

Fast Sweeping Algorithms for a Class of Hamilton-Jacobi Equations

Yen-Hsi Richard Tsai^{*†}, Li-Tien Cheng^{‡§}, Stanley Osher[¶], and Hong-Kai Zhao^{||}

Abstract

We derive a Godunov-type numerical flux for the class of strictly convex, homogeneous Hamiltonians that includes $H(p, q) = \sqrt{ap^2 + bq^2 - 2cpq}$, $c^2 < ab$. We combine our Godunov numerical fluxes with simple Gauss-Seidel type iterations for solving the corresponding Hamilton-Jacobi Equations. The resulting algorithm is fast since it does not require a sorting strategy as found, e.g., in the fast marching method. In addition, it provides a way to compute solutions to a class of HJ equations for which the conventional fast marching method is not applicable. Our experiments indicate convergence after a few iterations, even in rather difficult cases.

1 Introduction

Hamilton-Jacobi equations have a rich pool of applications, ranging from those of optimal control theory, geometrical optics, to essentially any problem that needs the (weighted) distance function [13]. Examples include crystal growth, ray tracing, etching, robotic motion planning, and computer vision. Solutions of these types of equations usually develop singularities in their derivatives, and thus, the unique viscosity solution [5] is sought.

^{*}Department of Mathematics and PACM, Princeton University, Princeton, New Jersey 08542. Email: ytsai@math.princeton.edu

[†]Research supported by ONR N00014-97-1-0027, DARPA/NSF VIP grant NSF DMS 9615854 and ARO DAAG 55-98-1-0323

[‡]Department of Mathematics, UCSD, La Jolla, CA 92093-0112. Email: lcheng@math.ucsd.edu

[§]Research supported by NSF Grant #0112413 and #0208449

[¶]Department of Mathematics, UCLA, Los Angeles, California 90095. Email: sjo@math.ucla.edu

^{||}Department of Mathematics, UCI, Irvine, CA 92697-3875. Email: zhao@math.ucla.edu

In this article, we focus on the class of time independent Hamilton-Jacobi Equations with Dirichlet boundary condition:

$$H(\mathbf{x}, \nabla u) = r(\mathbf{x}), \quad u|_{\Gamma} = 0;$$

$H(\mathbf{x}, \mathbf{p})$ are strictly convex, non-negative, and $\lim_{\lambda \rightarrow 0} H(\mathbf{x}, \lambda \mathbf{p}) = 0$. We explain our method using this following important model equation:

$$H(\phi_x, \phi_y) = \sqrt{a\phi_x^2 + b\phi_y^2 - 2c\phi_x\phi_y} = r, \quad (1)$$

where $\phi : \mathbb{R}^2 \mapsto \mathbb{R}$ continuous and a, b, c and r can either be constants or scalar functions, in which case H depends also on x , defined on \mathbb{R}^2 , satisfying $ab > c^2$, $a, b, r > 0$. With $a = b = 1$, and $c = 0$, we have the standard eikonal equation for which many numerical methods have been developed. This equation has the essential features of HJ equations with convex Hamiltonians so that we can easily explain our algorithm, and is general enough that fast marching is not applicable.

In the following subsections, we will review some of the solution methods for the eikonal equation since it forms the motivation of our work. We then present a fast Gauss-Seidel type iteration method for equation (1) which utilizes a monotone upwind Godunov flux for the Hamiltonian. We show numerically that this algorithm can be applied directly to equations of the above type with variable coefficients.

1.1 Solving Eikonal Equations

In geometrical optics [9], the eikonal equation

$$\sqrt{\phi_x^2 + \phi_y^2} = r(x, y) \quad (2)$$

is derived from the leading term in an asymptotic expansion

$$e^{i\omega(\phi(x,y)-t)} \sum_{j=0}^{\infty} A_j(x, y, t) (i\omega)^{-j}$$

of the wave equation:

$$w_{tt} - c^2(x, y)(w_{xx} + w_{yy}) = 0,$$

where $r(x, y) = 1/|c(x, y)|$, is the function of slowness. The level sets of the solution ϕ can thus be interpreted as the first arrival time of the wave front that is initially Γ . It can also be interpreted as the “distance” function to Γ .

We first restrict our attention for now to the case in which $r = 1$. Let Γ be a closed subset of \mathbb{R}^2 . It can be shown easily that the distance function defined by

$$d(\mathbf{x}) = \text{dist}(\mathbf{x}, \Gamma) := \min_{p \in \Gamma} \|\mathbf{x} - p\|, \quad \mathbf{x} = (x, y) \in \mathbb{R}^2,$$

is the viscosity solution to equation (2) with the boundary condition

$$\phi(x, y) = 0 \text{ for } (x, y) \in \Gamma.$$

Rouy and Tourin [19] proved the convergence to the viscosity solution of an iterative method solving equation (2) with the Godunov Hamiltonian approximating $\|\nabla\phi\|$. The Godunov Hamiltonian function can be written in the following form:

$$H_G(p_-, p_+, q_-, q_+) = \sqrt{\max\{p_-^+, p_+^-\}^2 + \max\{q_-^+, q_+^-\}^2} \quad (3)$$

where $p_{\pm} = D_{\pm}^x \phi_{i,j}$, $q_{\pm} = D_{\pm}^y \phi_{i,j}$, $D_{\pm}^x \phi_{i,j} = \pm(\phi_{i\pm 1,j} - \phi_{i,j})/h$ and accordingly for $D_{\pm}^y \phi_{i,j}$, and $x^+ = \max(x, 0)$, $x^- = -\min(x, 0)$.

Osher [12] provided a link to time dependent eikonal equations by proving that the t -level set of $\phi(x, y)$ is the zero level set of the viscosity solution of the evolution equation at time t

$$\psi_t = \|\nabla\psi\|$$

with appropriate initial conditions. In fact, the same is true for a very general class of Hamilton-Jacobi equations (see [12]). As a consequence, one can try to solve the time-dependent equation by the level set formulation [16] with high order approximations on the partial derivatives [8][17]. Crandall and Lions proved that the discrete solution obtained with a consistent, monotone Hamiltonian converges to the desired viscosity solution [4].

Tsitsiklis [24] combined heap sort with a variant of the classical Dijkstra algorithm to solve the steady state equation of the more general problem

$$\|\nabla\phi\| = r(\mathbf{x}).$$

This was later rederived in [22] and also reported in [7]. It has become known as the fast marching method whose complexity is $\mathcal{O}(N \log(N))$, where N is the number of grid points. Osher and Helmsen [14] have extended the fast marching type method to somewhat more general Hamilton-Jacobi equations. We will comment on this in a following section.

1.2 Anisotropic Eikonal Equation

We return to the Hamiltonian in question: $H(p, q) = \sqrt{ap^2 + bq^2 - 2cpq}$. Writing the quadratic form as

$$ap^2 + bq^2 - 2cpq = \begin{pmatrix} p & q \end{pmatrix} \begin{pmatrix} a & -c \\ -c & b \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix},$$

it is easy to see that we can diagonalize the symmetric matrix in the middle of the equation for our previously noted choices of a, b, c and find a coordinate system ξ - η such that after rescaling, the Hamiltonian becomes

$$H(\tilde{p}, \tilde{q}) = \sqrt{\tilde{p}^2 + \tilde{q}^2}.$$

The eigensystem of the above matrix defines the anisotropy. Indeed, the authors in [14] proposed to solve the constant coefficient equation (1) by first transforming it to equation (2) in the ξ - η coordinate system.

This anisotropy occurs in fields such as ray tracing in special media, e.g. crystals, in which there are “preferred” directions. Furthermore, we will see that it can be a result of considering the geodesic distance function on a manifold M that is defined as the graph of a smooth function f .

Let ϕ be the distance function such that

$$\phi(x, y) = \min_{\gamma \subset M} \int_{\gamma} ds,$$

and γ connects the point (x, y) with the set $\Gamma \subset M$. The minimizing curve is called the geodesic, and ϕ the distance function to Γ on M . Moreover, ϕ solves

$$\|P_{\nabla\psi}\nabla\phi\|^2 = 1, \quad \phi|_{\Gamma} = 0, \quad (4)$$

where $\psi(x, y, z) = f(x, y) - z$, and the projection operator [3]

$$P_{\nabla\psi} = I - \frac{\nabla\psi \otimes \nabla\psi}{\|\nabla\psi\|^2},$$

which projects a vector onto a plane whose normal is parallel to $\nabla\psi$. Using the fact that $P_{\nabla\psi}$ is a projection operator, a simple calculation shows that

$$\|P_{\nabla\psi}\nabla\phi\|^2 = \left(1 - \frac{f_x^2}{f_x^2 + f_y^2 + 1}\right)\phi_x^2 + \left(1 - \frac{f_y^2}{f_x^2 + f_y^2 + 1}\right)\phi_y^2 - 2\frac{f_x f_y}{f_x^2 + f_y^2 + 1}\phi_x \phi_y. \quad (5)$$

This is clearly of the form of Hamiltonians that we are interested in. We will apply our algorithm to compute the geodesic distance later in this paper.

There are other approaches that are designed to compute distances on manifolds. [10] provided an algorithm to compute the geodesic distance on triangulated manifolds. Barth [2] uses the Discontinuous Galerkin Method to find distance on graphs of functions that are represented by spline functions. In [3], the authors embed the manifold as the zero level set of a Lipschitz continuous function and solve the corresponding time-dependent eikonal equation (4) in the embedding space. As we have mentioned in the previous subsection, the zero level set of the time-dependent eikonal equation at time t_1 is the t_1 -level set of the solution to the stationary eikonal equations (see [12]). In [11], the authors adopted the standard fast marching method to solve the isotropic eikonal equation in a thin band of thickness ϵ , that encloses the manifold M , and proved that the restriction of the solution to M converges to the geodesic distance as ϵ goes to 0. In [20, 21], the authors provide an ordered upwind method to solve a general class of static Hamilton-Jacobi equations. We will comment on their method in a later subsection.

1.3 Osher's Fast Marching Criteria

Since the fast marching method is by now well known, we will not give much detail on its implementation in this paper. In general, this involves a sorting procedure and the solution of

$$H_G(D_-^x \phi_{i,j}, D_+^x \phi_{i,j}, D_-^y \phi_{i,j}, D_+^y \phi_{i,j}) = 1 \quad (6)$$

for ϕ_{ij} in terms of its four neighboring values. More precisely, the heap sort strategy of the fast marching method requires a monotone update sequence. The updated value of a grid node has to be greater than or equal to those of the grid nodes used to form the finite difference stencil. This amounts to the condition

$$pH_p + qH_q \geq 0,$$

which dictates that the solution is non-decreasing along the characteristics. However, if we use one sided upwind finite difference approximations for partial derivatives of ϕ on a Cartesian grid, it is equivalent to demanding that the partial derivatives of ϕ (i.e. p and q) and their corresponding components of the characteristics directions (i.e. dx/dt and dy/dt) have the same sign. Since $dx/dt = H_p$ and $dy/dt = H_q$, we have the stricter Osher's fast marching criterion:

$$pH_p \geq 0, qH_q \geq 0. \quad (7)$$

It does not matter whether the Hamiltonian is convex or not; as long as criterion (7) is satisfied, a simple fast marching algorithm can be applied. But if the criterion is not satisfied, fast marching cannot be applied to the problem on a Cartesian grid. Of course there are Hamiltonians that do not satisfy (7). In the class of Hamiltonians that we consider, as long as $c \neq 0$, it is likely that the values of p and q differ to the extent that the above criterion is no longer satisfied. In light of criterion (7), we have also tried to find directions $\xi(x, y)$ and $\eta(x, y)$ locally in which $\tilde{p}H_{\tilde{p}} \geq 0, \tilde{q}H_{\tilde{q}} \geq 0$. However, if one insists on using Cartesian grids, the implementation of this approach might be a bit hairy. We are interested especially in tackling, over a Cartesian grid, problems where the solution is non-decreasing along characteristics but Osher’s fast marching criterion is not satisfied.

1.4 The Sweeping Idea

Danielsson [6] proposed an algorithm to compute Euclidean distance to a subset of grid points on a two dimensional grid by visiting each grid node in some predefined orders. In [?], Boué and Dupuis suggest a similar “sweeping” approach to solve the steady state equation which, by experience, results in a $\mathcal{O}(N)$ algorithm for the problem at hand. This “sweeping” approach has recently been used in [23] and [26] to compute the distance function to an arbitrary data set in computer vision. In [25], the author provides some theoretical evidence indicating that sweeping converges to an approximate Euclidean distance function, i.e. to an approximate viscosity solution of $|\nabla\phi| = 1$ in $2d$ predetermined iterations. We will talk about these iterations in a later section. Using this “sweeping” approach, the complexity of the algorithms drops from $\mathcal{O}(N \log N)$ in fast marching to $\mathcal{O}(N)$, and the implementation of the algorithms becomes a bit easier than the fast marching method that requires heap sort.

This sweeping idea is best illustrated by solving the eikonal equation in $[0, 1]$:

$$|u_x| = 1, \quad u(0) = u(1) = 0.$$

Let $u_i = u(x_i)$ denote the grid values associated to the uniform grid composed of the gridpoints $0 = x_0 < x_1 < \dots < x_n = 1$. We then solve the discretized nonlinear system

$$\sqrt{\max(\max(D_- u_i, 0)^2, \min(D_+ u_i, 0)^2)} = 1, \quad u_0 = u_n = 0 \quad (8)$$

by our sweeping approach. We set initially, $u_i^{(0)} = \infty, i = 1, \dots, n - 1$. In practice, ∞ can be replaced by some number K which is larger than $\max_{x \in [0, 1]} u$. Let us begin by sweeping from 0 to 1, i.e. we update u_i from $i = 1$ increasing to

$i = n - 1$. This is “equivalent” to following the characteristics emanating from x_0 . Let $u_i^{(1)}$ denote the grid values after this sweep. We then have

$$u_i^{(1)} = \begin{cases} i/n, & \text{if } i = 1, \dots, n-2 \\ 1/n, & \text{if } i = n-1. \end{cases} \quad i/n, \quad i = 1, \dots, n-1.$$

Notice that at $i = n - 1$, we actually use the upwind information from the neighboring right boundary point. Furthermore, notice that $u_i^{(1)}$ already has the correct desired values for $i \leq n/2$ since the sweep goes from left to right, the desired upwind direction for these i . In the second sweep, we update u_i from $i = n - 1$ decreasing to 1, starting with $u_i^{(1)}$. During this sweep, we follow the characteristics emanating from x_n . The use of (8) is essential, since it determines what happens when two characteristics cross each other. It is then not hard to see that after the second sweep,

$$u_i = \begin{cases} i/n, & \text{if } i \leq n/2 \\ (n-i)/n & \text{otherwise.} \end{cases}$$

Notice the correct values at $i \leq n/2$ derived after the first sweep are unchanged and new and correct values for $i > n/2$ are created. In summary, this simple iterative algorithm can be described as: at the k -th iteration, solve

$$\max \left(\max \left(\frac{u_i^{(k)} - u_{i-1}^{(k-1)}}{\Delta x}, 0 \right), \min \left(\frac{u_{i+1}^{(k-1)} - u_i^{(k)}}{\Delta x}, 0 \right) \right) = 1$$

for $u_i^{(k)}$ for each i going from 1 to $n - 1$ in the first iteration ($k = 1$), and from $n - 1$ to 1 for the second iteration ($k = 2$). However, for more complicated equations and boundary conditions, it is not so easy to write down the equivalent explicit solution.

In this paper, we will extend this sweeping approach to a class of H-J equations that cannot be solved by the fast marching algorithm by first deriving a Godunov Hamiltonian.

In [20, 21], the authors proposed a one-pass method that is based on a control-theoretic view point. In principle, they solve the Hamilton-Jacobi-Bellman equation

$$\max_{\mathbf{a}} \nabla u \cdot \mathbf{a} f(\mathbf{a}, \mathbf{x}) = 1, \quad (9)$$

where $\mathbf{p} = (p, q)$, and the function $f(\mathbf{a}, \mathbf{x})$ is the speed of motion. This formula is the second Legendre Transform taken on the sphere, see e.g. [15, 18].

The idea is still to follow the characteristics and update the grid value in a monotone sequence. In a notation similar to the 2 dimensional setting of [20, 21], we let u_o

be the grid value we are updating. To update u_o , they have to look for two other grid values u_r and u_s which are not necessarily on the immediate grid neighbors of u_o . For example, if u_o is the grid value $u_{i,j}$, the immediate neighbors of u_o are then $u_{i+1,j}, u_{i,j+1}, u_{i-1,j}$, and $u_{i,j-1}$. As we indicated in the previous subsection, it is possible that u_o is less than all its immediate neighboring values. We then need to find two other grid values, here denoted as u_s and u_r , to form an upwinding stencil. Then u_o is found by minimizing a nonlinear expression derived from (9), using the values of u_r , u_s and f . The heap sort data structures is used in order to find u_r and u_s , therefore, the complexity is $N \log N$, where N is the total number of grid points. Also, since u_r and u_s may not lie on the immediate neighbors, this algorithm may need a larger region around the initial wavefront to get started.

As one will see in the following section, our proposed method is also based on following the characteristics. To update u_o , our method only uses the immediate neighboring grid values and does not need the heap sort data structure. More importantly, our algorithm follows the characteristics with certain directions simultaneously, in a parallel way, instead of a sequential way as in the fast marching method. The Godunov flux is essential in our algorithm, since it determines what neighboring grid values should be used to update u on a given grid node o . At least in the examples presented, we only need to solve a simple quadratic equation and run some simple tests to determine the value to be updated. This simple procedure is performed in each sweep, and the solution is obtained after a few sweeps. Our code is not much more than what is presented in Section 3.2. We also point out the ease of implementing our proposed algorithm and its extension to more dimensions. This will appear in a sequel paper.

2 A Godunov Flux for Strictly Convex Hamiltonians

By solving the Riemann problem for Hamilton-Jacobi Equations (Godunov's procedure), Bardi and Osher [1] proved rigorously that

$$H_G(p_-, p_+; q_-, q_+) = \text{ext}_{p \in I[p_-, p_+]} \text{ext}_{q \in I[q_-, q_+]} H(p, q) \quad (10)$$

where

$$\text{ext}_{p \in I[a, b]} = \min_{p \in [a, b]} \quad \text{if } a \leq b,$$

$$\text{ext}_{p \in I[a, b]} = \max_{p \in [b, a]} \quad \text{if } a > b,$$

$$H_G(D_-^x \phi_{ij}, D_+^x \phi_{ij}, D_-^y \phi_{ij}, D_+^y \phi_{ij}) = H_G(p_-, p_+; q_-, q_+)$$

and $I[a, b]$ denotes the closed interval bounded by a and b . This is a monotone upwind flux function, which implies convergence. Godunov's scheme (10) for the eikonal equation $\sqrt{\phi_x^2 + \phi_y^2} = 1$ can be derived from the above formula. It is one of the central topics of this paper to derive an explicit formula for the class of strictly convex Hamiltonians in question. Especially, we will demonstrate our numerical methods on $H = \sqrt{a\phi_x^2 + b\phi_y^2 - 2c\phi_x\phi_y}, c^2 < ab$.

Note that in general, if we reverse the order on p and q in our ext-ext decision, the result might be different, although they both give convergent monotone methods. However, in the convex Hamiltonian at hand, the results are order independent.

For convenience, we will also use $H_G(\phi_{i,j}, \phi_{i\pm 1,j}, \phi_{i,j\pm 1})$ to denote the evaluation of our Godunov Hamiltonian $H_G(D_-^x \phi_{ij}, D_+^x \phi_{ij}, D_-^y \phi_{ij}, D_+^y \phi_{ij})$.

2.1 Derivation of the Flux

In order to derive a compact expression that satisfies equation (10), we need to study the extremum of the Hamiltonian on $I_p \times I_q \subset \mathbb{R}^2$, where I_p is a shorthand for $I[p_-, p_+]$.

The extremum may occur on either the critical points of H , or the boundary of $I_p \times I_q$. Let us first look at the partial derivatives of H , i.e. H_p and H_q , and their zeros. Fix a q_0 , the extremum of $H(p, q_0)$ occurs at either the critical point of $H(p, q_0)$ (i.e. where $H_p = 0$) or the boundary of $I[p_-, p_+]$. We denote the critical point by $p_\sigma(q_0)$. Similarly, given p_0 , we obtain the critical point $q_\sigma(p_0)$. For convenience, we shall denote $p_\sigma(q_0)$ by p_σ when q_0 can be determined from the context, and (p_σ, q_σ) is the critical point of H such that $H_p(p_\sigma, q_\sigma) = H_q(p_\sigma, q_\sigma) = 0$. Therefore, we consider separately $H(p_\sigma, q_\sigma)$, $H(p_-, q_\sigma(p_-))$, $H(p_+, q_\sigma(p_+))$, $H(p_\sigma(q_-), q_-)$, $H(p_\sigma(q_+), q_+)$, and $H(p_\pm, q_\pm)$ as possible evaluations of (10).

For fixed p , we have

$$H_G(p, q_-, q_+) = H(p, \text{sgn max}\{(q_- - q_\sigma)^+, (q_+ - q_\sigma)^-\} + q_\sigma), \quad (11)$$

where

$$\begin{aligned} \text{sgn max}(x, y) &= x^+ \quad \text{if } \max\{x^+, y^-\} = x^+ \\ \text{sgn max}(x, y) &= -y^- \quad \text{if } \max\{x^+, y^-\} = y^-. \end{aligned}$$

The expression for fixed q is a direct analogy to (11). It is easy to see that $H_G(\cdot, \cdot; q_-, q_+)$ is increasing in q_- and decreasing in q_+ . By symmetry, $H_G(p_-, p_+; \cdot, \cdot)$ is increasing in p_- and decreasing in p_+ .

Details of the derivation of the above expression are provided in the appendix.

The following proposition will be of use in analyzing this introduced Godunov flux.

Proposition 1 *If $H_{pp} > 0, H_{qq} > 0$ and $pH_p \geq 0, qH_q \geq 0$, and $p_\sigma(0) = q_\sigma(0) = 0$, then $p_\sigma(q) \equiv 0 \forall q$, and $q_\sigma(p) \equiv 0 \forall p$.*

Proof. $p_\sigma(q)$ by definition, is the zero of $H_p(p_\sigma(q), q) = 0$. We will write p_σ in place of $p_\sigma(q)$ for brevity. This Proposition is then proved by simple manipulation of the definitions.

$$\begin{aligned}
\frac{d}{dq} p_\sigma H(p_\sigma, q) &= p'_\sigma H_p(p_\sigma, q) + p_\sigma (H_{pp}(p_\sigma, q) p'_\sigma + H_{pq}(p_\sigma, q)) \\
&= p_\sigma p'_\sigma H_{pp}(p_\sigma, q) + \frac{\partial}{\partial q} H_p(p_\sigma, q) \\
&= p_\sigma p'_\sigma H_{pp}(p_\sigma, q) \\
&= 0
\end{aligned}$$

The hypothesis $H_{pp} > 0$ implies that

$$p_\sigma(q) = 0 \quad \forall q \quad \text{or} \quad p'_\sigma(q) = 0 \quad \forall q.$$

Again, by the hypothesis that $p_\sigma(0) = q_\sigma(0) = 0$, we can conclude that $p_\sigma(q) \equiv 0 \forall q$.

Similarly, $q_\sigma(p) \equiv 0 \forall p$.

Q.E.D.

Notice that if the Hamiltonian is $\sqrt{p^2 + q^2}$, our upwinding expression in (11) is identical to the conventional expression $\max(p_-^+, p_+^-)$. (In this case, the sign of the second argument does not matter since we are really evaluating the square product of it in the eikonal equation.) In fact, we have the following Corollary, which is a direct consequence of Proposition 1.

Corollary 1 *If $H_{pp} > 0, H_{qq} > 0, pH_p \geq 0, qH_q \geq 0, p_\sigma(0) = q_\sigma(0) = 0$, and $H(p, q) = H(|p|, |q|)$, then the Godunov flux can be simplified to*

$$H_G(p_-, p_+; q_-, q_+) = H(\max\{p_-^+, p_+^-\}, \max\{q_-^+, q_+^-\}).$$

3 The Sweeping Algorithms

We will use the model equation (1) as a concrete example for the exposition of our algorithm. We stress here again that the scheme described below is valid for a general class of convex, homogeneous Hamilton-Jacobi Equations.

From the assumption that the solution is nondecreasing along the characteristics, i.e.

$$pH_p + qH_q \geq 0,$$

we can easily deduce that the solution is non-decreasing at least in either the x - or y - direction; i.e. either $pH_p \geq 0$ or $qH_q \geq 0$. Since we approximate the derivatives $\phi_x(x_{i,j})$ by finite differencing using the neighbors of $\phi_{i,j}$, the above monotonicity property translates to the following requirement in the solution $\phi_{i,j}$:

Definition: Let $\phi_{i,j}$ be the solution of $H_G(\phi, \phi_{i\pm 1,j}, \phi_{i,j\pm j}) = r_{i,j}$. We say that ϕ satisfies the monotonicity requirement if

$$\phi_{i,j} \geq \min\{\phi_{i\pm 1,j}, \phi_{i,j\pm 1}\}.$$

3.1 Derivation of the Scheme

Without loss of generality, we assume that $r(x, y) = 1$. Let us re-examine the equation to be solved:

$$H(p, q) = 1 \tag{12}$$

where

$$H : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}.$$

Equation (12) dictates a level set relation; namely, the solution is the 1-level set of H in the p - q plane (denoted here as Λ). Correspondingly, the solutions of the Hamilton-Jacobi equation with the Godunov Hamiltonian

$$H_G(p_+, p_-; q_+, q_-) = \text{ext}_{p \in I[p_-, p_+]} \text{ext}_{q \in I[q_-, q_+]} H(p, q) = r \tag{13}$$

satisfy the following two properties:

- they are the intersections of Λ and the set $I[p_-, p_+] \times I[q_-, q_+]$;
- they are either the critical points of H or the boundary points of the set $I[p_-, p_+] \times I[q_-, q_+]$.

Figure 1: The 1-level set of H and the box $I[p-,p+] \times I[q-,q+]$

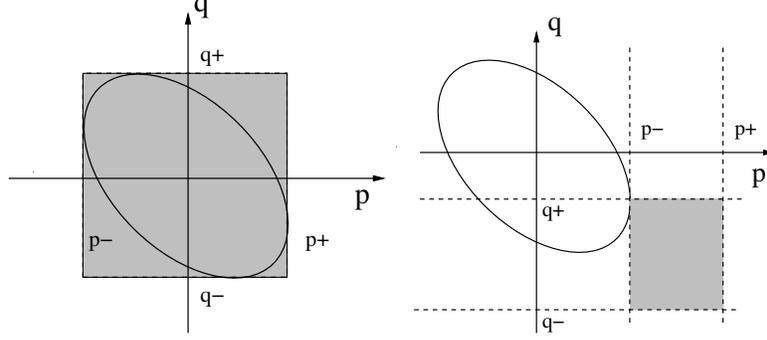


Figure 1 demonstrates two possible configurations of the intervals. So what our algorithm should do is find a suitable value of ϕ on each grid node so that the divided forward and backward differences of ϕ at that grid node satisfy equation (13).

Suppose we are on the grid node (i, j) , and it is determined that

$$H_G(p_+, p_-; q_+, q_-) = H(p_-, q_+) = 1.$$

Correspondingly, for our model equation (1), we have to solve the following quadratic equation:

$$a\left(\frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}\right)^2 + b\left(\frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y}\right)^2 - 2c\left(\frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}\right)\left(\frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y}\right) = 1. \quad (14)$$

The solution $\phi_{i,j}$ has to satisfy not only the above equation, but it ultimately has to be a solution to equation (13) given its four neighbors $\phi_{i-1,j}$, $\phi_{i+1,j}$, $\phi_{i,j-1}$, and $\phi_{i,j+1}$. The subfigure on the right in figure 1 shows one such possible configuration; i.e.

$$\frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} < \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} \quad \text{and} \quad \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y} < \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y},$$

such that

$$\text{ext}_{p \in I[p-, p+]} \text{ext}_{q \in I[q-, q+]} H(p, q) = \min_{p \in I[p-, p+]} \min_{q \in I[q-, q+]} H(p, q) = 1.$$

One can, of course, implement a tree of all the probable cases from the complete listing of that of the Godunov Hamiltonian (10). However, we have a more straightforward approach that utilizes the compact expressions for the Godunov Hamiltonian (11) that we obtained from the previous section.

Instead, we solve the equation with the following reduced formulas for the original Godunov Hamiltonian:

$$H_G(p_+, p_-; q_+, q_-) = \text{ext}_{q \in I[q_-, q_+]} H(p_-, q), \quad (15)$$

$$H_G(p_+, p_-; q_+, q_-) = \text{ext}_{q \in I[q_-, q_+]} H(p_+, q), \quad (16)$$

$$H_G(p_+, p_-; q_+, q_-) = \text{ext}_{p \in I[p_-, p_+]} H(p, q_-), \quad (17)$$

$$H_G(p_+, p_-; q_+, q_-) = \text{ext}_{p \in I[p_-, p_+]} H(p, q_+), \quad (18)$$

$$H_G(p_+, p_-; q_+, q_-) = H(p_\sigma, q_\sigma) \quad (19)$$

For example, in the first case, the flux is equivalent to

$$H(p_-, \text{sgn max}\{(q_- - q_\sigma)^+, (q_+ - q_\sigma)^-\} + q_\sigma) = 1.$$

The possible evaluations of $\text{sgn max}\{(q_- - q_\sigma)^+, (q_+ - q_\sigma)^-\} + q_\sigma$ are q_- , q_+ , $q_\sigma(p_-)$, and 0. We thus end up solving the H-J equation with all possible arguments for the Hamiltonian.

Suppose we solve algebraically $H(p_-, q_+) = 1$ for $\phi_{i,j}$ and call the solution ϕ^{can} . We then compute the divided differences p_\pm and q_\pm using this ϕ^{can} in place of $\phi_{i,j}$. We call ϕ^{can} valid if both

$$H(p_-, \text{sgn max}\{(q_- - q_\sigma)^+, (q_+ - q_\sigma)^-\} + q_\sigma) = 1,$$

$$H(\text{sgn max}\{(p_- - p_\sigma)^+, (p_+ - p_\sigma)^-\} + p_\sigma, q_+) = 1,$$

and ϕ^{can} satisfies the monotonicity requirement (Definition 1).

Finally, we set $\phi_{i,j}$ to be the minimum of those in the set of all valid candidate solutions ϕ^{can} obtained from using all the possible combinations of the arguments of H . This is motivated by the first arrival time interpretation of the function ϕ .

In essence, we are solving for the central value in the Godunov Hamiltonian in terms of its four neighbors. It's well known and easy to show that any monotone Hamiltonian, let alone Godunov's, is a monotone function of this value. For these Hamiltonians, this value goes from $-\infty$ to $+\infty$. Thus there is always a unique solution.

Definition: (Sweeping iteration)

A compact way of writing this sweeping iterations in C/C++ is:

```

for (s1=-1; s1<=1; s1+=2)
for (s2=-1; s2<=1; s2+=2)
for (i=(s1<0?nx:0); (s1<0?i>=0:i<=nx); i+=s1)
for (j=(s2<0?ny:0); (s2<0?j>=0:j<=ny); j+=s2)
    update  $\phi_{i,j}$ 

```

3.2 The Algorithm

For the brevity of the algorithm, we define respectively

$$h_{G1}(p, q_-, q_+) := \text{sgn} \max\{(q_- - q_\sigma(p))^+, (q_+ - q_\sigma(p))^- \} + q_\sigma(p),$$

$$h_{G2}(p_-, p_+, q) := \text{sgn} \max\{(p_- - p_\sigma(q))^+, (p_+ - p_\sigma(q))^- \} + p_\sigma(q).$$

where $q_\sigma(p) = pc/b$ and $p_\sigma(q) = qc/a$.

Algorithm: We assume that $\phi(i, j)$ is given the exact values in a small neighborhood of Γ . We denote this neighborhood $\text{Nbd}(\Gamma)$. We initialize ϕ by setting $\phi(i, j) = \phi_{i,j}^{(0)}$ to ∞^1 . We begin by computing $\phi_{i,j}^{(n)}$, for $n = 1$.

Do the following steps while $\|\phi^{(n)} - \phi^{(n-1)}\| > \delta$: ($\delta > 0$ is the given tolerance.)

1. **For each grid point (i, j) visited in the sweeping iteration, if $x_{i,j} \neq \text{Nbd}(\Gamma)$, do the following:**

- (a) For $(s_x, s_y) = (-1, 1), (-1, -1), (1, -1)$, and $(1, 1)$

- i. Solve

$$H\left(\frac{s_x \cdot (\phi_{tmp}(s_x, s_y) - \phi^{(n)}(i - s_x, j))}{dx}, \frac{s_y \cdot (\phi_{tmp}(s_x, s_y) - \phi^{(n)}(i, j - s_y))}{dy}\right) = r(i, j)$$

for $\phi_{tmp}(s_x, s_y)$.

- ii. Let

$$p(s_x, s_y) = \frac{s_x \cdot (\phi_{tmp}(s_x, s_y) - \phi^{(n)}(i - s_x, j))}{dx},$$

and

$$q(s_x, s_y) = \frac{s_y \cdot (\phi_{tmp}(s_x, s_y) - \phi^{(n)}(i, j - s_y))}{dy}.$$

¹Notice that we only need to use a large value in actual implementation.

iii. Let $T_{G1}(s_x, s_y)$ be the logical evaluation of the equality:

$$H(p(s_x, s_y), h_{G1}(p(s_x, s_y), q(s_x, 1), q(s_x, -1))) = r(i, j),$$

and $T_{G2}(s_x, s_y)$ be that of

$$H(h_{G2}(p(1, s_y), p(-1, s_y), q(s_x, s_y)), q(s_x, s_y)) = r(i, j).$$

iv. Let $M(s_x, s_y) = \phi_{tmp}(s_x, s_y) - \min(\phi^{(n)}(i - s_x, j), \phi^{(n)}(i, j - s_y))$.

v. If $T_{G1}(s_x, s_y)$, $T_{G2}(s_x, s_y)$ are true, and $M(s_x, s_y) \geq 0$, add $\phi_{tmp}(s_x, s_y)$ to the list `phi_candidate`.

(b) For $(s_x, s_y) = (1, 0), (-1, 0)$,

i. Solve

$$H\left(\frac{s_x \cdot (\phi_{tmp}(s_x, 0) - \phi^{(n)}(i - s_x, j))}{dx}, \frac{s_x \cdot (\phi_{tmp}(s_x, 0) - \phi^{(n)}(i - s_x, j)) \frac{c}{b}}{dx}\right) = r(i, j),$$

for $\phi_{tmp}(s_x, s_y)$

ii. Compute $p(s_x, s_y)$, and $q(s_x, s_y)$ following the definition.

iii. Evaluate $T_{G1}(s_x, s_y)$.

iv. If $T_{G1}(s_x, s_y)$ is true and $M(s_x, s_y) \geq 0$, add $\phi_{tmp}(s_x, s_y)$ to the list `phi_candidate`.

(c) For $(s_x, s_y) = (0, 1)$ and $(0, -1)$,

i. Solve

$$H\left(\frac{s_y \cdot (\phi_{tmp}(0, s_y) - \phi^{(n)}(i, j - s_y)) \frac{c}{a}}{dy}, \frac{s_y \cdot (\phi_{tmp}(0, s_y) - \phi^{(n)}(i, j - s_y))}{dy}\right) = r(i, j),$$

for $\phi_{tmp}(s_x, s_y)$.

ii. Compute $p(s_x, s_y)$, and $q(s_x, s_y)$ following the definition.

iii. Evaluate $T_{G2}(s_x, s_y)$.

iv. If $T_{G2}(s_x, s_y)$ is true and $M(s_x, s_y) \geq 0$, add $\phi_{tmp}(s_x, s_y)$ to the list `phi_candidate`.

(d) Let ϕ_{min} be the minimum element of `phi_candidate`.

$$\phi^{(n)}(i, j) = \min(\phi^{(n)}(i, j), \phi_{min}).$$

(e) Clear `phi_candidate`.

2. Set $n = n + 1$, go back to 1.

As described in the previous section, we have to solve the H-J equation with all possible arguments for the Hamiltonian and take the minimum of those in the set of all valid candidate solutions. The possible arguments of the Hamiltonian consists of the forward/backward differences of ϕ and the critical points centered at each grid node. In the above algorithm, this set of all possible arguments is indexed by $\{-1, 0, 1\}^2$. Therefore, by $X(-1, 1)$, we denote the quantity X that is computed from using $H(p_-, q_+)$. The number 0 encodes the cases of critical points. For example, $\phi_{tmp}(-1, 1)$ denotes the roots of the quadratic equation formed by $H(p_-, q_+) = r$; $\phi_{tmp}(1, 0)$ denotes that of $H(p_+, q_\sigma(p_+))$.

We remark that in the case of $c = 0$, our algorithm is equivalent to what is used in the fast marching method under the Rouy-Tourin formula (10). Secondly, in our numerical implementation, we put a threshold value in the evaluations T_{G1} and T_{G2} due do numerical accuracy reasons.

4 Examples

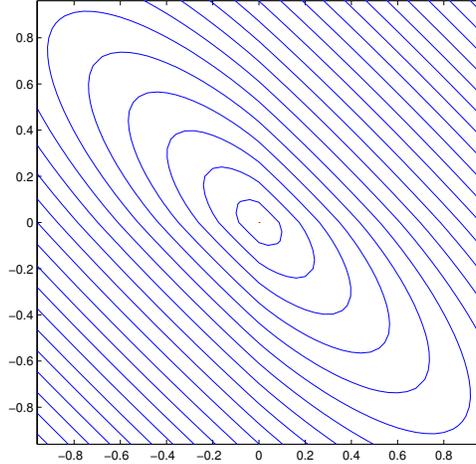
Proposition 1 and Corollary 1 show the equivalence of the Godunov flux derived in this paper to the one commonly used in the fast marching applications. The use of this sweeping approach with the Godunov flux (10) has been reported in [23, 25] for eikonal equations, we will not repeat the examples in this paper. Instead, we present results of our algorithm applied to our model equation.

4.1 Quadratic Hamiltonians $\sqrt{ap^2 + bq^2 - 2cpq}$, $ab > c^2$, $a, b > 0$

In each of the following examples, we compute the difference in the approximations in each successive iteration, i.e. $\|\phi^{n+1} - \phi^n\|_{L_1}$, and say that the iterations have converged if this distance is less than $\varepsilon\Delta x$, where $\varepsilon > 0$, and Δx is the grid size. In the examples presented in this paper, we simply set the threshold to be 10^{-10} . Notice also that the set Γ , on which $\phi = 0$, is depicted in red in the contour plots in the following examples.

We started out by testing our algorithm on constant coefficient cases. In the case of $a = b$, $c = 0$, we have solutions that match the fast marching solutions. Figure 2 shows a result of a computation of the anisotropic case where $a = b = 1$, $c = 0.9$. This is our first example in which the fast marching method is not applicable.

Figure 2: A sweeping result after 2 sweeping iteration on a 50x50 grid. The initial boundary is a single point in the center. $a = 1.0$, $b = 1.0$, $c = 0.9$.



Next we apply the sweeping algorithm directly to cases in which the coefficients of the quadratic Hamiltonian or the right hand sides are not constant. Figure 3 shows a computational result on a constant coefficient isotropic Hamiltonian and rather oscillatory forcing function. The rectangle in the middle is the set Γ . Figure 4 shows a computational result for a very anisotropic case. We notice that the number of iterations needed for convergence seems to depend on the anisotropy of the Hamiltonian and also on how oscillatory the forcing term is. Figure 5,6,7 show results obtained from variable coefficient Hamiltonians with constant and variable forcing function $r(x, y)$.

4.2 Examples of Distance on Manifolds

We now apply our sweeping algorithm to compute the geodesic distance on manifolds that are the graphs of certain functions. Given a function $f(x, y)$, with graph $z = f(x, y)$, we compute the coefficients $a(x, y)$, $b(x, y)$ and $c(x, y)$ according to (5), and apply our algorithm directly to the corresponding H-J equation. We first test the algorithm on a half sphere with radius one. Figures 8 and 9 show the equidistance lines to one and two seed points respectively. Figures 10, 11 and 12 show similar computation results applied to somewhat more oscillatory manifolds. As we expected, more sweeping iterations are required for convergence.

Figure 3: $a = 1, b = 1, c = 0$, with a more oscillatory $r(x) = 2.1 - \cos(4\pi^2 xy)$, on a 200x200 grid, convergence is reached in 7 sweeping iterations. The subplot on the top is the contour of the solution started with the square in the center. On the bottom is the graph of $r(x)$. Level curves with step 0.02 are plotted.

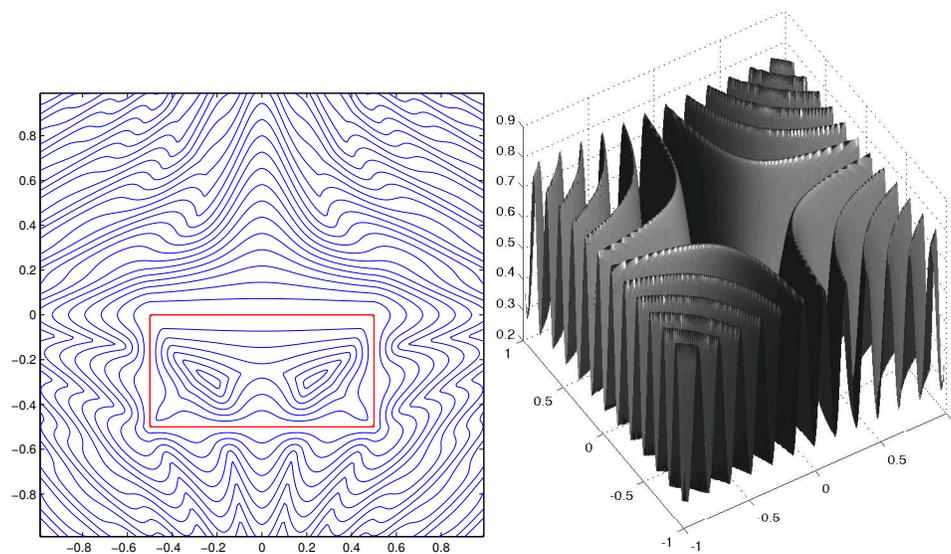


Figure 4: (A very degenerate case) $a = 0.375$, $b = 0.25$, $c = 0.29$, with a more oscillatory $r(x) = (2.1 - \cos(4\pi^2 xy))/4.0$, on a 100×100 grid. Notice that in this case, $ab = 0.0938$ is barely greater than $c^2 = 0.0841$. The contour of the solution is plotted. The convergence is reached at 43 sweeping iterations.

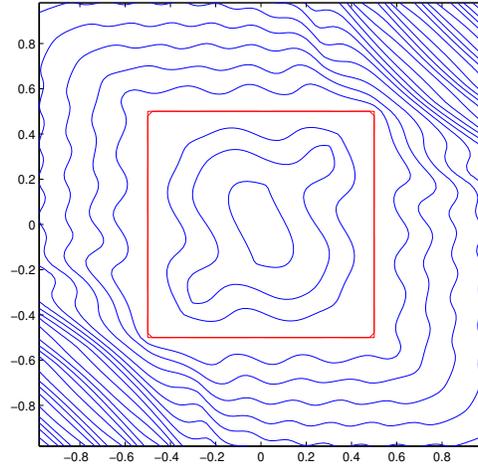


Figure 5: $a = 1$, $b = 1$, $c(x, y) = 0.9 \sin(5\pi x)$, and $r(x, y) = 1$. 50×50 grid. Convergence after 10 iterations.

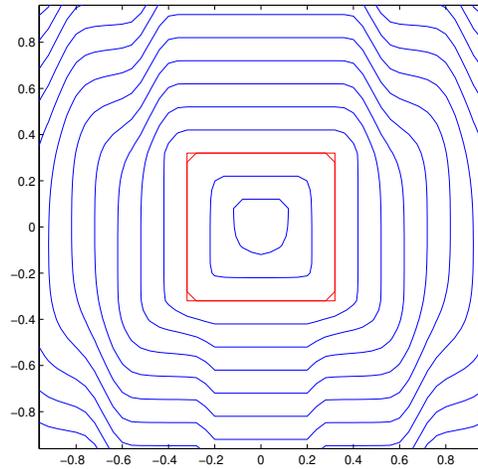


Figure 6: $a = 1.5 + \sin(5\pi x)$, $b = 1$, $c = -0.6$. 50x50 grid. Convergence after 10 iterations.

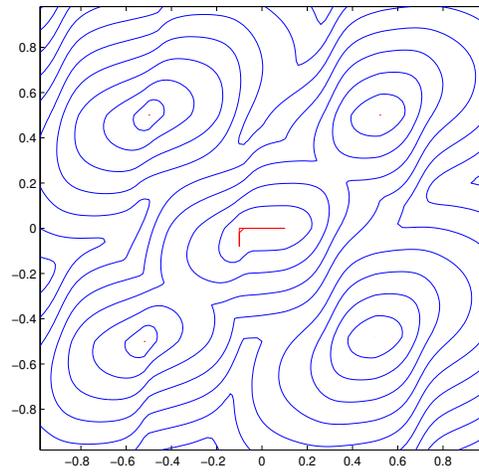


Figure 7: $a = 1.5 + \sin(5\pi x)$, $b = 1$, $c = -0.6$, and $r(x, y) = 2.1 + \cos(4\pi xy)$. 100x100 grid. Convergence after 10 iterations.

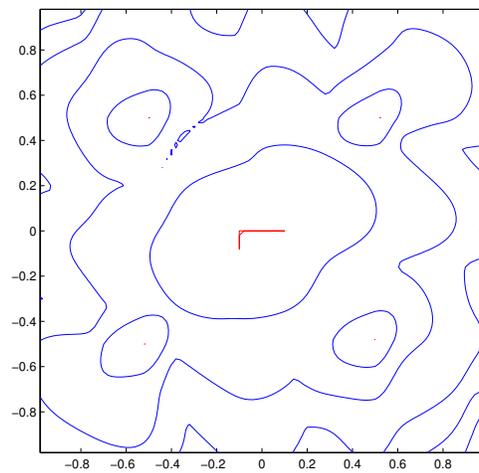


Figure 8: This is an example of the distance on a half sphere. The sweeping algorithm applied to the graph of $f(x, y) = \sqrt{1.0 - (x^2 + y^2)}$, with $\phi(0, 0) = 0$ as boundary condition on a 100x100 grid.

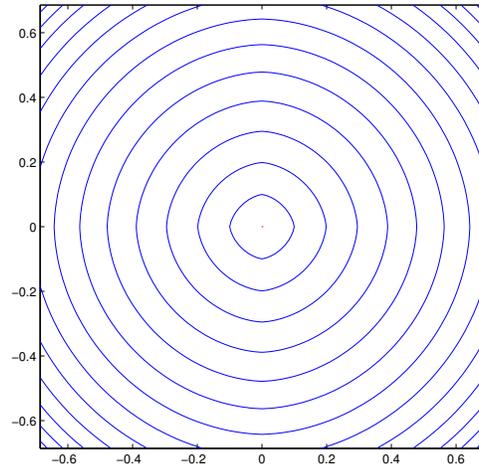


Figure 9: This is an example of the distance on a half sphere. The sweeping algorithm applied to the graph of $f(x, y) = \sqrt{1.0 - (x^2 + y^2)}$, with two seed points. The convergence is reached after 2 sweeping iterations.

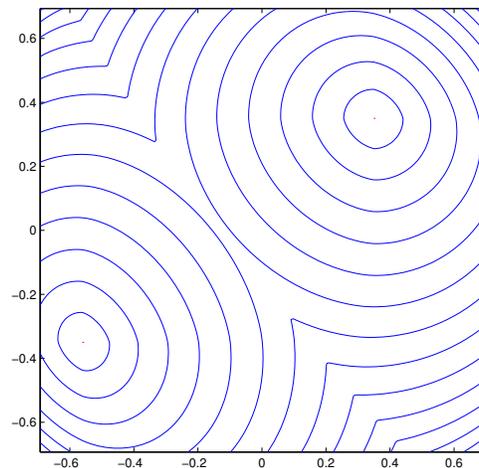


Figure 10: The distance contour from the seed point $(0, 0)$ on the graph of $f(x, y) = \cos(2\pi x) \sin(2\pi y)$. 100x100 grid, convergence after 9 iterations.

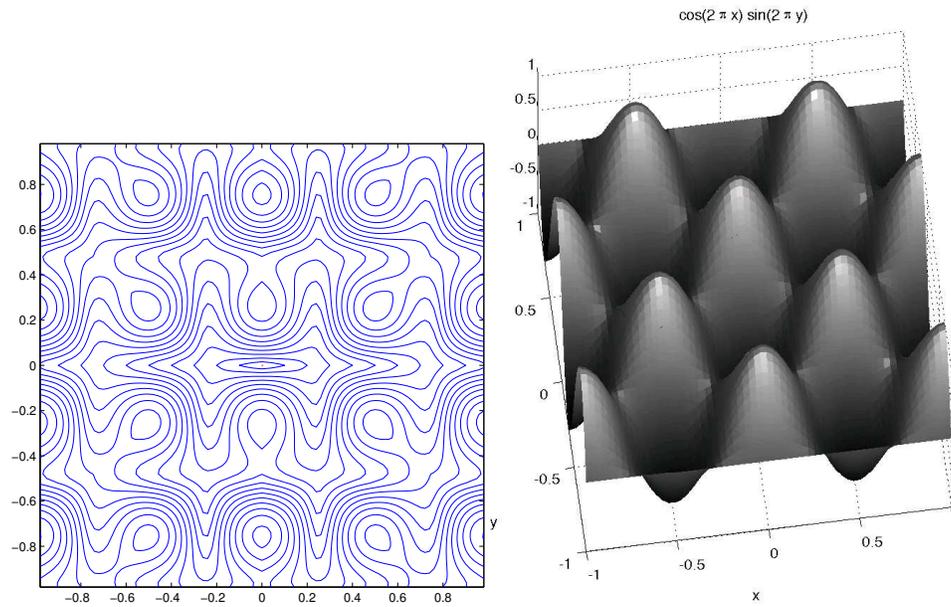
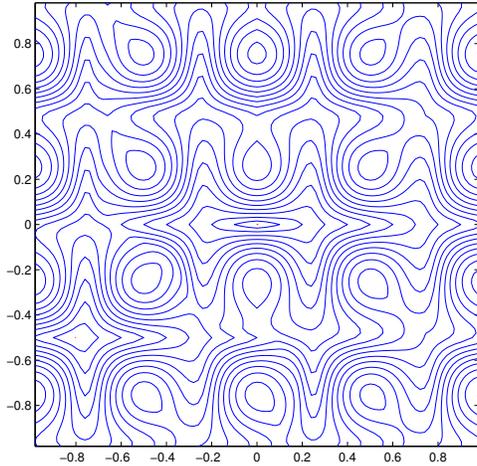


Figure 11: The distance contour from the seed point $(0, 0)$ and $(-0.8, -0.5)$ on the graph of $f(x, y) = \cos(2\pi x) \sin(2\pi y)$. 100x100 grid, convergence after 11 iterations.



4.3 Grid Effects

We first perform a rotation of the coordinate system. We represent this by

$$(x, y) \mapsto (\tilde{x}, \tilde{y}),$$

and let

$$(a, b, c) \mapsto (\tilde{a}, \tilde{b}, \tilde{c})$$

To study the grid effects of our sweeping algorithm, we set $u = 0$ on a rotated square whose sides do not align with the grid lines. Comparing the results, shown in figure 13, we see that the second picture, concentrating especially on the diamond shaped contour in the middle, indeed shows grid effects compared to the first picture. However, with further grid refinement, as shown in the third picture, grid effects become unnoticeable and the solution from our sweeping algorithm accurately approximates the exact solution.

Figure 12: The distance contour from the seed point $(0, 0)$ on the graph of $f(x, y) = \cos(2\pi x - \pi) \sin(2\pi y - \pi/2)$. 100x100 grid, convergence after 9 iterations.

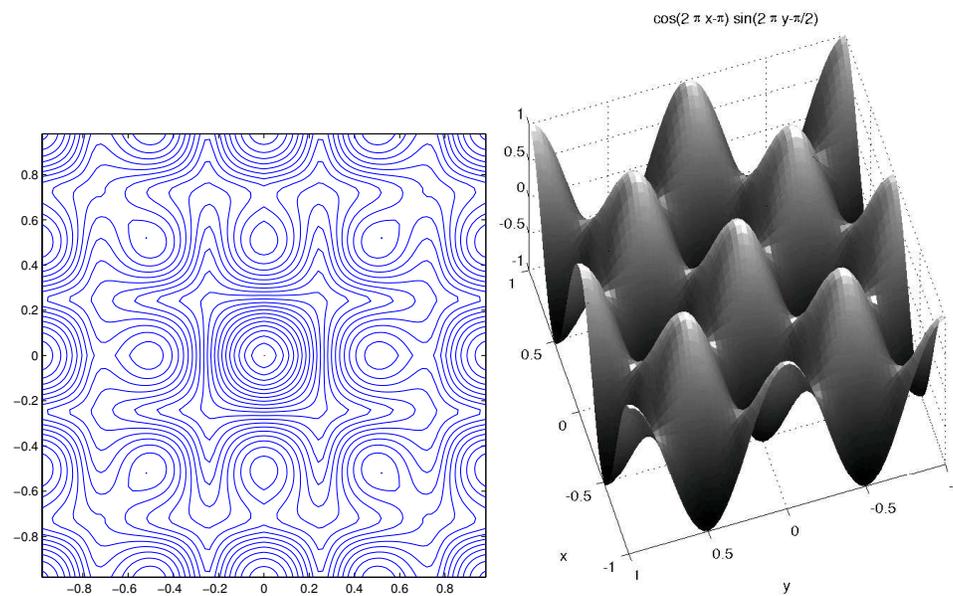


Figure 13: Anisotropic case with a point source at $(0, 0)$. $a = 1, b = 1, c = 0.9$ and $\tilde{a} = 1.70365, \tilde{b} = 0.296352$, and $\tilde{c} = -0.561141$. 50×50 and 100×100 . Convergence after 2 iterations.

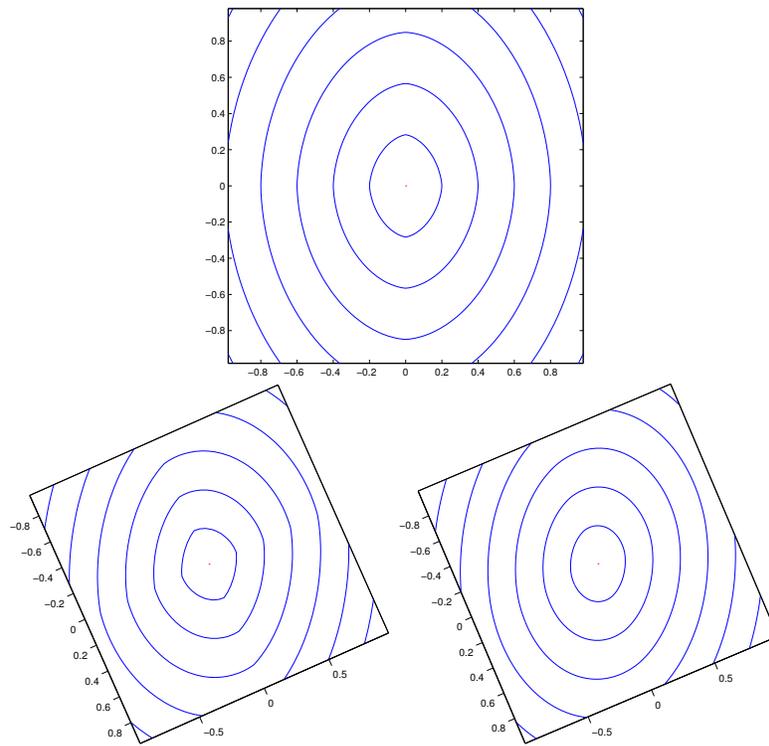


Table 1: Comparison of the time-marching solution and the sweeping solution to the example shown in figure 12.

	$dx = 2/50$	$2/100$	$2/200$	$2/400$	$2/800$
$\ \phi - \tilde{\phi}\ _{L_1}$	2.85423	1.83377	1.04008	0.56206	0.295738
$\ \phi - \tilde{\phi}\ _{\infty}$	1.03825	0.708986	0.436469	0.246439	0.133858

4.4 Comparison with the Time Marching Solutions

We use the first order RK-Lax-Friedrichs method [17] to discretize the following equation and march to steady state:

$$\tilde{\phi}_t + \text{sgn}(\phi(x, y))(H(x, y, \tilde{\phi}_x, \tilde{\phi}_y) - r(x, y)) = 0. \quad (20)$$

where $\tilde{\phi}(x, y, t = 0) = \phi(x, y) = 0$ for $(x, y) \in \Gamma$, and ϕ is the solution obtained from the sweeping algorithm.

We remark that solving equation (20) is by no means a practical method for solving the steady state equation. Thousands of iterations are required for steady state, even if we take ϕ as the initial Cauchy data. We only use it to verify the validity of our algorithm. Secondly, the solutions of equation (20) suffer from excessive smearing due to the numerical viscosity introduced by the Lax-Friedrichs method. As a consequence, $\tilde{\phi}$ does not match well with ϕ on coarse grids. The reader can compare figure 14 with figure 12, for example. However, we do see that $\|\phi - \tilde{\phi}\|$ decreases with the refinement of the grid size. See table 1 and figure 14. We remark that higher order approximation schemes such as RK3-WENO5 will greatly reduce the numerical viscosity. The reader is referred to [17]. Our purpose here is only to show that the sweeping approximations converge to the viscosity solution.

4.5 Numerical Convergence

Since we can easily compute the geodesic distance on a sphere, we will use it as an example to show numerical convergence of our algorithm. A distance contour plot is shown in figure 8. Table 2 shows a numerical convergence of order 1. We have also noticed that the number of iterations, that is needed for the L_1 difference of the approximations in each successive iteration to decrease below the given tolerance, seem to be bounded independently of the grid size. This number seems to depend on the anisotropy (c^2/ab), the forcing function r , and the configuration of the interface Γ .

Figure 14: Steady state of the time marching on a 100x100 grid and 800x800 grid.

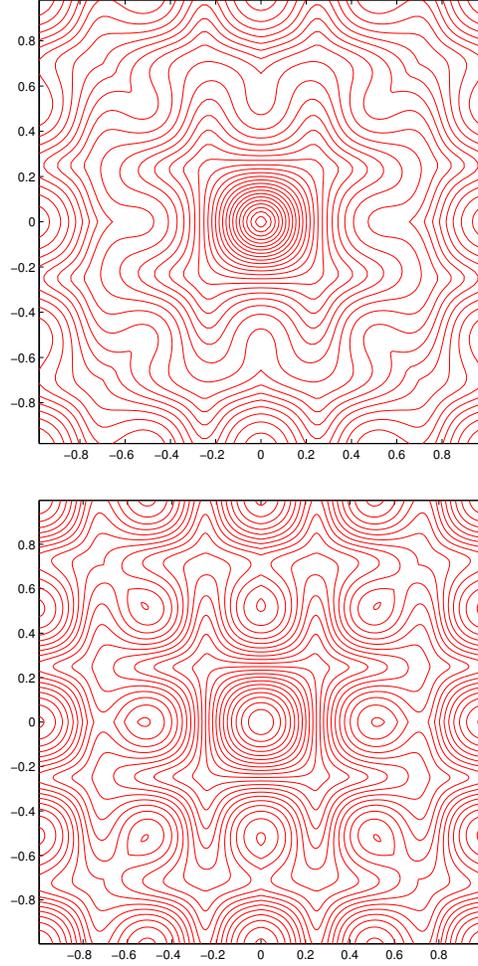


Table 2: A numerical convergence study of the sweeping algorithm applied to the graph of $f(x, y) = \sqrt{1.0 - (x^2 + y^2)}$, with $\phi(0, 0) = 0$ as boundary condition on the domain $[-0.7, 0.7] \times [-0.7, 0.7]$.

	$dx = 1.4/200$	$1.4/400$	$1.4/800$	$1.4/1600$
$\ \phi - \tilde{\phi}\ _{L_1}$	0.0138803	0.0079927	0.00453004	0.00253513
rate		0.796	0.819	0.84

5 Conclusion

In this article, we have studied a fast method for solving a class of time independent Hamilton-Jacobi equations with Dirichlet boundary conditions. The Hamiltonians of interest are homogeneous and convex. This fast method combines the idea of tracing the characteristics with Godunov construction and Gauss-Seidel iterations with smart choices of different updating sequences. In particular, we discussed some important properties of the Hamiltonian $H = \sqrt{ap^2 + bq^2 - 2cpq}$, $c^2 < ab$, and the corresponding H-J equations. By the simple structure of the convexity, we derived a compact expression for the Godunov Hamiltonian that involves taking extrema of the Hamiltonian in relation to the evaluations of the derivatives of the solution. With our compact Godunov flux, the complexity of evaluating the Godunov Hamiltonian is reduced to only eight cases in two space dimensions. We then incorporated the expression into a simple Gauss-Seidel type iteration procedure. We have produced some computational results using this algorithm. In particular, we have applied our algorithm to compute geodesic distances on graphs of functions. This is of some importance since people are interested in finding the geodesics on terrain-like manifolds.

We also remark that this Godunov-flux sweeping approach can be extended to higher dimensional cases. We are currently preparing another paper on this subject.

Our experience shows that the number of iterations needed depends on the amount of anisotropy and the nature of the forcing function. Under normal non-degenerate circumstances, experience shows a $\mathcal{O}(N)$ complexity for convergence, where N is the number of grid points. Recently, in [25], the author provided some theoretical evidence on the bound of the number of iterations for isotropic, homogeneous eikonal equations. This points out a future research direction of bounding the number of sweeping iterations needed for convergence in relation to the anisotropy.

6 Appendix

6.1 Derivation of the Flux for Homogeneous Convex Hamiltonians

To obtain the formula used earlier in this paper, we simply verify its equivalence to the following cases, which rely only on the convexity of H :

$p_- < p_+$, and $q_- < q_+$:

$$H_G = \min_{p \in [p_-, p_+]} \min_{q \in [q_-, q_+]} H(p, q).$$

- if $q_\sigma \in [q_-, q_+]$,
 - $p_\sigma < p_- < p_+$, $H(p_-, q_\sigma)$
 - $p_- < p_+ < p_\sigma$, $H(p_+, q_\sigma)$
 - $p_- < p_\sigma < p_+$, $H(p_\sigma, q_\sigma)$
- if $q_\sigma < q_-$,
 - $p_\sigma < p_- < p_+$, $H(p_-, q_-)$
 - $p_- < p_+ < p_\sigma$, $H(p_+, q_-)$
 - $p_- < p_\sigma < p_+$, $H(p_\sigma, q_-)$
- if $q_\sigma > q_+$,
 - $p_\sigma < p_- < p_+$, $H(p_-, q_+)$
 - $p_- < p_+ < p_\sigma$, $H(p_+, q_+)$
 - $p_- < p_\sigma < p_+$, $H(p_\sigma, q_+)$

$p_- < p_+$, and $q_- > q_+$:

$$H_G = \min_{p \in [p_-, p_+]} \max_{q \in [q_+, q_-]} H(p, q) = \min_{p \in [p_-, p_+]} \max\{H(p, q_-), H(p, q_+)\}.$$

- if $q_\sigma < q_+$,
 - $p_\sigma < p_- < p_+$, $H(p_-, q_-)$
 - $p_- < p_+ < p_\sigma$, $H(p_+, q_-)$
 - $p_- < p_\sigma < p_+$, $H(p_\sigma, q_-)$
- if $q_\sigma > q_-$,
 - $p_\sigma < p_- < p_+$, $H(p_-, q_+)$
 - $p_- < p_+ < p_\sigma$, $H(p_+, q_+)$
 - $p_- < p_\sigma < p_+$, $H(p_\sigma, q_+)$
- if $q_+ < q_\sigma < q_-$

- $(q_\sigma - q_+) > (q_- - q_\sigma), H(\cdot, q_+)$
- $(q_\sigma - q_+) \leq (q_- - q_\sigma), H(\cdot, q_-)$

$p_- > p_+$, and $q_- > q_+$:

$$H_G = \max_{p \in [p_+, p_-]} \max_{q \in [q_+, q_-]} H(p, q)$$

- if $q_\sigma > q_-$,
 - $p_\sigma > p_-, H(p_+, q_+)$.
 - $p_\sigma < p_+, H(p_-, q_+)$.
- if $q_\sigma < q_+$,
 - $p_\sigma > p_-, H(p_+, q_-)$.
 - $p_\sigma < p_+, H(p_-, q_-)$.
- if $q_+ < q_\sigma < q_-$
 - $(q_\sigma - q_+) > (q_- - q_\sigma), H(\cdot, q_+)$
 - $(q_\sigma - q_+) \leq (q_- - q_\sigma), H(\cdot, q_-)$

$p_- > p_+$, and $q_- < q_+$:

$$H_G = \max_{p \in [p_+, p_-]} \min_{q \in [q_-, q_+]} H(p, q)$$

- if $q_\sigma \in [q_-, q_+]$,
 - $p_\sigma > p_-, H(p_+, q_\sigma)$.
 - $p_\sigma < p_+, H(p_-, q_\sigma)$.
- if $q_\sigma < q_-$,
 - $p_\sigma > p_-, H(p_+, q_-)$.
 - $p_\sigma < p_+, H(p_-, q_-)$.
- if $q_\sigma > q_+$,
 - $p_\sigma > p_-, H(p_+, q_+)$.
 - $p_\sigma < p_+, H(p_-, q_+)$.

Acknowledgment

The author YT thanks Paul Burchard and C.Y. Kao for their useful conversations on this topic. The authors thank Prof. Giovanni Russo for his helpful suggestions to improve this paper.

References

- [1] Martino Bardi and Stanley Osher. The nonconvex multi-dimensional Riemann problem for Hamilton-Jacobi equations. *SIAM J Math Anal*, 22(2):344–351, 1991.
- [2] Timothy J. Barth. On the marchability of interior stabilized discontinuous galerkin approximations of the eikonal and related pdes with non-divergence structure. *NASA Technical Report, NAS-01-010, NASA Ames Research Center*, 2001.
- [3] Li-Tien Cheng, Paul Burchard, Barry Merriman, and Stanley Osher. Motion of curves constrained on surfaces using a level set approach. *J. Comput. Phys.*, 175:604–644, 2002.
- [4] M.G. Crandall and P.L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Mathematics of Computation*, 43:1–19, 1984.
- [5] Michael G. Crandall and Pierre-Louis Lions. Viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, 277(1):1–42, 1983.
- [6] Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.
- [7] J. Helmsen, E. Puckett, P. Colella, and M. Dorr. Two new methods for simulating photolithography development in 3d. In *SPIE 2726*, pages 253–261, 1996.
- [8] Guang-Shan Jiang and Danping Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21(6):2126–2143 (electronic), 2000.
- [9] Joseph B. Keller. Geometrical theory of diffraction. *J. Opt. Soc. Amer.*, 52:116–130, 1962.
- [10] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proc. Natl. Acad. Sci. USA*, 95(15):8431–8435 (electronic), 1998.

- [11] Facundo Memoli and Guillermo Sapiro. Fast computation of distance functions and geodesics on implicit surfaces. To appear, *Journal of Computational Physics* (2001).
- [12] Stanley Osher. A level set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations. *SIAM J Math Anal*, 24(5):1145–1152, 1993.
- [13] Stanley Osher and Ronald P. Fedkiw. Level set methods: an overview and some recent results. *J. Comput. Phys.*, 169(2):463–502, 2001.
- [14] Stanley Osher and J. Helmsen. A generalized fast algorithm with applications to ion etching. *In Preparation*.
- [15] Stanley Osher and Barry Merriman. The Wulff shape as the asymptotic limit of a growing crystalline interface. *Asian J. Math.*, 1(3):560–571, 1997.
- [16] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.
- [17] Stanley Osher and Chi-Wang Shu. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28(4):907–922, 1991.
- [18] Danping Peng, Stanley Osher, Barry Merriman, and Hong-Kai Zhao. The geometry of Wulff crystal shapes and its relations with Riemann problems. In *Nonlinear partial differential equations (Evanston, IL, 1998)*, pages 251–303. Amer. Math. Soc., Providence, RI, 1999.
- [19] Elisabeth Rouy and Agnès Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J Num Anal*, 29(3):867–884, 1992.
- [20] J. A. Sethian and A. Vladimirsky. Fast methods for the eikonal and related Hamilton-Jacobi equations on unstructured meshes. *Proc. Natl. Acad. Sci. USA*, 97(11):5699–5703 (electronic), 2000.
- [21] J. A. Sethian and A. Vladimirsky. Ordered upwind methods for static Hamilton-Jacobi equations. *Proc. Natl. Acad. Sci. USA*, 98(20):11069–11074 (electronic), 2001.
- [22] J.A. Sethian. Fast marching level set methods for three dimensional photolithography development. In *SPIE 2726*, pages 261–272, 1996.

- [23] Yen-Hsi Richard Tsai. Rapid and accurate computation of the distance function using grids. *J. Comput. Phys.*, 178(1):175–195, 2002.
- [24] John Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.
- [25] Hong-Kai Zhao. Fast sweeping method for eikonal equations I: Distance function. www.math.uci.edu/~zhao, 2002. Under review, SINUM.
- [26] Hong-Kai Zhao, Stanley Osher, Barry Merriman, and Myungjoo Kang. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Computer Vision and Image Understanding*, 80:295–319, 2000.