^{1+2d} \times B(\Omega) \times \mathbb{R}$, for FMM it will only be used on domain $\underline{\Omega}^{1+2d} \times B(\underline{\Omega}) \times \mathbb{R}$. The broader definition of domain is important for the consistency proof in Section 4.1. The restriction of Ω^{1+2d} to $\underline{\Omega}^{1+2d}$ poses no problems to the definition of $\underline{H}$. Furthermore, to evaluate $\underline{H}$, ϕ need only be defined on $\mathcal{N}$, which is true of any function in $B(\underline{\Omega})$.

The discretized Dirichlet problem is to find a function $\underline{u} : (\underline{\Omega} \cup \underline{\partial\Omega}) \to \mathbb{R}$, such that

$$\underline{H}(N(\underline{x}), \underline{u}, \underline{u}(\underline{x})) = 0, \qquad \underline{x} \in \underline{\Omega} \tag{3.3a}$$

$$\underline{u}(\underline{x}) = g(\underline{x}), \quad \underline{x} \in \underline{\partial\Omega}. \tag{3.3b}$$

Definition 3.1. *Let FMM be Algorithm 1 with the* `Update` *function defined as follows. A call to* `Update`$(\underline{x}_0, \underline{u})$ *returns the solution* $\mu = \tilde{\mu}$ *to*

$$\underline{H}(N(\underline{x}_0), \underline{u}, \mu) = 0. \tag{3.4}$$

In this way it determines a node's value $\underline{u}(\underline{x}_0) \leftarrow \tilde{\mu}$ given the values of its neighbors, $\underline{u}_j^{\pm} = \underline{u}(\underline{x}_j^{\pm})$. When we are varying only μ, it will be convenient to write $\underline{H}(\mu) = \underline{H}(N, \phi, \mu)$ and $\underline{D}^s(\mu) = \underline{D}^s(N, \phi, \mu)$. For the lemmas and theorems stated below we assume G satisfies Properties 1 to 4.

3.1. Unique Update.

Let the minimum value of all neighbours of $\underline{x}_0$ be

$$\check{\underline{u}} = \min_{\underline{x} \in \mathcal{N}(\underline{x}_0)} (\underline{u}(\underline{x})) . \tag{3.5}$$

We show there is a unique solution $\mu = \tilde{\mu}$ to (3.4), such that $\tilde{\mu} > \check{\underline{u}}$. First, we prove a useful lemma.

Lemma 3.2. *The numerical Hamiltonian, $\underline{H}(\mu)$, is strictly increasing on $\mu \geq \check{\underline{u}}$. Furthermore, $\underline{H}(\mu) = -c$ for $\mu \leq \check{\underline{u}}$. Finally, $\underline{H}(\mu)$ is nondecreasing on all μ.*

Proof. Let $\mu_a > \mu_b \geq \check{\underline{u}}$. Let $s \in \mathcal{S}$ and $1 \leq j \leq d$. If $\mu_a > \underline{u}_j^{s_j}$ then $\underline{D}_j^s(\mu_a)\underline{D}_j^s(\mu_b) \geq 0$ and $|\underline{D}_j^s(\mu_a)| > |\underline{D}_j^s(\mu_b)|$. On the other hand, if $\mu_a \leq \underline{u}_j^{s_j}$ then $\underline{D}_j^s(\mu_a) = \underline{D}_j^s(\mu_b) = 0$. Also, there exists $s \in \mathcal{S}$ and $1 \leq j \leq d$, such that $\underline{D}_j^s(\mu_a) \neq 0$, since $\mu_a > \check{\underline{u}}$. For such s, $G(\underline{D}^s(\mu_a)) > G(\underline{D}^s(\mu_b)) \geq 0$, by strict one-sided monotonicity (Proposition 2.4) and nonnegative definiteness of G. Otherwise, $G(\underline{D}^s(\mu_a)) = G(\underline{D}^s(\mu_b)) = 0$. Consequently, $\underline{H}(\mu_a) > \underline{H}(\mu_b)$, so $\underline{H}$ is strictly increasing on $\mu \geq \check{\underline{u}}$, proving the first claim.

If $\mu \leq \check{\underline{u}}$ then

$$\underline{D}_j^s(\mu) = \frac{\max(0, \mu - \underline{u}_j^{s_j})}{-h_j^{s_j}} = 0,$$

for all $s \in \mathcal{S}$, $1 \leq j \leq d$. By the nonnegative definiteness of G, $G(\underline{D}^s(v_j)) = 0$, for all s. Therefore, by (3.1), $\underline{H}(\mu) = -c$, proving the second claim.

Because $\underline{H}(\mu)$ is constant on $\mu \leq \check{\underline{u}}$ and increasing on $\mu \geq \check{\underline{u}}$, $\underline{H}(\mu)$ is nondecreasing on all μ. □

THEOREM 3.3. *There exists a unique solution, $\mu = \tilde{\mu}$, to $\underline{H}(\mu) = 0$, such that $\tilde{\mu} > \check{\underline{u}}$.*

Proof. By Lemma 3.2, $\underline{H}(\mu) = -c$ for $\mu \leq \check{\underline{u}}$. Since $c > 0$, $\underline{H}(\mu) < 0$ for $\mu \leq \check{\underline{u}}$.

Take any $s \in \mathcal{S}$ and $1 \leq j \leq d$. Let $K = G(-s_j e_j)$, where $K > 0$, by the nonnegative definiteness of G. Note that $(-s_j)\underline{D}^s_j(\mu) \geq 0$ and $(-s_j)\underline{D}^s_j(\mu)$ is unbounded on μ. Choose $\hat{\mu}$ such that $\underline{D}^s_j(\hat{\mu}) = t(-s_j)e_j$, where $t > c/K > 0$. Note that $\underline{D}^s(\hat{\mu}) \trianglerighteq t(-s_j)e_j$, where $\trianglerighteq$ is as in Definition 2.1. By the one-sided monotonicity and one-sided homogeneity of G we have

$$G(\underline{D}^s(\hat{\mu})) \geq G(t(-s_j)e_j) = tG(-s_j e_j) > c.$$

As a result, $\underline{H}(\hat{\mu}) > 0$.

Each $\underline{D}^s_j(\mu)$ is continuous on μ. Furthermore, by the continuity of G, $G(\underline{D}^s(\mu))$ in continuous on μ, for all s. Since max is continuous, $\underline{H}(\mu)$ is continuous. Therefore, since $\underline{H}(\check{\underline{u}}) < 0$ and $\underline{H}(\hat{\mu}) > 0$, by the Intermediate Value Theorem, there exists a solution, $\mu = \tilde{\mu}$, to $\underline{H}(\mu) = 0$, such that $\check{\underline{u}} < \tilde{\mu} < \hat{\mu}$. Moreover, since $\underline{H}$ is strictly increasing on $\mu \geq \check{\underline{u}}$ by Lemma 3.2, the solution is unique. □

3.2. Monotonicity. We show that $\underline{H}$ is monotone in the neighbor's values. Monotonicity requires that if none of the neighbor's values decreases, the numerical Hamiltonian $\underline{H}$ should not increase. Monotonicity is useful both for showing that FMM finds a unique solution to (3.3) and for proving convergence in Section 4.

THEOREM 3.4. *Let $\underline{v}$ and $\underline{u}$ be grid functions. Let $\underline{v}^{\pm}_j \geq \underline{u}^{\pm}_j$ for all j, such that $1 \leq j \leq d$. Then for $\mu \in \mathbb{R}$, we have $\underline{H}(N, \underline{v}, \mu) \leq \underline{H}(N, \underline{u}, \mu)$. Furthermore, if $\mu = \mu_v$ is the unique solution to $\underline{H}(N, \underline{v}, \mu) = 0$ and $\mu = \mu_u$ is the unique solution to $\underline{H}(N, \underline{u}, \mu) = 0$, then $\mu_v \geq \mu_u$.*

Proof. Let $\mu \in \mathbb{R}$. We have $\underline{D}^s(N, \underline{u}, \mu) \trianglerighteq \underline{D}^s(N, \underline{v}, \mu)$, for all $s \in \mathcal{S}$. By one-sided monotonicity of G (Property 4), $G(\underline{D}^s(N, \underline{u}, \mu)) \geq G(\underline{D}^s(N, \underline{v}, \mu)) = 0$, for all $s \in \mathcal{S}$. Consequently, $\underline{H}(N, \underline{u}, \mu) \geq \underline{H}(N, \underline{v}, \mu)$, proving the first claim.

To prove the second claim, we let μ_v and μ_u be as defined above. We note that $\underline{H}(N, \underline{u}, \mu_u) = 0 \geq \underline{H}(N, \underline{v}, \mu_u)$. By Lemma 3.2, $\underline{H}(N, \underline{v}, \mu)$ is nondecreasing on all μ, so in order that $\underline{H}(N, \underline{v}, \mu_v) = 0$, it must be that $\mu_v \geq \mu_u$. □

3.3. Causality. We note that (3.3) defines a very large system of nonlinear equations, one equation for each node $\underline{x} \in \Omega$. FMM can be used to solve this system very efficiently, if the solution $\mu = \tilde{\mu}$ to (3.4) is dependent only on neighbors with smaller values. This property represents a causal relationship between node values. There is an information flow from nodes with smaller values to those with larger values. The causal relationship is meant to mimic that of the PDE (1.1). The solution u of (1.1) is completely defined at x using only values of u from states that are backwards along the characteristic line that passes through x.

FMM exploits the causal property of $\underline{H}$ by computing $\underline{u}(\underline{x})$ in increasing order in a single pass through the nodes. The following theorem proves that $\underline{H}$ and the `Update` function are causal.

THEOREM 3.5. *Let $\underline{v}$ and $\underline{u}$ be grid functions. Let*

$$\tilde{\mathcal{N}}(\underline{x}_0) = \{\underline{x} \in \mathcal{N}(\underline{x}_0) \mid \underline{v}(\underline{x}) \neq \underline{u}(\underline{x})\}.$$

Let

$$\check{w} = \begin{cases} \min_{\underline{x} \in \tilde{\mathcal{N}}(\underline{x}_0)} \min(\underline{v}(\underline{x}), \underline{u}(\underline{x})), & \textit{if } \tilde{\mathcal{N}}(\underline{x}_0) \neq \emptyset, \\ +\infty, & \textit{otherwise.} \end{cases}$$

Then $\underline{H}(N, \underline{v}, \mu) = \underline{H}(N, \underline{u}, \mu)$, for $\mu \leq \check{w}$.

Furthermore, let $\mu = \tilde{\mu}_u$ be the unique solution to $\underline{H}(N, \underline{u}, \mu) = 0$, and $\mu = \tilde{\mu}_v$ be the unique solution to $\underline{H}(N, \underline{v}, \mu) = 0$. If $\tilde{\mu}_u \leq \check{w}$ or $\tilde{\mu}_v \leq \check{w}$ then $\tilde{\mu}_u = \tilde{\mu}_v$.

Proof. Let $\mu \leq \check{w}$. By (3.2) and the definition of $\check{w}$, we have $D^s_j(N, \underline{v}, \mu) = D^s_j(N, \underline{u}, \mu)$, for all $s \in \mathcal{S}$, $1 \leq j \leq d$. This implies that $\underline{H}(N, \underline{v}, \mu) = \underline{H}(N, \underline{u}, \mu)$, proving the first claim.

For the second claim, let $\tilde{\mu}_u$ and $\tilde{\mu}_v$ be as defined above. Let $\tilde{\mu}_u \leq \check{w}$. Then $\underline{H}(N, \underline{v}, \tilde{\mu}_u) = \underline{H}(N, \underline{u}, \tilde{\mu}_u) = 0$, so $\mu = \tilde{\mu}_u$ is a solution to $\underline{H}(N, \underline{v}, \mu) = 0$. By Theorem 3.3, this solution is unique. By a symmetric argument, if $\tilde{\mu}_v \leq \check{w}$ then $\mu = \tilde{\mu}_v$ is the unique solution to $\underline{H}(N, \underline{u}, \mu) = 0$. □

3.4. Solution. We show that FMM finds a unique solution to (3.3). The results in this section are based on those in the original FMM papers [23, 19].

LEMMA 3.6. *Let $|\underline{\Omega}|$ be finite. If* `ExtractMin` *and* `Update` *always terminate then Algorithm 1 terminates with at most $2d|\underline{\Omega}|$ calls to* `Update`*.*

Proof. Since $\mathcal{Q}$ is initialized to $\underline{\Omega}$ in Line 6 of Algorithm 1, it has a finite number of elements. For every iteration of the **while** loop one element is removed from $\mathcal{Q}$ and at most $2d$ neighbors are updated. The **while** loop terminates when $\mathcal{Q} = \emptyset$. Therefore, there are at most $|\underline{\Omega}|$ iterations of the **while** loop and at most $2d|\underline{\Omega}|$ calls to `Update`. □

As Algorithm 1 is run the grid function $\underline{u}$ evolves. Accordingly, it is useful to talk about a sequence of grid functions $\underline{u}^k$. Let $\underline{u}^0$ be the state of $\underline{u}$ just after initialization on Line 6 in Algorithm 1. Let $\underline{u}^k$ for $k \geq 1$ be the state of the grid function after the kth call to `Update`. For two grid functions $\underline{u}$ and $\tilde{\underline{u}}$, we say that $\underline{u} \leq \tilde{\underline{u}}$ if and only if $\underline{u}(\underline{x}) \leq \tilde{\underline{u}}(\underline{x})$ for all $\underline{x} \in (\underline{\Omega} \cup \underline{\partial\Omega})$.

LEMMA 3.7. *The sequence of grid functions $\underline{u}^k$ is nonincreasing on k.*

Proof. Consider a node $\underline{x} \in \underline{\Omega}$ being updated using

$$\underline{u}^{k'+1}(\underline{x}) \leftarrow \texttt{Update(}\underline{x}\texttt{, }\underline{u}^{k'}\texttt{)}.$$

Assume that the subsequence $\underline{u}^l$ is nonincreasing on $0 \leq l \leq k'$. Either $\underline{x}$ has been updated before or it has not and $\underline{u}^{k'}(\underline{x}) = \infty$. In the latter case, certainly $\underline{u}^{k'+1}(\underline{x}) \leq \underline{u}^{k'}(\underline{x}) = \infty$. In the former case, let the previous update of $\underline{x}$ be

$$\underline{u}^{k''+1}(\underline{x}) \leftarrow \texttt{Update(}\underline{x}\texttt{, }\underline{u}^{k''}\texttt{)},$$

where $k'' < k'$. By the assumption, $\underline{u}^{k'} \leq \underline{u}^{k''}$. By Theorem 3.4,

$$\underline{u}^{k'+1}(\underline{x}) \leq \underline{u}^{k''+1}(\underline{x}) = \underline{u}^{k'}(\underline{x}).$$

Consequently, $\underline{u}^{k'+1} \leq \underline{u}^{k'}$. We also have $\underline{u}^1 \leq \underline{u}^0$ since the first update to any node $\underline{x} \in \underline{\Omega}$ cannot be such that $\underline{u}^1(\underline{x}) > \underline{u}^0(\underline{x}) = \infty$. Therefore, by induction on k, $\underline{u}^k$ is a nonincreasing sequence. □

THEOREM 3.8. *Let $\underline{u} : (\underline{\Omega} \cup \underline{\partial\Omega}) \to \mathbb{R}$ be the grid function after FMM terminates. Then $\underline{u}$ is the unique solution of (3.3).*

If a node $\underline{x} \notin \mathcal{Q}$ we say that it is *known*, meaning that its value can no longer be changed by Algorithm 1. If a node $\underline{x} \in \mathcal{Q}$ we say that it is *estimated*, meaning that its value can still be changed. When a node $\underline{x}$ is extracted from $\mathcal{Q}$ we say that it *becomes known.* Let $\mathcal{Q}^l$ be a sequence of node sets such that $\mathcal{Q}^0 = \underline{\Omega}$ and $\mathcal{Q}^l$ for $l \geq 1$ is the state of $\mathcal{Q}$ after the lth call to `ExtractMin`.

We prove the theorem in two stages. First, we show that when a node $\underline{x}$ *becomes known*, (3.3a) is satisfied for that node. Then, we show that after $\underline{x}$ *becomes known* only neighbors' values not less than $\underline{u}(\underline{x})$ may change and the changed neighbors' values will not be less than $\underline{u}(\underline{x})$. Using Theorem 3.5, (3.3a) remains satisfied for $\underline{x}$ that is *known*.

Proof. Consider a node $\underline{x} \in \underline{\Omega}$ that has just become known, such that $\mathcal{Q}^{l'+1} = \mathcal{Q}^{l'} \setminus \{\underline{x}\}$. Let $\underline{u}^{k'}$ be the grid function when $\underline{x}$ becomes known. Since $\underline{u}^{k'}(\underline{x})$ can no longer change, we say $\underline{u}(\underline{x}) = \underline{u}^{k'}(\underline{x})$. Let the previous update of $\underline{x}$ be

$$\underline{u}^{k''+1}(\underline{x}) \leftarrow \texttt{Update(}\underline{x}\texttt{, }\underline{u}^{k''}\texttt{)},$$

where $k'' + 1 \leq k'$. Note that $\underline{u}(\underline{x}) = \underline{u}^{k''+1}(\underline{x})$ and

$$\underline{H}(N(\underline{x}), \underline{u}^{k''+1}, \underline{u}^{k''+1}(\underline{x})) = 0.$$

Let $\mathcal{Q}^{l''}$ be the state of $\mathcal{Q}$ at the time of this previous update. Since the previous update of $\underline{x}$ occurred just after the last time a neighbor $\underline{\tilde{x}} \in \mathcal{N}(\underline{x})$ became known, we have $\mathcal{Q}^{l'} \cup \mathcal{N}(\underline{x}) = \mathcal{Q}^{l''} \cup \mathcal{N}(\underline{x})$. But any neighbor $\underline{\tilde{x}} \in \mathcal{N}(\underline{x})$ for which $\underline{u}^{k'}(\underline{\tilde{x}}) \neq \underline{u}^{k''+1}(\underline{\tilde{x}})$ must be in $\mathcal{Q}^{l'} \cup \mathcal{N}(\underline{x})$, since only *estimated* nodes are updated. Also, since `ExtractMin` removes the *estimated* node with minimum value, $\underline{u}^{k'}(\underline{\tilde{x}}) \geq \underline{u}(\underline{x})$, for all $\underline{\tilde{x}} \in \mathcal{Q}^{l'} \cup \mathcal{N}(\underline{x})$. Furthermore, $\underline{u}^{k''+1}(\underline{\tilde{x}}) \geq \underline{u}(\underline{x})$, for all $\underline{\tilde{x}} \in \mathcal{Q}^{l'} \cup \mathcal{N}(\underline{x})$, since $\underline{u}^{k''+1} \geq \underline{u}^{k'}$ by Lemma 3.7. By Theorem 3.5,

$$\underline{H}(N(\underline{x}), \underline{u}^{k'}, \underline{u}(\underline{x})) = \underline{H}(N(\underline{x}), \underline{u}^{k''+1}, \underline{u}^{k''+1}(\underline{x})) = 0.$$

Now consider the ith neighbor $\underline{x}^i \in \mathcal{N}(\underline{x}) \cap Q^{l'+1}$ updated after $\underline{x}$ became known but before any other *estimated* nodes become known with

$$\underline{u}^{k'+i}(\underline{x}^i) \leftarrow \texttt{Update(}\underline{x}^i\texttt{, }\underline{u}^{k'+i-1}\texttt{)}.$$

Let the previous update of $\underline{x}^i$ be

$$\underline{u}^{k'''+1}(\underline{x}^i) \leftarrow \texttt{Update(}\underline{x}^i\texttt{, }\underline{u}^{k'''}\texttt{)}, \tag{3.6}$$

where $k''' + 1 \leq k' \leq k' + i - 1$. Assume $\underline{u}^{k'+i-1}(\underline{y}) \geq \underline{u}(\underline{x})$, for all $\underline{y} \in \mathcal{Q}^{l'}$. We also have $\underline{u}^{k'''+1}(\underline{y}) \geq \underline{u}(\underline{x})$, for all $\underline{y} \in \mathcal{Q}^{l'}$, since $\underline{u}^{k'''+1} \geq \underline{u}^{k'+i-1}$ by Lemma 3.7.

Let $\tilde{x} \in \mathcal{N}(\underline{x}^i)$ be such that $\underline{u}^{k'+i-1}(\tilde{x}) \neq \underline{u}^{k'''+1}(\tilde{x})$. Such $\tilde{x}$ could not have become known before $\underline{x}$ became known, because the previous update (3.6) of $\underline{x}^i$ would follow and that would imply $\underline{u}^{k'+i-1}(\tilde{x}) = \underline{u}^{k'''+1}(\tilde{x})$, a contradiction. Consequently, if $\tilde{x} \in \mathcal{N}(\underline{x}^i)$ and $\underline{u}^{k'+i-1}(\tilde{x}) \neq \underline{u}^{k'''+1}(\tilde{x})$, then $\tilde{x} \in \mathcal{Q}^{l'}$ and $\underline{u}^{k'''+1}(\tilde{x}) > \underline{u}^{k'+i-1}(\tilde{x}) \geq \underline{u}(\underline{x})$. By Theorem 3.5, if $\underline{u}^{k'+i}(\underline{x}^i) \neq \underline{u}^{k'''+1}(\underline{x}^i)$, then $\underline{u}^{k'+i}(\underline{x}^i) \geq \underline{u}(\underline{x})$. On the other hand, we have $\underline{u}^{k'+i}(\underline{x}^i) = \underline{u}^{k'+i-1}(\underline{x}^i)$ and by the assumption $\underline{u}^{k'+i}(\underline{x}^i) \geq \underline{u}(\underline{x})$. So, for all $\underline{y} \in \mathcal{Q}^{l'}$, $\underline{u}^{k'+i}(\underline{y}) \geq \underline{u}(\underline{x})$.

We also know that $\underline{u}^{k'}(\underline{y}) \geq \underline{u}(\underline{x})$, for all $\underline{y} \in \mathcal{Q}^{l'}$, since `ExtractMin` removes the *estimated* node with minimum value. By induction on i, $\underline{u}^{k'+i}(\underline{y}) \geq \underline{u}(\underline{x})$, for all $\underline{y} \in \mathcal{Q}^{l'}$, for all i such that $1 \leq i \leq |\mathcal{N}(\underline{x}) \cap \mathcal{Q}^{l'+1}|$.

By induction on the sequence of nodes to become known after $\underline{x}$ becomes known we find that any such node $\underline{y}$ must be such that $\underline{u}(\underline{y}) \geq \underline{u}(\underline{x})$. Note that only nodes $\underline{y}$ that become known after $\underline{x}$ becomes known may have values such that $\underline{u}(\underline{y}) \neq \underline{u}^{k'}(\underline{y})$ Therefore, by Theorem 3.5,

$$\underline{H}(N(\underline{x}), \underline{u}, \underline{u}(\underline{x})) = \underline{H}(N(\underline{x}), \underline{u}^{k'}, \underline{u}(\underline{x})) = 0.$$

Since, by Lemma 3.6, every node $\underline{x} \in \underline{\Omega}$ *becomes known* in the course of Algorithm 1, $\underline{u}$ is a solution of (3.3). By Theorem 3.3, this solution is unique. □

4. Convergence. If the update scheme is consistent, monotone, and stable [3] the FMM-computed $\underline{u}$ will converge to the viscosity solution u of (1.1). Monotonicity is proven in Section (3.2).

4.1. Consistency. We show that the numerical Hamiltonian $\underline{H}$ is consistent with (2.1). Let $C_b^\infty(\Omega)$ be the set of smooth, bounded functions on domain Ω.

THEOREM 4.1. *Let $\phi \in C_b^\infty(\Omega)$. Let $x \in \Omega$. Let G be continuous in the first argument and satisfy Properties 1 to 4. Let c be continuous. Then*

$$\lim_{\substack{\underline{x}_0 \to x,\ \xi \to 0 \\ h_j^{s_j} \to 0,\ 1 \le j \le d}} \underline{H}(N, \phi + \xi, \phi(\underline{x}_0) + \xi) = H(x, D\phi(x)). \tag{4.1}$$

Proof. Let ϕ, x, G, and c be as defined above. Let

$$D\phi(x) = (\partial_1 \phi(x), \partial_2 \phi(x), \ldots, \partial_d \phi(x)).$$

Let $s \in \mathcal{S}$ and $1 \le j \le d$. We have by (3.2) and the smoothness of ϕ

$$\begin{aligned}
\lim_{\substack{\underline{x}_0 \to x,\ \xi \to 0, \\ h_j^{s_j} \to 0}} D_j^s(N, \phi + \xi, \phi(\underline{x}_0) + \xi) &= \lim_{\substack{\underline{x}_0 \to x,\ \xi \to 0, \\ h_j^{s_j} \to 0}} \frac{\max(0, \phi(\underline{x}_0) + \xi - \phi(\underline{x}_j^{s_j}) - \xi)}{-h_j^{s_j}} \\
&= \lim_{\substack{\underline{x}_0 \to x \\ h_j^{s_j} \to 0}} \frac{\min(0, \phi(\underline{x}_0 + h_j^{s_j} e_j) - \phi(\underline{x}_0))}{h_j^{s_j}} \\
&= \begin{cases} \partial_j \phi(x), & \text{if } s_j \partial_j \phi(x) \le 0, \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}$$

Define $\tilde{s}$ as

$$\tilde{s}_j = \begin{cases} +1, & \text{if } \partial_j \phi(x) \le 0, \\ -1, & \text{otherwise,} \end{cases} \tag{4.2}$$

for $1 \le j \le d$. We have

$$\lim_{\substack{\underline{x}_0 \to x,\ \xi \to 0, \\ h_j^{s_j} \to 0,\ 1 \le j \le d}} D^{\tilde{s}}(N, \phi + \xi, \phi(\underline{x}_0) + \xi) = D\phi(x). \tag{4.3}$$

By the continuity and one-sided monotonicity of G,

$$\lim_{\substack{\underline{x}_0 \to x,\ \xi \to 0, \\ h_j^{s_j} \to 0,\ 1 \le j \le d}} G(\underline{x}_0, D^{\tilde{s}}(N, \phi + \xi, \phi(\underline{x}_0) + \xi)) \ge \lim_{\substack{\underline{x}_0 \to x,\ \xi \to 0, \\ h_j^{s_j} \to 0,\ 1 \le j \le d}} G(\underline{x}_0, D^s(N, \phi + \xi, \phi(\underline{x}_0) + \xi)),$$

for all s. Therefore, by the continuity of max, G, and c

$$\begin{aligned}
&\lim_{\substack{\underline{x}_0\to x,\ \xi\to 0,\\ h_j^{s_j}\to 0,\ 1\le j\le d}} \underline{H}(N, \phi+\xi, \phi(\underline{x}_0)+\xi)\\
&= \lim_{\substack{\underline{x}_0\to x,\ \xi\to 0,\\ h_j^{s_j}\to 0,\ 1\le j\le d}} \left[\max_{s\in\mathcal{S}} G(\underline{x}_0, D^s(N, \phi+\xi, \phi(\underline{x}_0)+\xi)) - c(\underline{x}_0)\right]\\
&= \lim_{\substack{\underline{x}_0\to x,\ \xi\to 0,\\ h_j^{s_j}\to 0,\ 1\le j\le d}} \left[G(\underline{x}_0, D^{\tilde{s}}(N, \phi+\xi, \phi(\underline{x}_0)+\xi))\right] - c(x)\\
&= G(x, D\phi(x)) - c(x) = H(x, D\phi(x)).
\end{aligned}$$

□

4.2. Stability. We show that for an orthogonal discretization $\underline{\Omega}$, the solution $\underline{u}^{\Omega}$ to the discretized problem is bounded. We begin by showing with a lemma that the magnitude of the slope in $\underline{u}^{\Omega}$ as measured between two neighbors is bounded.

LEMMA 4.2. *Let*

$$\hat{K} = \max_{\substack{x\in\Omega,\\ s\in\mathcal{S}, 1\le j\le d}} \frac{c(x)}{G(x, s_j e_j)}. \tag{4.4}$$

Let $\underline{x}_0 \in \underline{\Omega}$ *and* $\underline{H}(N(\underline{x}_0), \underline{u}, \mu) = 0$. *Then*

$$|D_j^s(N(\underline{x}_0), \underline{u}, \mu)| \le \hat{K},$$

for all $s \in \mathcal{S}$ *and* $1 \le j \le d$.

Proof. Assume

$$|D_j^s(N(\underline{x}_0), \underline{u}, \mu)| > \frac{c(\underline{x}_0)}{G(\underline{x}_0, -s_j e_j)}$$

By the one-sided monotonicity of G, the definition of D_j^s and the one-sided homogeneity of G we have

$$\begin{aligned}
G(\underline{x}_0, D^s(N(\underline{x}_0), \underline{u}, \mu)) &\ge G(\underline{x}_0, D_j^s(N(\underline{x}_0), \underline{u}, \mu) e_j)\\
&= G\left(\underline{x}_0, \frac{\max(0, \mu - \underline{u}(\underline{x}_j^{s_j}))}{-h_j^{s_j}} e_j\right)\\
&= G(\underline{x}_0, -|D_j^s(N(\underline{x}_0), \underline{u}, \mu)| s_j e_j)\\
&= |D_j^s(N(\underline{x}_0), \underline{u}, \mu)| G(\underline{x}_0, -s_j e_j)\\
&> c(\underline{x}_0),
\end{aligned}$$

contradicting the fact that $\underline{H}(N(\underline{x}_0), \underline{u}, \mu) = 0$. Therefore,

$$|D_j^s(N(\underline{x}_0), \underline{u}, \mu)| \le \frac{c(\underline{x}_0)}{G(\underline{x}_0, -s_j e_j)} \le \hat{K}.$$

□

Let $\underline{X} = (\underline{x}_1, \underline{x}_2, \dots \underline{x}_k)$ be a sequence of neighboring nodes, such that $\underline{x}_l \in \underline{\Omega} \cup \underline{\partial\Omega}$ for $1 \le l \le k$ and $\underline{x}_l \in \mathcal{N}(\underline{x}_{l-1})$ for $2 \le l \le k$. Define the grid path length of $\underline{X}$ by

$$\omega(\underline{X}) = \sum_{l=2}^{k} \|\underline{x}_l - \underline{x}_{l-1}\|_2.$$

Define the minimum grid path length between $\underline{x}$ and $\underline{x}'$ to be

$$\breve{\omega}(\underline{x}, \underline{x}') = \min\{\omega(\underline{X}) \mid \underline{x}_1 = \underline{x} \,\text{and}\, \underline{x}_k = \underline{x}'\}.$$

Finally, define the minimum node-to-boundary grid path length of $\underline{x}$ as

$$\tilde{\omega}(\underline{x}) = \min_{\underline{x}' \in \underline{\partial\Omega}} \breve{\omega}(\underline{x}, \underline{x}').$$

Let $\underline{u}^{\underline{\Omega}} : \underline{\Omega} \cup \underline{\partial\Omega} \to \mathbb{R}$ be the solution to the discretized problem computed by FMM on the grid $\underline{\Omega}$. Let $u^{\underline{\Omega}} : \Omega \to \mathbb{R}$ extend $\underline{u}^{\underline{\Omega}}$ to the continuous domain using some non-expansive interpolator, such as nearest-neighbor or linear interpolation, for values between nodes. We show that for a reasonable discretization, such that the minimum distance along grid lines from any node to the boundary is bounded, the solution $u^{\underline{\Omega}}$ is bounded.

Theorem 4.3. *Let $\underline{\Omega}$ and $\underline{\partial\Omega}$ be an orthogonal discretization such that for all $\underline{x} \in \underline{\Omega}$, $\tilde{\omega}(\underline{x}) \le \hat{W}$, for some constant $\hat{W}$. Then $\min_{\underline{x}' \in \underline{\partial\Omega}} g(\underline{x}') \le u^{\underline{\Omega}} \le \hat{U}$, for some constant $\hat{U}$.*

Proof. Let $\underline{x} \in \underline{\Omega}$. Let $\underline{x}' \in \underline{\partial\Omega}$ and $\underline{X} = (\underline{x}_1, \underline{x}_2, \dots \underline{x}_k)$ be such that $\underline{x}_1 = \underline{x}$, $\underline{x}_k = \underline{x}'$, and $\omega(\underline{X}) = \tilde{\omega}(\underline{x})$. Obtain a modified discretization defined by

$$\underline{\tilde{\Omega}} = \{\underline{x} \mid \underline{x} \in \underline{\Omega} \,\text{and}\, \underline{x} = \underline{x}_j, \text{ for some}\, j \text{ such that } 1 \le j \le k\} \text{ and}$$
$$\underline{\partial\tilde{\Omega}} = (\underline{\Omega} \cup \underline{\partial\Omega}) \setminus \underline{\tilde{\Omega}}.$$

Define the boundary condition $\tilde{g} : \underline{\partial\tilde{\Omega}} \to \mathbb{R}$ for the modified discretization

$$\tilde{g}(\underline{x}) = \begin{cases} g(\underline{x}), & \underline{x} \in \underline{\partial\Omega}, \\ \infty, & \text{otherwise.} \end{cases}$$

Run FMM on the modified grid and let $\underline{u}^{\underline{\tilde{\Omega}}}(\underline{x})$ be the value computed at $\underline{x}$. Note that $\underline{u}^{\underline{\tilde{\Omega}}}(\underline{x}) \le \hat{W}\hat{K} + g(\underline{x}')$, where $\hat{K}$ is given by (4.4) and g is the original boundary condition. Also note, by the monotonicity of $\underline{H}$ (Theorem 3.4), we have $\underline{u}^{\underline{\Omega}}(\underline{x}) \le \underline{u}^{\underline{\tilde{\Omega}}}(\underline{x})$, where $\underline{u}^{\underline{\Omega}}(\underline{x})$ is the value computed at $\underline{x}$ by FMM on the original grid defined by $\underline{\Omega}$ and $\underline{\partial\Omega}$. Therefore, $\underline{u}^{\underline{\Omega}} \le \hat{W}\hat{K} + \max_{\underline{x}' \in \underline{\partial\Omega}} g(\underline{x}')$. Since $u^{\underline{\Omega}}$ is defined by a non-expansive interpolation of the node values of $\underline{u}^{\underline{\Omega}}$,

$$u^{\underline{\Omega}} \le \hat{W}\hat{K} + \max_{\underline{x}' \in \underline{\partial\Omega}} g(\underline{x}').$$

For the lower bound, by Theorem 3.3, any solution $\mu = \tilde{\mu}$ to $\underline{H}(N(\underline{x}_0), \underline{u}, \mu) = 0$ must be such that $\tilde{\mu} > \breve{\underline{u}}$, where $\breve{\underline{u}}$ is the minimum neighbor value (3.5). By induction on the nodes of $\underline{\Omega}$, we find that for any node $\underline{x} \in \underline{\Omega} \cup \underline{\partial\Omega}$,

$$\underline{u}^{\underline{\Omega}}(\underline{x}) \ge \min_{\underline{x}' \in \underline{\partial\Omega}} g(\underline{x}').$$

Because of the non-expansive interpolation to form $u^{\underline{\Omega}}$, it follows that

$$u^{\underline{\Omega}} \ge \min_{\underline{x}' \in \underline{\partial\Omega}} g(\underline{x}').$$

□

4.3. Convergence. Let $\hat{h}(\underline{\Omega})$ be the maximum grid spacing between any two neighbors in $\underline{\Omega}$, i.e.,

$$\hat{h}(\underline{\Omega}) = \max_{\underline{x} \in \underline{\Omega},\ \underline{x}' \in \mathcal{N}(\underline{x})} \|\underline{x} - \underline{x}'\|.$$

We aim to show that $u^{\underline{\Omega}}$ converges to the viscosity solution of (1.1) as $\hat{h}(\underline{\Omega}) \to 0$. An upper (respectively, lower) semi-continuous function u is a viscosity subsolution (respectively, supersolution) of (1.1) if for all $\phi \in C_b^\infty(\Omega)$ such that $u - \phi$ has a local maximum (respectively, minimum) at x we have

$$H(x, D\phi(x)) \leq 0 \quad \text{(respectively, } H(x, D\phi(x)) \geq 0). \tag{4.5}$$

A continuous function u is a viscosity solution if it is both a viscosity subsolution and a viscosity supersolution.

Assume any orthogonal discretization of Ω and $\partial\Omega$ is reasonable, as required by Theorem 4.3. We then have bounded $u^{\underline{\Omega}}$ and may define $\hat{u}, \check{u} \in B(\Omega)$ by

$$\hat{u}(x) = \limsup_{\substack{x' \to x \\ \hat{h}(\underline{\Omega}) \to 0}} u^{\underline{\Omega}}(x') \quad \text{and} \quad \check{u}(x) = \liminf_{\substack{x' \to x \\ \hat{h}(\underline{\Omega}) \to 0}} u^{\underline{\Omega}}(x'). \tag{4.6}$$

Note that $\hat{u}$ is upper semi-continuous and $\check{u}$ is lower semi-conintuous. The following proofs closely follow the exposition of [3].

Theorem 4.4. *$\hat{u}$ is a viscosity subsolution of (1.1) and $\check{u}$ is a viscosity supersolution of (1.1).*

We only prove that $\hat{u}$ is a viscosity subsolution of (1.1), as the second part of the theorem can be proved symmetrically.

Proof. Let $\hat{x} \in \Omega$ be a local maximum of $\hat{u} - \phi$ for some $\phi \in C_b^\infty(\Omega)$. Without loss of generality, assume $\hat{x}$ is a strict local maximum [9, p. 542] and $(\hat{u} - \phi)(\hat{x}) = 0$. Then, there exist sequences $\underline{\Omega}_k$ and $\underline{x}_k \in \underline{\Omega}_k$ such that as $k \to \infty$,

$$\begin{gathered} \hat{h}(\underline{\Omega}_k) \to 0, \quad \underline{x}_k \to \hat{x}, \quad u^{\underline{\Omega}_k}(\underline{x}_k) \to \hat{u}(\hat{x}), \quad \textit{and} \\ (u^{\underline{\Omega}_k} - \phi)(\underline{x}_k) \geq (u^{\underline{\Omega}_k} - \phi)(\underline{x}_k') \text{ for all } \underline{x}_k' \in \mathcal{N}(\underline{x}_k). \end{gathered}$$

Let $\xi_k = (u^{\underline{\Omega}_k} - \phi)(\underline{x}_k)$. We have $\xi_k \to 0$ and $\phi(\underline{x}_k') + \xi_k \geq u^{\underline{\Omega}_k}(\underline{x}_k')$, for all $\underline{x}_k' \in \mathcal{N}(\underline{x}_k)$. Consequently, by the monotonicity of $\underline{H}$ (Theorem 3.4) and the definition of $u^{\underline{\Omega}_k}$ we have

$$\begin{aligned} \underline{H}(N(\underline{x}_k), \phi + \xi_k, \phi(\underline{x}_k) + \xi_k) &= \underline{H}(N(\underline{x}_k), \phi + \xi_k, u^{\underline{\Omega}_k}(\underline{x}_k)) \\ &\leq \underline{H}(N(\underline{x}_k), u^{\underline{\Omega}_k}, u^{\underline{\Omega}_k}(\underline{x}_k)) = 0 \end{aligned}$$

Take the limit as $k \to \infty$, and use the consistency of the $\underline{H}$ (Theorem 4.1) to get

$$\begin{aligned} 0 &\geq \lim_{k \to \infty} \underline{H}(N(\underline{x}_k), \phi + \xi_k, \phi(\underline{x}_k) + \xi_k) \\ &= \lim_{\substack{\underline{x}_0 \to x,\ \xi \to 0, \\ h_j^{s_j} \to 0,\ 1 \leq j \leq d}} \underline{H}(N(\underline{x}_0), \phi + \xi, \phi(\underline{x}_0) + \xi) = H(x, D\phi(x)) \end{aligned}$$

Therefore, $\hat{u}$ is a viscosity subsolution. ☐

Remark 1. *We note that if H is continuous, (1.1) satisfies a comparison principal [2]:* for any bounded upper semi-continuous u^* and bounded lower semi-continuous

u_*, which are a viscosity subsolution and supersolution, respectively, of (1.1), such that $u^* \leq u_*$ on $\partial\Omega$, we have $u^* \leq u_*$ in Ω.

THEOREM 4.5. *Let H be continuous. The function $\hat{u} = \check{u} = u$ is the unique viscosity solution of (1.1). As $\hat{h}(\underline{\Omega}) \to 0$, $u^{\underline{\Omega}}$ converges uniformly to u.*

Proof. By Theorem 4.4, $\hat{u}$ is an upper semi-continuous viscosity subsolution of (1.1) and $\check{u}$ is a lower semi-continuous viscosity supersolution of (1.1). It follows by the comparison principal that $\hat{u} \leq \check{u}$. But $\check{u} \leq \hat{u}$ by (4.6), so $u = \hat{u} = \check{u}$ is a viscosity solution of (1.1). Again, by the comparison principal, u must be the unique viscosity solution of (1.1). Therefore, by (4.6) $u^{\underline{\Omega}}$ converges uniformly to u, as $\hat{h}(\underline{\Omega}) \to 0$. □

5. Implementation.

In this section, we discuss efficient ways to implement the `Update` function, which calculates the unique solution, $\mu = \tilde{\mu}$, to (3.4). For practical grid sizes, solving (3.4) in `Update` remains the dominant computational cost, and for this reason we focus on the efficiency of `Update`.

Efficiency can be gained by determining which neighbors $\underline{x} \in \mathcal{N}(\underline{x}_0)$ have no influence on the solution and eliminating them from consideration. Let

$$\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_d),$$

where $\sigma_j \subseteq \{\pm 1\}$, indicate all $\underline{x} \in \mathcal{N}$ that are considered in determining the solution $\mu = \tilde{\mu}$. Let $\mathcal{N}_\sigma$ be the reduced set of neighbor nodes defined by σ. Let $\mathcal{S}_\sigma$ be the set of neighboring simplices that can be formed by the neighbors in $\mathcal{N}_\sigma$. For example, in $d = 4$ dimensions, take

$$\sigma = (\emptyset, \{\pm 1\}, \{-1\}, \{\pm 1\}).$$

We have

$$\begin{aligned} &\mathcal{N}_\sigma = \{\underline{x}_2^\pm, \underline{x}_3^-, \underline{x}_4^\pm\} \quad \text{and} \\ &\mathcal{S}_\sigma = \{(0,-1,-1,-1), (0,+1,-1,-1), (0,-1,-1,+1), (0,+1,-1,+1)\}. \end{aligned}$$

Let

$$\underline{H}_\sigma(N, \phi, \mu) = \max_{s \in \mathcal{S}_\sigma} [G(\underline{x}_0, \underline{D}^s(N, \phi, \mu))] - c(\underline{x}_0), \tag{5.1}$$

be the *reduced-neighbor numerical Hamiltonian*, a modification of (3.1) that only considers the neighbors and simplices indicated by σ. For $s \in \mathcal{S}_\sigma$ and $1 \leq j \leq d$, $s_j = 0$ indicates that $\underline{x}_j^{s_j}$ is not considered in computing the gradient approximation $D^s \underline{u}(\mu)$:

$$\underline{D}_j^s(N, \phi, \mu) = \begin{cases} \frac{\max(0, \mu - \phi(\underline{x}_j^{s_j}))}{-h_j^{s_j}}, & s_j = \pm 1, \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

To implement `Update` we first reduce the set of considered neighbors and then solve

$$\underline{H}_\sigma(N(\underline{x}_0), \underline{u}, \mu) = 0 \tag{5.3}$$

for $\mu = \tilde{\mu}$ to determine a node's value $\underline{u}(\underline{x}_0)$. As in Section 3, we may write $\underline{H}_\sigma(\mu) = \underline{H}_\sigma(N, \phi, \mu)$ and $\underline{D}^s(\mu) = \underline{D}^s(N, \phi, \mu)$, where no ambiguity results. Note that some properties of (5.3) are retained from (3.4) as long as at least one considered neighbor remains in σ. Let

$$\check{\underline{u}}_\sigma = \min_{\underline{x} \in \mathcal{N}_\sigma} \left(\underline{u}(\underline{x}) \right).$$

Proposition 5.1. *(analogue of Lemma 3.2) The reduced-neighbor numerical Hamiltonian, $\underline{H}_\sigma(\mu)$, is strictly increasing on $\mu \geq \check{\underline{u}}_\sigma$. Furthermore, $\underline{H}_\sigma(\mu) = -c$ for $\mu \leq \check{\underline{u}}_\sigma$. Finally, $\underline{H}_\sigma(\mu)$ is nondecreasing on all μ.*

Proposition 5.2. *(analogue of Theorem 3.3) There exists a unique solution $\mu = \tilde{\mu}$ to $\underline{H}_\sigma(\mu) = 0$ such that $\tilde{\mu} > \check{\underline{u}}_\sigma$.*

Algorithm 2 implements the `Update` function with a number of binary parameters: doFullNum, doSymElim, doBinSearch, doSolElim, and doAnalytic. These parameters are used to create variants of `Update` by enabling or disabling features, which are described in Table 5.1. Section 6.1 examines the effect of these features on algorithm efficiency by comparing operation counts. The rest of this section describes the subroutines `SymmetryElim`, `NoncausalElim`, and `FindSolution` used in `Update`. `FarNeighborElim` eliminates neighbors from σ which have $\underline{u}(\underline{x}) = \infty$ because they have not yet been updated or which do not exist because they are beyond the edge of the grid. `NumericalSolve` uses a numerical root finder, such as the interval method [11, p. 224] or secant method [11, p. 231], to determine the solution to an equation. All results reported in this paper use the secant method, which is generally faster than the interval method but less robust.

```
Input: doFullNum, doSymElim, doBinSearch, doSolElim, doAnalytic
Define: $\sigma = (\{\pm 1\}, \{\pm 1\}, \ldots, \{\pm 1\})$
$\sigma$ = FarNeighborElim($\sigma$)
if doFullNum then
    $\mu_\sigma$ = NumericalSolve($\underline{H}_\sigma(\mu) = 0, \mu$)
else
    if doSymElim then $\sigma$ = SymmetryElim($\sigma$)
    $\sigma$ = NoncausalElim(doBinSearch, $\sigma$)
    $\mu_\sigma$ = FindSolution(doSolElim, doAnalytic, $\sigma$)
end
Output: $\mu_\sigma$
```

Algorithm 2: `Update`

5.1. Symmetry. We show that the considered neighbors, σ, can be reduced by keeping only the neighbor with smaller value of a pair of opposite neighbors in the jth dimension when (3.1) is symmetric in that dimension. First, we introduce notation and prove a useful lemma.

Let $q \in \mathbb{R}^d$. Let $T^i(q)$ be a reflection of q in the hyperplane orthogonal to the ith axis, such that

$$T^i_j(q) = \begin{cases} -q_j, & \text{if } j = i, \\ q_j, & \text{otherwise,} \end{cases}$$

for $1 \leq j \leq d$. Let Ψ_j indicate symmetry of (3.1) in the jth dimension, as follows:

$$\Psi_j = \begin{cases} 1, & \text{if } |h_j^-| = |h_j^+| \text{ and } \forall q \in \mathbb{R}^d, G(q) = G(T^j(q)), \\ 0, & \text{otherwise.} \end{cases}$$

parameter	true / false
doFullNum	solve (5.1) directly using root finder / solve (5.1) simplex by simplex (see Section 5.3)
doSymElim	call `SymmetryElim` to eliminate neighbors based on symmetry / do not call `SymmetryElim`
doBinSearch	use a binary search to eliminate neighbors based on causality / use a linear search to eliminate neighbors based on causality
doSolElim	eliminate solutions from some simplices / consider solutions from all simplices
doAnalytic	compute solution from simplex analytically / compute solution from simplex using root finder

Table 5.1
Binary parameters of `Update`*.*

In other words, $\Psi_j = 1$ if and only if the grid spacing and G are symmetric in the jth dimension.

Lemma 5.3. *Let j be such that $1 \leq j \leq d$. Let $s \in \mathcal{S}$. Let $s' = T^j(s)$. If $\Psi_j = 1$ and $\underline{u}_j^{s'_j} \leq \underline{u}_j^{s_j}$, then $G(D^{s'}\underline{u}(\mu)) \geq G(D^s\underline{u}(\mu))$, for all μ.*

Proof. Let j, s, and s' be as defined above. Let $\Psi_j = 1$ and $\underline{u}_j^{s'_j} \leq \underline{u}_j^{s_j}$. Consider the components of $T^j(D^{s'}\underline{u}(\mu))$. We have

$$\begin{aligned}
&T_j^j(D^{s'}\underline{u}(\mu))D_j^s\underline{u}(\mu) \\
&= -D_j^{s'}\underline{u}(\mu)D_j^s\underline{u}(\mu) \\
&= -\frac{\max(0, \mu - \underline{u}_j^{s'_j})}{-h_j^{s'_j}}\frac{\max(0, \mu - \underline{u}_j^{s_j})}{-h_j^{s_j}} \geq 0,
\end{aligned}$$

since $h_j^{s'_j}h_j^{s_j} < 0$. Furthermore,

$$\begin{aligned}
|T_j^j(D^{s'}\underline{u}(\mu))| &= |-D^{s'_j\underline{u}}(\mu)| \\
&= \left|-\frac{\max(0, \mu - \underline{u}_j^{s'_j})}{-h_j^{s'_j}}\right| \\
&\geq \left|\frac{\max(0, \mu - \underline{u}_j^{s_j})}{-h_j^{s_j}}\right| = |D_j^s\underline{u}(\mu)|,
\end{aligned}$$

since $h_j^{s'_j} = -h_j^{s_j}$ and $\underline{u}_j^{s'_j} \leq \underline{u}_j^{s_j}$. For $i \neq j$,

$$T_i^j(D^{s'}\underline{u}(\mu)) = D_i^{s'}\underline{u}(\mu) = D_i^s\underline{u}(\mu),$$

since $s'_i = s_i$. Consequently, $T^j(D^{s'}\underline{u}(\mu)) \trianglerighteq D_i^s\underline{u}(\mu)$.

Therefore, by the symmetry of G in the jth dimension and by the one-sided monotonicity of G,

$$G(D^{s'}\underline{u}(\mu)) = G(T^j(D^{s'}\underline{u}(\mu))) \geq G(D^s\underline{u}(\mu)).$$

□

Theorem 5.4. *Let σ be such that $\sigma_j \subseteq \{\pm 1\}$, for $1 \le j \le d$. Let $\tilde{\sigma}$ be defined by*

$$\tilde{\sigma}_j = \begin{cases} \{-1\}, & \text{if } \sigma_j = \{\pm 1\},\ \Psi_j = 1 \text{ and } \underline{u}_j^- \le \underline{u}_j^+, \\ \{+1\}, & \text{if } \sigma_j = \{\pm 1\},\ \Psi_j = 1 \text{ and } \underline{u}_j^- > \underline{u}_j^+, \\ \sigma_j, & \text{otherwise}, \end{cases}$$

for $1 \le j \le d$. Let $\mu = \mu_\sigma$ be the unique solution to $\underline{H}_\sigma(\mu) = 0$. Let $\mu = \mu_{\tilde{\sigma}}$ be the unique solution to $\underline{H}_{\tilde{\sigma}}(\mu) = 0$. Then $\mu_{\tilde{\sigma}} = \mu_\sigma$.

Proof. Let σ, $\tilde{\sigma}$, μ_σ and $\mu_{\tilde{\sigma}}$ be as defined above. Consider a sequence $\sigma^0, \sigma^1, \ldots, \sigma^d$, such that

$$\sigma_j^i = \begin{cases} \tilde{\sigma}_j^i, & \text{if } i \ge j, \\ \sigma_j^i, & \text{otherwise}, \end{cases}$$

for $1 \le j \le d$. Let $\mu = \mu_{\sigma^i}$ be the unique solution to $\underline{H}_{\sigma^i}(\mu) = 0$.

Let i be such that $1 \le i \le d$. Assume $\mu_{\sigma^{i-1}} = \mu_\sigma$. For each $s \in \mathcal{S}_{\sigma^{i-1}}$, if $s \notin \mathcal{S}_{\sigma^i}$, then $s' = T^i(s) \in \mathcal{S}_{\sigma^i}$, $\Psi_i = 1$, and $\underline{u}_i^{s'_i} \le \underline{u}_i^{s_i}$. So, by Lemma 5.3, $G(D^{s'}\underline{u}(\mu)) \ge G(D^s\underline{u}(\mu))$, for all μ. In particular, $G(D^{s'}\underline{u}(\mu_{\sigma^{i-1}})) \ge G(D^s\underline{u}(\mu_{\sigma^{i-1}}))$. As a result, for each $s \in \mathcal{S}_{\sigma^{i-1}}$, there exists $\tilde{s} \in \mathcal{S}_{\sigma^i}$, such that $G(D^{\tilde{s}}\underline{u}(\mu_{\sigma^{i-1}})) \ge G(D^s\underline{u}(\mu_{\sigma^{i-1}}))$. Also, note that $\mathcal{S}_{\sigma^i} \subseteq \mathcal{S}_{\sigma^{i-1}}$. Consequently,

$$\begin{aligned} \underline{H}_{\sigma^i}(\mu_{\sigma^{i-1}}) &= \max_{s \in \mathcal{S}_{\sigma^i}} [G(D^s\underline{u}(\mu_{\sigma^{i-1}}))] - c \\ &= \max_{s \in \mathcal{S}_{\sigma^{i-1}}} [G(D^s\underline{u}(\mu_{\sigma^{i-1}}))] - c = \underline{H}_{\sigma^i}(\mu_{\sigma^{i-1}}) = 0, \end{aligned}$$

Accordingly $\mu = \mu_{\sigma^{i-1}}$ is the unique solution to $\underline{H}_{\sigma^i}(\mu) = 0$. By the definition of μ_{σ^i} and the assumption, $\mu_{\sigma^i} = \mu_{\sigma^{i-1}} = \mu_\sigma$. Therefore, by induction on i, $\mu_{\tilde{\sigma}} = \mu_\sigma$, since $\sigma^0 = \sigma$ and $\sigma^d = \tilde{\sigma}$. □

Algorithm 3 uses the result obtained in Theorem 5.4 to eliminate $\underline{x} \in \mathcal{N}$ from consideration in solving (5.1) by exploiting symmetries in (3.1).

```
Input: $\sigma$
for $j \leftarrow 1 : d$ do
    if $\sigma_j = \{\pm 1\}$ and $\Psi_j = 1$ then
        if $\underline{u}_j^- \le \underline{u}_j^+$ then
            $\sigma_j \leftarrow \{-1\}$
        else $\sigma_j \leftarrow \{+1\}$
    else $\sigma_j \leftarrow \{\pm 1\}$
end
Output: $\sigma$
```

Algorithm 3: `SymmetryElim`

5.2. Causality. The causality of (3.1) can also be exploited to eliminate $\underline{x} \in \mathcal{N}_\sigma$ from consideration. We show with the following theorem that the condition $\underline{H}_\sigma(\underline{u}(\underline{x})) \ge 0$ can be checked to determine that a node $\underline{x}$ is non-causal, i.e., that the solution $\mu = \mu_\sigma$ to (5.1) is not dependent on the node $\underline{x}$ and its value $\underline{u}(\underline{x})$.

THEOREM 5.5. *Let σ be such that $\sigma_j \subseteq \{\pm 1\}$, for $1 \leq j \leq d$. Pick any $s \in \mathcal{S}_\sigma$ and $i \in \{1, 2, \dots, d\}$, such that $s_i \neq 0$ and $\underline{H}_\sigma(\underline{u}_i^{s_i}) \geq 0$. Let $\tilde{\sigma}$ be defined by*

$$\tilde{\sigma}_j = \begin{cases} \sigma_j \setminus \{s_j\}, & \textit{if } j = i, \\ \sigma_j, & \textit{otherwise,} \end{cases}$$

Let $\mu = \mu_\sigma$ be the unique solution to $\underline{H}_\sigma(\mu) = 0$. Let $\mu = \mu_{\tilde{\sigma}}$ be the unique solution to $\underline{H}_{\tilde{\sigma}}(\mu) = 0$. Then $\mu_{\tilde{\sigma}} = \mu_\sigma$.

Proof. Let σ, s, i, $\tilde{\sigma}$, μ_σ and $\mu_{\tilde{\sigma}}$ be as defined above. By Proposition 5.1 $\underline{H}_\sigma(\mu)$ is strictly increasing on $\mu \geq \underline{\check{u}}_\sigma$. Since $\underline{H}_\sigma(\underline{u}_i^{s_i}) \geq 0$ it must be that $\mu_\sigma \leq \underline{u}_i^{s_i}$. Note $\underline{H}_{\tilde{\sigma}}(\mu)$ is identical to $\underline{H}_\sigma(\mu)$ except for $D_i^s \underline{u}(\mu)$ which is set to zero in $\underline{H}_{\tilde{\sigma}}(\mu)$. But for $\mu \leq \underline{u}_i^{s_i}$, we also have $D_i^s \underline{u}(\mu) = 0$ in $\underline{H}_\sigma(\mu)$. Consequently, $\underline{H}_{\tilde{\sigma}}(\mu) = \underline{H}_\sigma(\mu)$ for $\mu \leq \underline{u}_i^{s_i}$. In particular, $\underline{H}_{\tilde{\sigma}}(\mu_\sigma) = \underline{H}_\sigma(\mu_\sigma) = 0$. Therefore, $\mu_{\tilde{\sigma}} = \mu_\sigma$. □

Theorem 5.5 states that the solution μ to (5.1) does not change when a non-causal node is removed from σ. This node removal can be repeated until all non-causal nodes have been removed and the solution $\mu = \mu_\sigma$ will remain unchanged. Algorithm 4 uses this result to remove all noncausal nodes, using a binary or linear search to determine the largest node value that might be causal. Because Theorem 5.5 does not place any restrictions on σ, Algorithm 4 can be used after Algorithm 3 to further eliminate nodes whose values cannot possibly influence the solution $\mu = \mu_\sigma$. Note that causal elimination does not require symmetry in (3.1). However, the test for non-causality requires an evaluation of $\underline{H}_\sigma$, which is more expensive than the comparison of two neighbors' values used for symmetry elimination.

```
Input: doBinSearch, $\sigma$
$w \leftarrow$ Sort($\mathcal{N}_\sigma, \underline{u}$)
$\check{l} \leftarrow 1$
$\hat{l} \leftarrow$ Length($w$)
if doBinSearch then
    while $\check{l} < \hat{l}$ do
        $k \leftarrow \lceil (\check{l} + \hat{l})/2 \rceil$
        if $\underline{H}_\sigma(w_k) < 0$ then $\check{l} \leftarrow k$ else $\hat{l} \leftarrow k$
    end
else /*do backwards linear search*/
    while $\underline{H}_\sigma(w_{\hat{l}}) \geq 0$ do
        $\hat{l} \leftarrow \hat{l} - 1$
    end
for $j \leftarrow 1 : d$ do /*remove non-causal nodes from $\sigma$*/
    if $-1 \in \sigma_j$ and $u_j^- \geq w_{\hat{l}}$ then $\sigma_j \leftarrow \sigma_j \setminus \{-1\}$
    if $+1 \in \sigma_j$ and $u_j^+ \geq w_{\hat{l}}$ then $\sigma_j \leftarrow \sigma_j \setminus \{+1\}$
end
Output: $\sigma$
```

Algorithm 4: NoncausalElim

The Sort routine in Algorithm 4 sorts, according to $\underline{u}$, the considered nodes, $\underline{x} \in \mathcal{N}_\sigma$. It returns a vector w containing sorted $\underline{u}(\underline{x})$, such that $w_1 \leq w_2 \leq \cdots \leq w_m$, where $m = |\mathcal{N}_\sigma|$. The first **while** loop performs a binary search for the largest index $\check{l} \leq m$, such that the solution $\tilde{\mu}$ may depend on $w_{\check{l}}$. Because of the ordering of w, $w_{\check{l}}$

is the value of the largest causal node. The **for** loop removes from σ any non-causal nodes.

5.3. Solution. After eliminating nodes from σ using Algorithms 3 and 4, we can determine the solution $\check{\mu}$ to (5.1). Let

$$\check{\mu} = \min_{s \in \mathcal{S}_\sigma}(\mu_s), \tag{5.4}$$

where $\mu = \mu_s$ is the solution to

$$G(D^s\underline{u}(\mu)) - c = 0. \tag{5.5}$$

We show with the following proposition that, instead of solving (5.1) directly, we can solve (5.5) for each $s \in \mathcal{S}_\sigma$ and take the minimum such solution $\check{\mu}$. It can be shown that $G(D^s\underline{u}(\mu))$ is continuous and nondecreasing on μ and that (5.5) has a unique solution in an analogous but simpler manner to the proof of Theorem 3.3.

PROPOSITION 5.6. *Let $\hat{\mu}$ be the solution to (5.1). Then $\hat{\mu} = \check{\mu}$.*

Proof. Let μ_s, $\check{\mu}$ and $\hat{\mu}$ be as defined above. For any $s \in \mathcal{S}_\sigma$, we know $\mu_s \geq \check{\mu}$. Since $G(D^s\underline{u}(\mu))$ is nondecreasing on μ, it must be that $G(D^s\underline{u}(\mu)) \leq G(D^s\underline{u}(\mu_s)) = c$, for all $\mu \leq \mu_s$. In particular, $G(D^s\underline{u}(\check{\mu})) \leq c$. Furthermore, by the definition of $\check{\mu}$, there exists an $\check{s} \in \mathcal{S}_\sigma$, such that $G(D^{\check{s}}\underline{u}(\check{\mu})) = c$. Consequently,

$$\underline{H}_\sigma(\check{\mu}) = \max_{s \in \mathcal{S}_\sigma} G(D^s\underline{u}(\check{\mu})) - c = 0. \tag{5.6}$$

Therefore, $\hat{\mu} = \check{\mu}$ is the solution to (5.1). □

We further show that we may be able to determine $\check{\mu}$ without solving (5.5) for each $s \in \mathcal{S}_\sigma$. We demonstrate using the following proposition that if we have computed a solution $\mu = \mu_s$ of (5.5) for some $s \in \mathcal{S}_\sigma$, we can easily determine if $\mu_{\tilde{s}} \geq \mu_s$, where $\mu = \mu_{\tilde{s}}$ is the solution to $G(D^{\tilde{s}}\underline{u}(\mu)) - c = 0$. Note we do not necessarily need to compute $\mu_{\tilde{s}}$ to rule it out as a minimal solution.

PROPOSITION 5.7. *Let $s \in \mathcal{S}_\sigma$ and $\tilde{s} \in \mathcal{S}_\sigma$. Let $\mu = \mu_s$ be the solution to $G(D^s\underline{u}(\mu)) - c = 0$ and $\mu = \mu_{\tilde{s}}$ be the solution to $G(D^{\tilde{s}}\underline{u}(\mu)) - c = 0$. Then $\mu_{\tilde{s}} < \mu_s$ if and only if $G(D^{\tilde{s}}\underline{u}(\mu_s)) > G(D^s\underline{u}(\mu_s))$.*

Proof. Let μ_s and $\mu_{\tilde{s}}$ be as defined above. If $G(D^{\tilde{s}}\underline{u}(\mu_s)) > G(D^s\underline{u}(\mu_s)) = c$, then the unique solution $\mu = \mu_{\tilde{s}}$ to $G(D^{\tilde{s}}\underline{u}(\mu)) = c$ must be such that $\mu_{\tilde{s}} < \mu_s$, since $G(D^{\tilde{s}}\underline{u}(\mu))$ is nondecreasing on μ. Similarily, if $G(D^{\tilde{s}}\underline{u}(\mu_s)) \leq G(D^s\underline{u}(\mu_s))$ then the unique solution $\mu = \mu_{\tilde{s}}$ to $G(D^{\tilde{s}}\underline{u}(\mu)) = c$ must be such that $\mu_{\tilde{s}} \geq \mu_s$. □

Algorithm 5 determines $\check{\mu}$, the minimal solution to (5.5). If $\mathsf{doSolElim}$ is true, it exploits the result of Proposition 5.7 in the inner **foreach** loop to eliminate simplices $s \in \mathcal{S}_\sigma$, for which solutions to (5.5) are irrelevant to the computation. We call this process *solution elimination*. At each iteration of the **while** loop we must solve (5.5) with s being a particular simplex. For the first iteration, we heuristically choose $\arg\max_{s \in \mathcal{S}_\sigma} G(D^s\underline{u}(\hat{u}))$ as the simplex s for which the solution to (5.5) is computed. This choice of simplex s makes it likely that the corresponding solution μ_s is the desired minimal solution from (5.4). Similarily, for subsequent iterations we find $\arg\max_{\tilde{s} \in \mathcal{S}_\sigma} G(D^{\tilde{s}}\underline{u}(\check{\mu}))$ as the simplex s for which the solution to (5.5) is computed.

If $\mathsf{doSolElim}$ is false, we iterate through all $s \in \mathcal{S}_\sigma$ to determine $\check{\mu}$ from (5.4). The function $\texttt{Solve}(\mathsf{doAnalytic}, G(D^s\underline{u}(\mu)) - c = 0, \mu)$ computes the solution to (5.5) either analytically or numerically. Analytic solutions for common G are derived in Appendix B. In other cases, it may be necessary to use a root-finding algorithm such as the interval or secant method.

```
Input: doSolElim, doAnalytic, $\sigma$
if doSolElim then
    $\hat{u} = \max_{\underline{x} \in \mathcal{N}_\sigma} \underline{u}(\underline{x})$
    $s = \arg\max_{s \in \mathcal{S}_\sigma} G(D^s \underline{u}(\hat{u}))$ ;   /* guess minimal solution simplex */
    while $\mathcal{S}_\sigma \neq \emptyset$ do
        $\check{\mu} = \texttt{Solve}(\text{doAnalytic}, G(D^s \underline{u}(\mu)) - c = 0, \mu)$
        $\mathcal{S}_\sigma \leftarrow \mathcal{S}_\sigma \setminus \{s\}$
        foreach $\tilde{s} \in \mathcal{S}_\sigma$ do
            $\theta = -\infty$
            if $G(D^{\tilde{s}} \underline{u}(\check{\mu})) \leq 0$ then
                $\mathcal{S}_\sigma \leftarrow \mathcal{S}_\sigma \setminus \{\tilde{s}\}$ ;          /* solution cannot come from $\tilde{s}$ */
            else if $G(D^{\tilde{s}} \underline{u}(\check{\mu})) > \theta$ then
                $\theta \leftarrow G(D^{\tilde{s}} \underline{u}(\check{\mu}))$
                $s \leftarrow \tilde{s}$
            end
        end
    end
else /* solve without eliminating solutions from simplices */
    $\check{\mu} = \infty$
    foreach $s \in \mathcal{S}_\sigma$ do
        $\mu_s = \texttt{Solve}(\text{doAnalytic}, G(D^s \underline{u}(\mu)) - c = 0, \mu)$
        if $\mu_s < \check{\mu}$ then $\check{\mu} \leftarrow \mu_s$
    end
end
Output: $\check{\mu}$
```

Algorithm 5: `FindSolution`

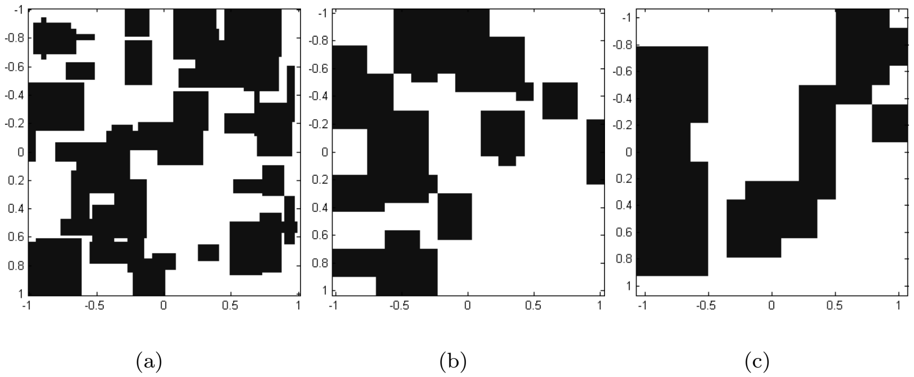
5.4. Discussion. Two different but equivalent algorithms for computing the analytic solution from a single simplex are described in [25, 14]. However, they both assume the isotropic Eikonal equation and equal spacing of the grid in all dimensions, in which case one can identify a single simplex to consider using symmetry alone. The symmetry elimination in Algorithm 3 is the straightforward generalization of the procedure in [14] to all axis-aligned anisotropic problems on unequally spaced grids. When symmetry does not by itself reduce the update to considering a single simplex, the causality and solution elimination procedures in Algorithms 4 and 5 can be applied in hopes of making `Update` more efficient. Causality elimination in Algorithm 4 is derived from the causality condition required for an analytic solution from a single simplex [14].

6. Experiments.

We conduct experiments to determine how to implement the `Update` function efficiently, show numerical evidence that the result of FMM converges to the viscosity solution of (1.1), and demonstrate other types of problems that can be solved.

Throughout this section, the boundary conditions are $g(x) = 0$ for $x \in \partial\Omega$. For all experiments below, except that in Section 6.4, we discretize $[-1, 1]^d$ as follows. We let

$$\underline{x}_{i_1, i_2, \ldots, i_d} = (-1 + (i_1 - 1) * h, -1 + (i_2 - 1) * h, \ldots, -1 + (i_d - 1) * h),$$



(a) (b) (c)

Fig. 6.1. *Inhomogeneous cost function. In the white regions* $c(x) = 1$ *and in the black regions* $c(x) = 5$*. The black regions were created by taking the union of a number of randomly-generated hyper-rectangular regions in the discretized space. (a)* $d = 2$*. (b)* $d = 3$*: slice at* $x_3 = 0$*. (c)* $d = 3$*: slice at* $x_3 = x_4 = 0$*.*

where $h = 2/(m-1)$, m is the number of nodes in each dimension and $i_j = 1, \ldots, m$, for $1 \leq j \leq d$. We let $m = 2 * l + 1$ for a positive integer l, so that $\underline{x}_{(m-1)/2,(m-1)/2,\ldots,(m-1)/2} = O$, where O is the origin.

6.1. Efficiency of Implementation. We compare variants of the `Update` function created by different combinations of the parameters doFullNum, doSymElim, doBinSearch, doSolElim, and doAnalytic. We examine the number of *Operations* for several test problems. *Operations* is defined to be the total number of G function evaluations used to compute $\underline{u}$ over all $\underline{\Omega}$. In the case where an analytic formula is used to solve (5.5), *Operations* is the sum of G function evaluations and analytic solution computations. We are assuming that computing a solution analytically is about as expensive as a G function evaluation.

All the test problems are standard Eikonal equations, i.e., they involve H as in (2.1) with $G(Du(x)) = \|Du(x)\|_2$. A test problem may have $\Omega = [-1,1]^d \setminus \{O\}$ and $\partial\Omega = \{O\}$ or $\Omega = \{x \mid \|x\|_2 < 0.9\}$ and $\partial\Omega = \{x \mid \|x\|_2 = 0.9\}$. In the former case, the FMM computation propagates outward from the center of the domain, and in the latter it propagates inwards from the hyperspherical boundary. Note that, for the latter case, we do not discretize $\partial\Omega$ accurately using a non-uniform grid as shown in Figure 1.2. Instead, we use the uniform grid defined above and any node $\underline{x}$, such that $\|\underline{x}\|_2 \geq 0.9$, is added to $\underline{\partial\Omega}$. Also, a test problem may have a homogeneous cost function ($c(x) = 1$) or a binary-valued inhomogeneous cost function (see Figure 6.1). These two choices result in 4 problem types for each of 2, 3, and 4 dimensions. Problem types a and b (resp., c and d) have the boundary condition in the center (resp., around the outside) of $\underline{\Omega}$. Problem types a and c (resp., b and d) have a homogeneous (resp., inhomogeneous) cost function.

In Table 6.1, we compare 6 different variants of `Update` on these 12 test problems. Variant 1, as a naive numerical solution, performs the worst for all test problems. Variants 3 through 6 each add a single feature to Variant 2. By comparing Variant 3 to Variant 2, we see that Algorithm 3 is always beneficial, especially for problem types b, c, and d with dimensions 3+. This is not suprising since symmetry elimination is a cheap operation, which can considerably reduce the considered neighbors σ. Sym-

d	m	n	type	Operations ($\times 10^3$)					
2	101	10201	a	153	147	139	147	147	49
			b	182	167	139	168	157	57
			c	104	98	88	98	98	34
			d	119	110	87	111	101	39
3	31	29791	a	889	700	612	715	700	262
			b	1530	1070	617	1030	894	458
			c	514	354	232	343	353	158
			d	736	497	237	471	391	224
4	15	50625	a	2010	1550	1350	1590	1550	597
			b	6840	3760	1380	3240	2890	1950
			c	1310	741	278	642	733	395
			d	2200	1190	286	1010	840	637
Variant				1	2	3	4	5	6
doFullNum				x					
doSymElim						x			
doBinSearch							x		
doSolElim								x	
doAnalytic									x

TABLE 6.1

Operation counts for variants of the update function, where H is as in (2.1) and $G(Du(x)) = \|Du(x)\|_2$. The parameters d, m, and n are the dimension, the number of nodes in each dimension, and the total number of nodes, respectively. The type column lists one of four problem types defined in Section 6.1. Operations *counts the total number of operations as defined in Section 6.1. The Variant parameters are defined in Section 5. 'x' indicates that a parameter is true for the relevant Variant.*

metry elimination should be used wherever possible. Note that if Algorithm 3 is used to reduce $\mathcal{S}_\sigma$ to a single simplex, then noncausal elimination and solution elimination become superfluous.

Comparing Variant 4 to Variant 2, we find the binary search in Algorithm 4 is only beneficial for problem types b, c, and d with dimension 3+. For low dimensions the binary search is overkill and may actually hurt. Comparing Variant 5 to Variant 2, we see that solution elimination in Algorithm 5 is beneficial for problem types b and d for all dimensions tested. Because of inhomongeneous cost, the solutions to problem types b and d have discontinuous derivatives where characteristics intersect. This results in competing solutions to (5.5) from more than one simplex. Solution elimination can be used to efficiently determine the smallest of these solutions.

The benefit in all test problems of using an analytic formula rather than the secant method to solve (5.5) can be seen by comparing Variant 6 to Variant 2. Although not shown in the table, the benefit of solution elimination is mostly lost if an efficient analytic formula is used. This is because computing the analytic solution to (5.5) is typically not much more expensive than the evalution of G that is required to eliminate the solution in Algorithm 5.

REMARK 2. *Note that a substantial problem with the efficiency of FMM is not made apparent in Table 6.1 as it is not the focus of this paper. For the 12 problems tested, between 48.5 and 73.2% of the times* `Update` *was called in Algorithm 1 the new value $\underline{u}(\underline{x})$ calculated by* `Update` *was identical to the previous value. In many of these cases, $\underline{u}(\underline{x})$ does not change because no neighbors values have changed since the* *previous* `Update` *call. This problem can be fixed by keeping a count of* `Update` *calls, storing for each node $\underline{x}$ the count of its last* `Update` *call, and only updating $\underline{u}(\underline{x})$ if at least one of its neighbor's values has been updated since the last update to $\underline{u}(\underline{x})$.*

6.2. Convergence Study. We examine the difference between the solution to 3.3 and the solution to 1.1 for two simple Dirichlet problems. In particular, we look at how the absolute error changes as the grid spacing decreases toward zero. Let $u : \Omega \to \mathbb{R}$ be the exact solution to (1.1) for some axis-aligned problem, i.e., for some H and g, where H is as defined in (2.1). Let $\underline{u} : \underline{\Omega} \to \mathbb{R}$ be the solution by FMM to the dicretized problem.

The $\mathcal{L}_\infty$-error (maximum error) is given by

$$e_\infty = \max_{\underline{x} \in \underline{\Omega}} |u(\underline{x}) - \underline{u}(\underline{x})|.$$

The $\mathcal{L}_1$-error (average error) is approximated using

$$e_1 = \frac{\sum_{\underline{x} \in \underline{\Omega}} |u(\underline{x}) - \underline{u}(\underline{x})|}{|\underline{\Omega}|}.$$

This is a reasonable approximation for the $\mathcal{L}_1$-error, since we use a uniform grid spacing for these experiments. We calculate the convergence rate, measured in the $\mathcal{L}_\infty$ norm, using

$$\frac{e_\infty(h)}{e_\infty(\tilde{h})} = \left(\frac{h}{\tilde{h}}\right)^{r_\infty},$$

where $e_\infty(h)$ is the $\mathcal{L}_\infty$-error using a uniform grid spacing h and r_∞ is the convergence rate. This can be solved for r_∞ to obtain

$$r_\infty = \frac{\log e_\infty(h) - \log e_\infty(\tilde{h})}{\log h - \log \tilde{h}}.$$

The $\mathcal{L}_1$ convergence rate, r_1 is calculated in the analogous manner.

For the problems considered, $\Omega = [-1, 1]^d \setminus \{O\}$. We take $G(Du(x)) = \|Du(x)\|_p$, where $p = 1$ or $p = 2$. The boundary conditions are $g(O) = 0$. Since there is a node at O, any error introduced is from the discretization of H and not from the discretization of the boundary condition. The approximation errors are summarized in Table 6.2. Figure 6.2 plots these errors against the grid spacing h, for $d = 2$.

6.3. Asymmetric Anisotropic Problem. For this anisotropic problem, H is as in (2.1), where G is defined by (2.6) (see Figure 2.3(b)). The domain is given by $\Omega = [-1, 1]^d \setminus \{O\}$ and $\partial\Omega = \{O\}$. The cost is $c(x) = 1$, except in four rectangular regions shown in black in Figure 6.3 where $c(x) \gg 1$. The number of nodes in each dimension is $m = 1281$. We plot the contours of $\underline{u}$ computed by FMM in Figure 6.3. Note the asymmetric contours where the characteristics bend through gaps. The relationship between the shape of the contours of G in Figure 2.3(b) and those of $\underline{u}$ is explained by the duality articulated in Proposition 2.7.

6.4. Two Robots. We consider the two-robot coordinated navigation problem illustrated in Figure 6.4. The circular robots are free to move independently in a 2-dimensional plane but may not collide with each other or the obstacles (black region). Each may travel at a maximum speed of $1/c(x)$ in any direction. The robots attempt

				$p=1$				$p=2$			
d	m	n	h	e_∞	r_∞	e_1	r_1	e_∞	r_∞	e_1	r_1
2	11	1.2e2	2.0e-1	2.2e-1		6.3e-2		1.2e-1		6.2e-2	
	21	4.4e2	1.0e-1	1.7e-1	.41	3.7e-2	.77	7.8e-2	.56	4.3e-2	.53
	41	1.7e3	5.0e-2	1.2e-1	.46	2.0e-2	.85	5.0e-2	.65	2.8e-2	.63
	81	6.6e3	2.5e-2	8.8e-2	.48	1.1e-2	.90	3.1e-2	.70	1.7e-2	.69
	161	2.6e4	1.3e-2	6.3e-2	.49	5.7e-3	.94	1.8e-2	.75	1.0e-2	.73
	321	1.0e5	6.3e-3	4.4e-2	.49	2.9e-3	.96	1.1e-2	.78	6.1e-3	.77
	641	4.1e5	3.1e-3	3.1e-2	.50	1.5e-3	.97	6.1e-3	.81	3.5e-3	.79
	1281	1.6e6	1.6e-3	2.2e-2	.50	7.6e-4	.98	3.4e-3	.83	2.0e-3	.82
3	11	1.3e3	2.0e-1	3.5e-1		1.2e-1		2.1e-1		1.2e-1	
	21	9.3e3	1.0e-1	2.6e-1	.43	6.9e-2	.78	1.4e-1	.58	8.4e-2	.57
	41	6.9e4	5.0e-2	1.9e-1	.47	3.9e-2	.85	8.7e-2	.66	5.4e-2	.65
	81	5.3e5	2.5e-2	1.3e-1	.49	2.1e-2	.89	5.3e-2	.72	3.3e-2	.70
	161	4.2e6	1.3e-2	9.5e-2	.50	1.1e-2	.92	3.1e-2	.76	2.0e-2	.74
4	11	1.5e4	2.0e-1	4.4e-1		1.7e-1		2.9e-1		1.8e-1	
	21	1.9e5	1.0e-1	3.2e-1	.45	9.8e-2	.78	1.9e-1	.60	1.2e-1	.58
	41	2.8e6	5.0e-2	2.3e-1	.48	5.5e-2	.83	1.2e-1	.67	7.7e-2	.66

TABLE 6.2

Errors of approximate solution computed by FMM compared to exact solution of (1.1), where H is as in (2.1) and $G(Du(x)) = \|Du(x)\|_p$. The variables d, m, and n are the dimension, the number of nodes in each dimension, and the total number of nodes, respectively. Other variables are the spacing h between grid nodes, the $\mathcal{L}_\infty$-error e_∞, the $\mathcal{L}_\infty$ convergence rate r_∞, the $\mathcal{L}_1$-error e_1, and the $\mathcal{L}_1$ convergence rate r_1.

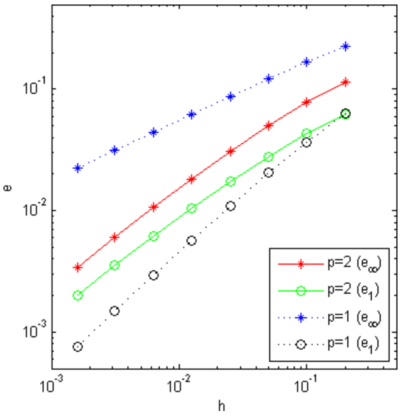


FIG. 6.2. *Errors of approximate solution computed by FMM compared to exact solution of (1.1), where H is as in (2.1), $G(Du(x)) = \|Du(x)\|_p$ and $d = 2$. For descriptions of variables see Table 6.2.*

to achieve a joint goal state. This goal should be achieved in minimal time from any initial state in the domain without incurring collisions.

Let the state of the dark-colored robot be $(x_1, x_2) \in \mathbb{R}^2$ and the state of the light-colored robot be $(x_3, x_4) \in \mathbb{R}^2$ so that the combined state of the two robots is

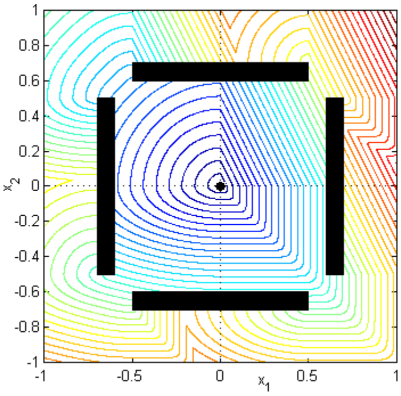


FIG. 6.3. *Contours of $\underline{u}$ computed for the anisotropic problem with G defined by (2.6). The black circle at $O = (0, 0)$ indicates $\underline{\partial\Omega}$ and in the black rectangles, $c(x) \gg 1$. In these regions, $\underline{u}$ has purposefully not been computed.*

$(x_1, x_2, x_3, x_4) \in \mathbb{R}^4$. We define the control-theoretic action set

$$\mathcal{A}(x) = \{a \mid F(a) = \|(\|(a_1, a_2)\|_2, \|(a_3, a_4)\|_2)\|_\infty \leq 1/c(x)\}.$$

Proposition 2.7 states that we can use the dual (2.2) of F to obtain

$$G(x, Du(x)) = \|(\|(\partial_1 u(x), \partial_2 u(x))\|_2, \|(\partial_3 u(x), \partial_4 u(x))\|_2)\|_1, \tag{6.1}$$

where $Du(x) = (\partial_1 u(x), \partial_2 u(x), \partial_3 u(x), \partial_4 u(x))$. Where x is a collision state, we set $c(x) \gg 1$. For all other states x, $c(x) = 1$.

We can compute $\underline{u}$ using FMM since G is a mixed p-norm, satisfying Properties 1 to 4. The domain Ω is discretized using a uniform orthogonal grid of $(81 \times 21)^2$ nodes. The discretized equation for an update from a single simplex (5.5) is given by

$$\begin{aligned}
&\left\|\left(\left\|\left(\frac{(\underline{u}_0 - \underline{u}_1)^+}{-h_1}, \frac{(\underline{u}_0 - \underline{u}_2)^+}{-h_2}\right)\right\|_2, \left\|\left(\frac{(\underline{u}_0 - \underline{u}_3)^+}{-h_3}, \frac{(\underline{u}_0 - \underline{u}_4)^+}{-h_4}\right)\right\|_2\right)\right\|_1 \\
&= \sqrt{\left(\frac{(\underline{u}_0 - \underline{u}_1)^+}{h_1}\right)^2 + \left(\frac{(\underline{u}_0 - \underline{u}_2)^+}{h_2}\right)^2} + \sqrt{\left(\frac{(\underline{u}_0 - \underline{u}_3)^+}{h_3}\right)^2 + \left(\frac{(\underline{u}_0 - \underline{u}_4)^+}{h_4}\right)^2} \\
&= c(\underline{x}_0),
\end{aligned}$$

where $(y)^+ = \max(0, y)$. Since this equation is quartic in $\underline{u}_0$ it is difficult to solve analytically for $\underline{u}_0$. However, Theorem 3.3 tells us that we can determine $\underline{u}_0$ that satisfies $\underline{H}(\underline{u}_0) = 0$ uniquely. As a result, numerical root-finders can easily be used to compute this solution.

Once we have determined $\underline{u}$ using FMM, we approximately solve the following ODE to determine an optimal collision-free trajectory from any initial state to the goal:

$$\frac{dx}{dt} = \arg\max_{a \in \mathcal{A}(x)} (-D\underline{u}(x) \cdot a) - 1. \tag{6.2}$$

Solving this ODE is not the focus of the paper so we take a simple approach. The ODE (6.2) is discretized using forward Euler. The gradient $D\underline{u}(x)$ is determined

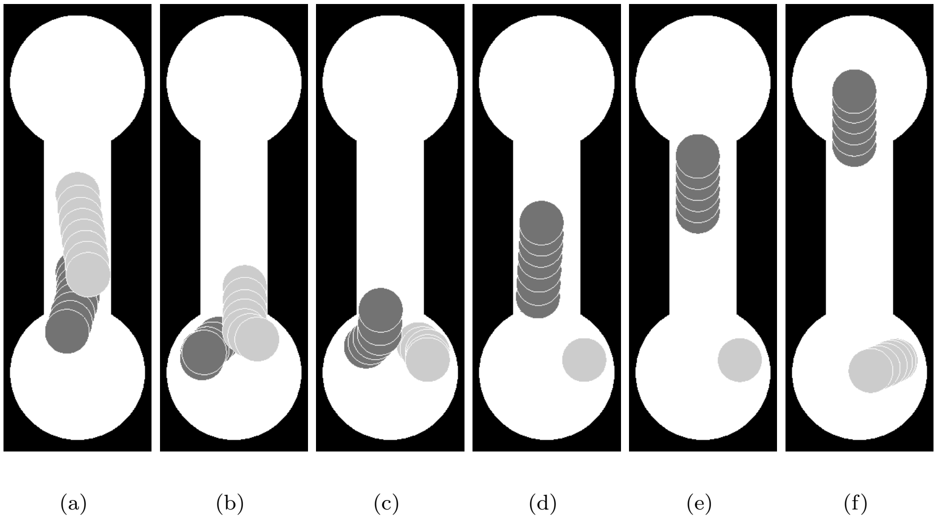


Fig. 6.4. *Two-robot coordinated optimal navigation problem. The joint goal is for the dark-colored robot to reach the center of the upper bulb and light-colored robot to reach the center of the lower bulb. Black indicates an obstacle region. The sequence shows the robots achieving their joint goal without collision from a particular initial state. The solution of (1.1), where H is as in (2.1) and G is given by (6.1) allows quick determination of the optimal collision-free trajectories for both robots from any initial condition [1].*

by a first-order finite difference scheme. At each time step, each robot moves at its maximum speed $1/c(x)$ in the direction of the relevent components of the negative gradient, e.g. (for dark-colored robot):

$$(a_1, a_2) = \frac{-1}{c(x)} \frac{(\partial_1 \underline{u}(x), \partial_2 \underline{u}(x))}{\|(\partial_1 \underline{u}(x), \partial_2 \underline{u}(x))\|}.$$

If the relevant components of the gradient fall below a small threshold the robot will not move at all, as is the case for the light-colored robot in Figures 6.4(d) and 6.4(e).

7. Conclusion.

We have described a new class of anisotropic static HJ PDEs corresponding to optimal control problems with axis-aligned but anisotropic and potentially asymmetric action sets. Assuming Properties 1 to 4, we showed that uniqueness, monotonicity and causality hold for a standard finite difference discretization of these PDEs on an orthogonal grid, and so the Fast Marching Method can be used to approximate their solution. We also demonstrated several methods for reducing the number of neighboring simplices which must be considered when computing node updates, including novel methods which work when the PDE and/or grid are asymmetric. In future work, these results might be generalized to unstructured grids.

Acknowledgments. We would like to thank Alexander Vladimirsky for enlightening discussions of FMM and OUM. We would also like to thank Adam Oberman for discussions and insights regarding [3].

Appendix A. Duals.

In this appendix we show that the dual relationship defined in (2.2) is meaningful, by proving Propositions 2.5 and 2.6. Let us first recall (2.2), the definition of the dual of G:

$$G_*(q) = \max_{G(a)\le 1} (q \cdot a). \tag{A.1}$$

Restricting the domain of G to any orthant in q space results in a norm satisfying Properties 1 to 4, which we call an *orthant-associated norm*. The function G_* can be formed using the duals of the orthant-associated norms. Propositions 2.5 and 2.6 are proved using the well known properties of dual norms.

Let $\mathcal{S} = \{(s_1, s_2, \dots, s_d) \mid s_j \in \{-1, +1\}, 1 \le j \le d\}$, such that $s \in \mathcal{S}$ corresponds to one of the 2^d orthants. Let

$$G_s(q) = G(R^s(q)), \tag{A.2}$$

where $R^s(q) = (s_1|q_1|, s_2|q_2|, \dots, s_d|q_d|)$.

Lemma A.1. *Let a function G satisfy Properties 1 to 4. For all $s \in \mathcal{S}$, G_s is a norm and satisfies one-sided monotonicity.*

Proof. If $q = 0$, then $G_s(q) = G(0) = 0$, by the definiteness of G. Otherwise, if $q \ne 0$, there is a j such that $s_j|q_j| \ne 0$, and so, $G(q) \ne 0$, by the definiteness of G. Therefore, G_s is definite. Furthermore, G_s is nonnegative, since G is nonnegative.

Also,

$$\begin{aligned} G_s(q_a + q_b) &= G(R^s(q_a + q_b)) \\ &\le G(R^s(q_a) + R^s(q_b)) \\ &\le G(R^s(q_a)) + G(R^s(q_b)) \\ &= G_s(q_a) + G_s(q_b). \end{aligned}$$

The first inequality applies, by the one-sided monotonicity of G, since $s_j|q_{a,j} + q_{b,j}| \le s_j|q_{a,j}| + s_j|q_{b,j}|$. The second inequality applies because G satisfies the triangle inequality. Therefore, G_s satisfies the triangle inequality.

Moreover,

$$\begin{aligned} G_s(tq) &= G(R^s(tq)) \\ &= G(|t|R^s(q)) \\ &= |t|G(R^s(q)) = |t|G_s(q), \end{aligned}$$

by the one-sided homegeneity of G, so G_s is homogeneous. Therefore, G_s satisfies the properties required of a norm.

Let $q_a, q_b \in \mathbb{R}^d$, such that $q_a \trianglerighteq q_b$. Note that $R^s(q_a) \trianglerighteq R^s(q_b)$. By the one-sided monotonicity of G,

$$\begin{aligned} G_s(q_a) &= G(R^s(q_a)) \\ &\ge G(R^s(q_b)) = G_s(q_b). \end{aligned}$$

Therefore, G_s satisfies one-sided monotonicity. □

Denote $\|\cdot\|_s = G_s$ in the following discussion to emphasize that G_s is a norm, for all $s \in \mathcal{S}$. We call $\|\cdot\|_s$ the s-orthant-associated norm of G. The dual of a norm $\|\cdot\|$ is defined as

$$\|q\|_* = \max_{\|a\|\le 1} (q \cdot a). \tag{A.3}$$

and is also a norm. Note this definition is identitical to (A.1). We further show that $\|\cdot\|_*$ satisfies one-sided monotonicity, if $\|\cdot\|$ satisfies one-sided monotonicity. But first we show that there exists a maximizer of (A.1) (or (A.3)) in the same orthant as the argument q.

Lemma A.2. *Let $q \in \mathbb{R}^d$. Let a function F Properties 1 to 4. There exists $\tilde{a}$ such that*

$$\tilde{a} \in \arg\max_{F(a)\leq 1} (q \cdot a)$$

and $q_j \tilde{a}_j \geq 0, \forall j$.

Proof. Let $\hat{a} \in \arg\max_{F(a)\leq 1}(q \cdot a)$. Define $\tilde{a} \in \mathbb{R}^d$:

$$\tilde{a}_j = \begin{cases} \hat{a}_j, & \text{if } q_j \hat{a}_j \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

Note $q \cdot \tilde{a} \geq q \cdot \hat{a}$, since $q_j \tilde{a}_j \geq q_j \hat{a}_j$, for each j. Also, $F(\hat{a}) \geq F(\tilde{a})$, by the one-sided monotonicity of F. This implies $\tilde{a} \in \arg\max_{F(a)\leq 1}(q \cdot a)$. □

Lemma A.3. *Let $\|\cdot\|_*$ be the dual of norm $\|\cdot\|$. If $\|\cdot\|$ satisfies one-sided monotonicity, then $\|\cdot\|_*$ satisfies one-sided monotonicity.*

Proof. Let $\|\cdot\|$ satisfy one-sided monotonicity. Let $q_a, q_b \in \mathbb{R}^d$, such that $q_a \trianglerighteq q_b$. Let $\tilde{b} \in \arg\max_{\|b\|\leq 1}(q_b \cdot b)$ such that $q_{b,j}\tilde{b}_j \geq 0, \forall j$, by Lemma A.2. Since $q_a \trianglerighteq q_b$ we know that $q_{a,j}\tilde{b}_j \geq q_{b,j}\tilde{b}_j \geq 0$. It follows that $q_a \cdot \tilde{b} \geq q_b \cdot \tilde{b}$. Therefore,

$$\begin{aligned} \|q_a\|_* &= \max_{\|b\|\leq 1} (q_a \cdot b) \\ &\geq q_a \cdot \tilde{b} \\ &\geq q_b \cdot \tilde{b} \\ &= \max_{\|b\|\leq 1} (q_b \cdot b) = \|q_b\|_*. \end{aligned}$$

□

We show that G_*, the dual of G from (A.1), can be defined in terms of the duals of orthant-associated norms $\|\cdot\|_s$ but first demonstrate that the dual of G can be computed by substituting the constraint $\|a\|_s \leq 1$ for $G(a) \leq 1$ in (A.1), where s is the orthant of the parameter q.

Lemma A.4. *Let a function G satisfy Properties 1 to 4. Let $q \in \mathbb{R}^d$ and $s \in S$ be such that $s_j q_j \geq 0$, for any j. Then*

$$\max_{G(a)\leq 1} (q \cdot a) = \max_{\|a\|_s\leq 1} (q \cdot a).$$

Proof. Let $q \in \mathbb{R}^d$ and $s \in S$ be such that $s_j q_j \geq 0$, for any j. Let $\tilde{a} \in \arg\max_{G(a)\leq 1}(q \cdot a)$ such that $q_j \tilde{a}_j \geq 0, \forall j$, by Lemma A.2. This implies $s_j \tilde{a}_j \geq 0$, for each j, and so

$$\|\tilde{a}\|_s = G(R^s(\tilde{a})) = G(\tilde{a}) \leq 1. \tag{A.4}$$

Let $b \in \mathbb{R}^d$ be such that $\|b\|_s = G(R^s(b)) \leq 1$. Define $\tilde{b} \in \mathbb{R}^d$:

$$\tilde{b}_j = \begin{cases} b_j, & \text{if } s_j b_j \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

We have

$$G(\tilde{b}) = G(R^s(\tilde{b})) \leq G(R^s(b)) = \|b\|_s \leq 1, \tag{A.5}$$

by the one-sided monotonicity of G. Furthermore,

$$q \cdot b \leq q \cdot \tilde{b} \leq q \cdot \tilde{a}.$$

The first inequality applies because $q_j \tilde{b}_j \geq q_j b_j$, for each j, and the second inequality applies because $\tilde{a} \in \arg\max_{G(a)\leq 1}(q \cdot a)$. and $G(\tilde{b}) \leq 1$, by (A.5). This together with (A.4) implies that $\tilde{a} \in \arg\max_{\|a\|_s \leq 1}(q \cdot a)$.

Therefore,

$$\max_{G(a)\leq 1}(q \cdot a) = \max_{\|a\|_s \leq 1}(q \cdot a),$$

since $a = \tilde{a}$ maximizes $(q \cdot a)$ over both constraints on a. □

Lemma A.5. *Let a function G satisfy Properties 1 to 4. For any $q \in \mathbb{R}^d$ and $s \in S$, such that $s_j q_j \geq 0$, for any j,*

$$G_*(q) = \|q\|_{s*}$$

where $\|\cdot\|_{s*}$ is the dual norm of $\|\cdot\|_s$.

Proof. The conclusion follows trivially from (A.1), Lemma A.4, and (A.3):

$$\begin{aligned} G_*(q) &= \max_{G(a)\leq 1}(q \cdot a) \\ &= \max_{\|a\|_s \leq 1}(q \cdot a) = \|q\|_{s*}. \end{aligned}$$

□

We now prove Propositions 2.5 and 2.6.

Proof. (of Proposition 2.5) Let $q \in \mathbb{R}^d$. Define s as

$$s_j = \begin{cases} +1, & \text{if } q_j \geq 0 \\ -1, & \text{otherwise,} \end{cases} \tag{A.6}$$

for $1 \leq j \leq d$. Since $s_j q_j \geq 0$, for all j, by Lemma A.5,

$$G_*(q) = \|q\|_{s*}.$$

Therefore, G_* is nonnegative definite (Property 1), because $\|q\|_{s*}$ is nonnegative definite.

Let $q_a, q_b \in \mathbb{R}^d$. Let $q_c = q_a + q_b$. Define s_a, s_b, and s_c as in (A.6) for $q = q_a$, $q = q_b$, and $q = q_c$, respectively. Recall that norms $\|\cdot\|_{s_a*}$, $\|\cdot\|_{s_a*}$ and $\|\cdot\|_{s_a*}$ are one-sided monotone, by Lemmas A.1 and A.3.

Define $\tilde{q}_a \in \mathbb{R}^d$:

$$\tilde{q}_{a,j} = \begin{cases} q_{a,j}, & q_{a,j} q_{b,j} \geq 0, \\ 0, & \text{if } q_{a,j} q_{b,j} \leq 0 \text{ and } |q_{a,j}| \leq |q_{b,j}|, \\ q_{c,j}, & \text{otherwise (i.e., if } q_{a,j} q_{b,j} \leq 0 \text{ and } |q_{a,j}| \geq |q_{b,j}|), \end{cases}$$

for $1 \leq j \leq d$. Note $q_a \trianglerighteq \tilde{q}_a$. By the one-sided monotonicity of $\|\cdot\|_{s_a*}$, we have $\|\tilde{q}_a\|_{s_a*} \leq \|q_a\|_{s_a*}$. Also note $\tilde{q}_{a,j} q_{c,j} \geq 0$, for all j. Finally, note $s_{a_j} \tilde{q}_{a,j} = s_{c_j} \tilde{q}_{a,j} \geq 0$, since $s_{a,j} \neq s_{c,j}$ if and only if $q_{a,j}q_{b,j} \leq 0$ and $|q_{a,j}| \leq |q_{b,j}|$, i.e., in the case where $\tilde{q}_{a,j} = 0$. By Lemma A.5,

$$\|\tilde{q}_a\|_{s_a*} = G_*(\tilde{q}_a) = \|\tilde{q}_a\|_{s_c*}.$$

Let $\tilde{q}_b \in \mathbb{R}^d$ be such that

$$\tilde{q}_{b,j} = q_{c,j} - \tilde{q}_{a,j} = \begin{cases} q_{b,j}, & q_{a,j}q_{b,j} \geq 0, \\ q_{c,j}, & \text{if } q_{a,j}q_{b,j} \leq 0 \text{ and } |q_{a,j}| \leq |q_{b,j}|, \\ 0, & \text{otherwise (i.e., if } q_{a,j}q_{b,j} \leq 0 \text{ and } |q_{a,j}| \geq |q_{b,j}|). \end{cases}$$

for $1 \leq j \leq d$. Note $q_b \trianglerighteq \tilde{q}_b$, for all j. By the one-sided monotonicity of $\|\cdot\|_{s_b*}$, we have $\|\tilde{q}_b\|_{s_b*} \leq \|q_b\|_{s_b*}$. Also note that $\tilde{q}_{b,j}q_{c,j} \geq 0$. Finally, note $s_{b_j}\tilde{q}_{b,j} = s_{c_j}\tilde{q}_{b,j} \geq 0$, since $s_{b,j} \neq s_{c,j}$ if and only if $q_{a,j}q_{b,j} \leq 0$ and $|q_{a,j}| \geq |q_{b,j}|$, i.e., in the case where $\tilde{q}_{b,j} = 0$. By Lemma A.5,

$$\|\tilde{q}_b\|_{s_b*} = G_*(\tilde{q}_b) = \|\tilde{q}_b\|_{s_c*}.$$

Therefore, we have

$$\begin{aligned} G_*(q_a) + G_*(q_b) &= \|q_a\|_{s_a*} + \|q_b\|_{s_b*} \\ &\geq \|\tilde{q}_a\|_{s_a*} + \|\tilde{q}_b\|_{s_b*} \\ &= \|\tilde{q}_a\|_{s_c*} + \|\tilde{q}_b\|_{s_c*} \\ &\geq \|\tilde{q}_a + \tilde{q}_b\|_{s_c*} \\ &= \|q_c\|_{s_c*} = G_*(q_c), \end{aligned}$$

demonstrating that G_* satisfies the triangle inequality (Property 2).

Let $t \geq 0$ and $q \in \mathbb{R}^d$. Define $s \in S$ as in (A.6). By Lemma A.5,

$$G_*(tq) = \|tq\|_{s*} = t\|q\|_{s*} = tG_*(q),$$

so G_* is one-sided homogeneous (Property 3).

Let $q_a, q_b \in \mathbb{R}^d$ be such that $q_a \trianglerighteq q_b$ for all j. Define s_a and s_b as in (A.6) for $q = q_a$ and $q = q_b$, respectively. Note that $s_a = s_b$. Recall that $\|\cdot\|_{s_a*}$ is one-sided monotone, by Lemmas A.1 and A.3. We have

$$\begin{aligned} G_*(q_a) &= \|q_a\|_{s_a*} \\ &\geq \|q_b\|_{s_a*} \\ &= \|q_b\|_{s_b*} = G_*(q_b), \end{aligned}$$

so G_* is one-sided monotone (Property 4). □

Proof. (of Proposition 2.6) Let $a \in \mathbb{R}^d$ and $s \in S$. Let G satisfy Properties 1 to 4. By Proposition 2.5, G_* also satisfies 1 to 4. Let $\|\cdot\|_{*s} = G_{*s}(\cdot)$ be the s-orthant-associated norm of G_*. We have

$$\begin{aligned} \|a\|_{*s} &= G_*(R^s(a)) \\ &= \|a\|_{s*}, \end{aligned} \tag{A.7}$$

where the first equality is by (A.2), and the second equality is by Lemma A.5, since $s_j R^s_j(a) \geq 0$, where $R^s_j(a)$ is the jth component of $R^s(a)$.

Let $q \in \mathbb{R}^d$. Define s as in (A.6). Note $s_j q_j \geq 0$ for all j. We have

$$\begin{aligned} G_{**}(q) &= \max_{G_*(a) \leq 1} (q \cdot a) \\ &= \max_{\|a\|_{*s} \leq 1} (q \cdot a) \\ &= \max_{\|a\|_{s*} \leq 1} (q \cdot a) \\ &= \|q\|_s = G(q), \end{aligned}$$

by (A.1), Lemma A.4, (A.7), (A.3), and Lemma A.5. Therefore, $G = G_{**}$. □

Appendix B. Analytic Solutions.

We derive analytic node updates for the cases where $G(Du(x)) = \|Du(x)\|_p$ and $p = 1$, $p = 2$, or $p = \infty$. The derivation for $p = 2$ fixes some errors in the appendix of [16]. In [1] we demonstrated that these cases could be treated by FMM and are useful for robotic applications. However, here we generalize the derivations of the update formuli to any dimension and grid spacing.

We are concerned with implementing the `Solve` routine in Algorithm 5. Let $(v_1, v_2, \ldots, v_m)$ be the values of the neighboring nodes in the simplex $s \in \mathcal{S}_\sigma$ and $(h_1, h_2, \ldots, h_m)$ be the corresponding grid spacings. We are solving for μ. Non-causal node values must already have been eliminated by Algorithm 4, so $\mu > \max_{1 \leq j \leq m} v_j$ for any solution of interest.

B.1. $p = 1$.

From (5.5) we have

$$\sum_j \left(\frac{|\mu - v_j|}{h_j} \right) = c.$$

Assume $\mu > \max_{1 \leq j \leq m} v_j$ and multiply through by $\prod_l h_l$ to obtain

$$\sum_j \left(\prod_{l \neq j} h_l \right) \mu - \sum_j \left(\prod_{l \neq j} h_l \right) v_j = \left(\prod_l h_l \right) c.$$

Then solve for μ to get

$$\mu = \frac{\sum_j \left(\prod_{l \neq j} h_l \right) v_j + \prod_l h_l c}{\sum_j \prod_{l \neq j} h_l}.$$

B.2. $p = 2$.

From (5.5) we have

$$\sum_j \left(\frac{\mu - v_j}{h_j} \right)^2 = c^2.$$

Multiply through by $\prod_l h_l^2$ to get

$$\left[\sum_j \prod_{l \neq j} h_l^2 \right] \mu^2 - 2 \left[\sum_j \left(\prod_{l \neq j} h_l^2 \right) v_j \right] \mu + \sum_j \left(\prod_{l \neq j} h_l^2 \right) v_j^2 - \left(\prod_l h_l^2 \right) c^2 = 0.$$

Then, using the quadratic formula, solve for μ:

$$\mu = \frac{2\sum_j \left(\prod_{l\neq j} h_l^2\right) v_j + \sqrt{\begin{array}{l}\left[2\sum_j \left(\prod_{l\neq j} h_l^2\right) v_j\right]^2 \\ -\,4\left[\sum_j \prod_{l\neq j} h_l^2\right]\left[\sum_j \left(\prod_{l\neq j} h_l^2\right) v_j^2 - \left(\prod_l h_l^2\right) c^2\right]\end{array}}}{2\sum_j \prod_{l\neq j} h_l^2}$$

$$= \frac{\sum_j \left(\prod_{l\neq j} h_l^2\right) v_j + \prod_l h_l \sqrt{\begin{array}{l}\left(\sum_j \prod_{l\neq j} h_l^2\right) c^2 \\ +\sum_{j_1}\sum_{j_2}\left(\prod_{l\neq j_1,j_2} h_l^2\right) v_{j_1} v_{j_2} \\ -\sum_{j_1}\sum_{j_2}\left(\prod_{l\neq j_1,j_2} h_l^2\right) v_{j_1}^2\end{array}}}{\sum_j \prod_{l\neq j} h_l^2}.$$

We only consider the larger of the two quadratic solutions since the alternative will result in $\mu \leq \max_{1\leq j\leq m} v_j$. The last two terms of the discriminant can be made more concise as follows:

$$\begin{aligned}
&\sum_{j_1}\sum_{j_2}\left(\prod_{l\neq j_1,j_2} h_l^2\right) v_{j_1}v_{j_2} - \sum_{j_1}\sum_{j_2}\left(\prod_{l\neq j_1,j_2} h_l^2\right) v_{j_1}^2 \\
&= \sum_{j_1}\sum_{j_2\neq j_1}\left(\prod_{l\neq j_1,j_2} h_l^2\right) v_{j_1}v_{j_2} + \sum_{j_1}\sum_{j_2=j_1}\left(\prod_{l\neq j_1,j_2} h_l^2\right) v_{j_1}v_{j_2} \\
&\quad - \sum_{j_1}\sum_{j_2\neq j_1}\left(\prod_{l\neq j_1,j_2} h_l^2\right) v_{j_1}^2 - \sum_{j_1}\sum_{j_2=j_1}\left(\prod_{l\neq j_1,j_2} h_l^2\right) v_{j_1}^2 \\
&= 2\sum_{j_1}\sum_{j_2>j_1}\left(\prod_{l\neq j_1,j_2} h_l^2\right) v_{j_1}v_{j_2} + \sum_j\left(\prod_{l\neq j} h_l^2\right) v_j^2 \\
&\quad - \sum_{j_1}\sum_{j_2>j_1}\left(\prod_{l\neq j_1,j_2} h_l^2\right)(v_{j_1}^2 + v_{j_2}^2) - \sum_j\left(\prod_{l\neq j} h_l^2\right) v_j^2 \\
&= \sum_{j_1}\sum_{j_2>j_1}\left(\prod_{l\neq j_1,j_2} h_l^2\right) 2v_{j_1}v_{j_2} - \sum_{j_1}\sum_{j_2>j_1}\left(\prod_{l\neq j_1,j_2} h_l^2\right)(v_{j_1}^2 + v_{j_2}^2) \\
&= -\sum_{j_1}\sum_{j_2>j_1}\left(\prod_{l\neq j_1,j_2} h_l^2\right)(v_{j_1} - v_{j_2})^2.
\end{aligned}$$

B.3. $p = \infty$**.** From (5.5) we have

$$\max_j \left(\frac{|\mu - v_j|}{h_j}\right) = c.$$

Assume $\mu > \max_{1\leq j\leq m} v_j$ solve for μ to obtain

$$\mu = \min_j \left(v_j + h_j c\right).$$

This $p = \infty$ case is identical to the update formula for Dijkstra's algorithm for shortest path on a discrete graph.

REFERENCES

[1] K. Alton and I. Mitchell. Optimal path planning under different norms in continuous state spaces. In *Proceedings of the International Conference on Robotics and Automation*, pages 866–872, 2006.

[2] M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations.* Birkhauser, Boston and Basel and Berlin, 1997.

[3] G. Barles and P. E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptot. Anal.*, 4:271–283, 1991.

[4] F. Bornemann and C. Rasch. Finite-element discretization of static Hamilton-Jacobi equations based on a local variational principle. *Comput. Vis. Sci.*, 9:57–69, 2006.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, UK, 2004.

[6] P. Bruin. Continuity of convex functions. http://planetmath.org/encyclopedia/ContinuityOfConvexFunctions.html.

[7] M. G. Crandall, H. Ishii, and P. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc.*, 27(1):1–67, 1992.

[8] E. W. Dijkstra. A note on two problems in connection with graphs. *Numer. Math. 1*, pages 269–271, 1959.

[9] L. C. Evans. *Partial Differential Equations.* American Mathematical Society, Providence, Rhode Island, 1991.

[10] P. A. Gremaud and C. M. Kuster. Computational study of fast methods for the eikonal equation. *SIAM J. Sci. Comput.*, 27:1803–1816, 2006.

[11] M. T. Heath. *Scientific Computing: An Introductory Survey, Second Edition.* McGraw-Hill, New York, 2002.

[12] S.-R. Hysing and S. Turek. The eikonal equation: Numerical efficiency vs. algorithmic complexity on quadrilateral grids. In *Proceedings of Algoritmy 2005*, pages 22–31, 2005.

[13] C. Y. Kao, S. Osher, and Y. Tsai. Fast sweeping methods for static Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 42(6):2612–2632, 2004-2005.

[14] R. Kimmel and J. A. Sethian. Optimal algorithm for shape from shading and path planning. *J. Math. Imaging Vision*, 14(3):237–244, May 2001.

[15] H. J. Kushner and P. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time.* Springer-Verlag, Berlin and New York, 1992.

[16] I. M. Mitchell and S. Sastry. Continuous path planning with multiple constraints. Technical Report UCB/ERL M03/34, UC Berkeley Engineering Research Laboratory, August 2003.

[17] L. C. Polymenakos, D. P. Bertsekas, and J. N. Tsitsiklis. Implementation of efficient algorithms for globally optimal trajectories. *IEEE Trans. Automat. Control*, 43:278–283, 1998.

[18] J. Qian, Y. Zhang, and H. Zhao. A fast sweeping method for static convex Hamilton-Jacobi equations. *to appear in J. Sci. Comput.*, 2006.

[19] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci.*, 93:1591–1595, 1996.

[20] J. A. Sethian and A. Vladimirsky. Fast methods for eikonal and related Hamilton-Jacobi equations on unstructured meshes. *Proc. Natl. Acad. Sci.*, 97(11):5699–5703, May 2000.

[21] J. A. Sethian and A. Vladimirsky. Ordered upwind methods for static Hamilton-Jacobi equations. *Proc. Natl. Acad. Sci.*, 98(20):11069–11074, 2001.

[22] J. A. Sethian and A. Vladimirsky. Ordered upwind methods for static Hamilton-Jacobi equations: Theory and applications. *SIAM J. Numer. Anal.*, 41(1):323–363, 2003.

[23] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Trans. Automat. Control*, 40(9):1528–1538, 1995.

[24] L. Yatziv, A. Bartesaghi, and G. Sapiro. O(n) implementation of the fast marching method. *J. Comput. Phys.*, 212(2):393–399, March 2006.

[25] H. Zhao. A fast sweeping method for Eikonal equations. *Math. Comp.*, 74(250):603–627, May 2004.