

Lax-Friedrichs Sweeping Scheme for Static Hamilton-Jacobi Equations*

Chiu Yen Kao[†], Stanley Osher[‡] and Jianliang Qian[§]

1st August 2003

Abstract

We propose a simple, fast sweeping method based on the Lax-Friedrichs monotone numerical Hamiltonian to approximate the viscosity solution of arbitrary static Hamilton-Jacobi equations in any number of spatial dimensions. By using the Lax-Friedrichs numerical Hamiltonian, we can easily solve for the value of a specific grid point in terms of its neighbors, so that a Gauss-Seidel type nonlinear iterative method can be utilized. Furthermore, by incorporating a group-wise causality principle into the Gauss-Seidel iteration by following a finite group of characteristics, we have an easy-to-implement, sweeping-type, and fast convergent numerical method. However, unlike other methods based on the Godunov numerical Hamiltonian, some computational boundary conditions are needed in the implementation. We give a simple recipe which enforces a version of discrete min-max principle. Some convergence analysis is done for the one-dimensional eikonal equation. Extensive 2-D and 3-D numerical examples illustrate the efficiency and accuracy of the new approach. To our knowledge, this is the first fast numerical method based on discretizing the Hamilton-Jacobi equation directly without assuming convexity and/or homogeneity of the Hamiltonian.

1 Introduction

The Hamilton-Jacobi equation arises in many applications such as geometrical optics, crystal growth, etching, computer vision, obstacle navigation, path planning, photolithography, and seismology. The solutions of these nonlinear differential equations usually develop singularities in their derivatives even with

*Research supported by ONR MURI N00014-02-1-0720

[†]Department of Mathematics, University of California Los Angeles, Los Angeles, California 90095, email:ckao@math.ucla.edu

[‡]Department of Mathematics, University of California Los Angeles, Los Angeles, California 90095, email:sjo@math.ucla.edu

[§]Department of Mathematics, University of California Los Angeles, Los Angeles, California 90095, email:qian@math.ucla.edu

smooth initial conditions. In these cases, the solutions do not satisfy the equations in the classical sense. The notion of viscosity solution was introduced by Crandall and Lions [3] to uniquely determine a solution of a Hamilton Jacobi equation. Numerically, in general, one looks for a consistent, convergent, e.g. monotone scheme to approximate viscosity solutions [19].

In this paper, we focus on static Hamilton-Jacobi equations of the following form:

$$\begin{cases} H(x, \nabla\phi(x)) = R(x) & x \in \Omega \\ \phi(x) = q(x) & x \in \Gamma \end{cases} \quad (1)$$

where H , q , and $R > 0$ are Lipschitz continuous, and Γ is a subset of Ω . Static Hamilton-Jacobi equation as a first order nonlinear PDE appears in many different applications. In the Dynamic Programming approach for infinite horizon optimal control, the value function of the optimized cost functional satisfies so-called Hamilton-Jacobi-Bellman equation, a static equation having a convex Hamiltonian in the gradient variable. In the Dynamic Programming approach for differential games, the value function for the zero-sum game satisfies the so-called Hamilton-Jacobi-Isaacs equation, the resulting Hamiltonian being nonconvex. In the classical high frequency asymptotic for wave propagation in elastic solids, the phase function, a.k.a traveltime function in some applications, satisfies the so-called eikonal equation which is an indispensable element of the family of Hamilton-Jacobi equations. To be more specific, in the isotropic elastic solid, the traveltime satisfies the isotropic eikonal equation $|\nabla\phi| = 1$ which is of quadratic nonlinearity and thus has a convex Hamiltonian. However, in the anisotropic elastic solid, high frequency asymptotic analysis gives rise to three different wave modes: one quasi-longitudinal wave and two quasi-shear waves. The Hamilton-Jacobi equation for quasi-longitudinal wave traveltime is convex in the gradient variable and homogeneous of degree one, but the Hamilton-Jacobi equation for one of the two shear waves is nonconvex in the gradient variable. In the semi-classical limit for Schroedinger equation, the eikonal equation arises as the Planck constant approaches zero. Therefore, it is of fundamental importance to design fast, accurate numerical schemes to solve the resulting static Hamilton-Jacobi equations for the above applications.

The numerical methods for this type of equations can be roughly divided into three categories. The first class of methods are those that involve the use of a fast marching method and heap-sort data structure. These methods [21] [17] [5] [18] are based on the monotonicity of the solution along the characteristics. The solutions are constructed with a variation of the classical Dijkstra algorithm. The complexity is $O(N \log N)$, where N is the total number of grid points in the domain. So far, these methods can only handle convex, usually homogeneous of degree one, Hamiltonians and become quite complicated with large initialized regions and cumbersome updating formulae if the Hamiltonian is not closely related to that of the eikonal equation $|\nabla\phi| = 1$.

The second class of methods are those that rely on time dependent Hamilton-Jacobi Equations. The advantage of these methods is that higher order schemes are easily derived. Osher [8] provided a rigorous link between static and time dependent Hamilton-Jacobi equations. The zero level set of the viscosity solution

ψ of

$$\psi_t(x, t) + H(x, \nabla\psi(x, t)) = 0 \tag{2}$$

with suitable initial conditions at a later time t is the set of x such that $\phi(x) = t$ of (1). This gives an approach that one can use to solve the time-dependent equation by the localized level set formulation [9] [11] with high order approximations on the partial derivatives [10] [6]. Another approach to obtaining a “time” dependent Hamilton-Jacobi equation from a static Hamilton-Jacobi equation is using so-called paraxial formulations by assuming that there is a preferred direction in the wave propagation. In [4], a paraxial formulation was first proposed for the eikonal equation $|\nabla\phi| = 1$. Later in [14] [15], a paraxial formulation was proposed for convex Hamilton-Jacobi equations which is efficient in some geophysical applications.

The third, and final, class of algorithms rely on iterative methods. Rouy and Tourin [16] used an upwind, monotone, and consistent discretization for $|\nabla\phi|$ to solve the discretized eikonal equation iteratively and proved that it converges to the viscosity solution. For convex Hamiltonians, Tsai et al. [20] used a fast Gauss-Seidel type iteration method, first surmised in [2], and a monotone upwind Godunov Hamiltonian, first obtained in [1] and evaluated in [10], as the numerical Hamiltonian. Kao et al. [7] proposed a new interpretation of the monotone upwind Godunov Hamiltonian for the numerical Hamiltonian based on the Legendre transform and also used a Gauss-Seidel fast sweeping method. The complexity of these two methods is $O(N)$. This has been rigorously proved in [22] only for special cases, but the numerical evidence is convincing that this is true in very general convex cases.

However, all of the above cited methods are designed for static Hamilton-Jacobi equations by assuming convexity and/or homogeneity of Hamiltonians. In this paper, we propose a new Gauss-Seidel sweeping type algorithm which is based on the Lax-Friedrichs Hamiltonian. It can handle both convex and nonconvex very complicated Hamiltonians. The evaluation of H uses data from the previous step. This makes the speed of the algorithm dramatically fast. The algorithm can deal with boundary conditions specified on complicated subset Γ . Also, the algorithm is extremely easy to implement, using less than 100 lines of code.

2 The Lax-Friedrichs Sweeping Scheme

If a monotone scheme based on the Godunov Hamiltonian is applied to equation (1), then a nontrivial calculation involving minima and maxima needs to be carried out at each grid point to solve for a grid value in terms of its neighbors. This can be done without too much the difficulty for convex Hamiltonians. For example, the ordered upwind method [18] updates the grid value using a minimization formula which essentially boils down to a version of Godunov type monotone scheme for convex and homogeneous Hamiltonians. To update the solution at each grid point, the ordered upwind method searches the whole “considered” front in order to find an approximately correct direction to satisfy

the point-wise causality; thus this can involve an extensive, computationally costly search, and the resulting method has $O(N \log N)$ complexity, where N is the total number of grid points. An optimal method of $O(N)$ complexity may be designed by following a group of characteristics at each iteration, so that a group-wise causality principle is satisfied. This led to the design of the fast sweeping methods proposed in [22] [20] [7], and the method for convex Hamilton-Jacobi equation apparently has only $O(N)$ complexity and is simple to implement.

However, if the Hamiltonian in (1) is nonconvex, then the Godunov Hamiltonian gives rise to a formula involving minima and maxima which is extremely hard to carry out; therefore, in this case it is a nontrivial task to solve for a grid value in terms of its neighbors. Hence we resort to using the Lax-Friedrichs Hamiltonian to avoid a complicated optimization process always needed for the Godunov Hamiltonian.

2.1 Lax-Friedrichs Hamiltonian

Our new numerical algorithm for static Hamilton-Jacobi equations

$$\begin{cases} H(x, \nabla \phi(x)) = R(x) & x \in \Omega \\ \phi(x) = q(x) & x \in \Gamma \end{cases}$$

is composed of an update formula and a sweeping process which can handle both convex and nonconvex cases. The one dimensional Lax-Friedrichs Hamiltonian is (dropping the obvious x dependence on H)

$$\tilde{H}^{LF}(p^-, p^+) = H\left(\frac{p^+ + p^-}{2}\right) - \sigma_x \frac{p^+ - p^-}{2}$$

where σ_x is the artificial viscosity satisfying

$$\sigma_x \geq \max \left| \frac{\partial H}{\partial p} \right|,$$

$p = \frac{\partial \phi}{\partial x}$ and p^\pm are corresponding forward and backward differences approximation of $\frac{\partial \phi}{\partial x}$. We want to use alternating Gauss-Seidel iterations to obtain the numerical approximation of solutions in the following discretization

$$\tilde{H}^{LF} = R.$$

In order to have simple update formula, we choose

$$\phi_i^{n+1} = \frac{\Delta x}{\sigma_x} \left(R(x_i) - H\left(x_i, \frac{(p^+)^n + (p^-)^n}{2}\right) \right) + \frac{\phi_{i+1} + \phi_{i-1}}{2} \quad (3)$$

We do not specify the step by putting superscripts on ϕ_{i+1} and ϕ_{i-1} because they depend on the sweeping directions. If we sweep from left to right, $\phi_{i-1} = \phi_{i-1}^{n+1}$ and $\phi_{i+1} = \phi_{i+1}^n$ because we use the newest values for Gauss-Seidel iteration.

Of course the opposite is true if we go from right to left. The formulas for two and three dimensions can be obtained by similar procedures:

$$\begin{aligned} \phi_{i,j}^{n+1} &= \left(\frac{1}{\frac{\sigma_x}{\Delta x} + \frac{\sigma_y}{\Delta y}} \right) \left(R - H \left(x, \frac{(p^+)^n + (p^-)^n}{2}, \frac{(q^+)^n + (q^-)^n}{2} \right) \right) \\ &+ \left(\frac{1}{\frac{\sigma_x}{\Delta x} + \frac{\sigma_y}{\Delta y}} \right) \left(\sigma_x \frac{\phi_{i+1,j} + \phi_{i-1,j}}{2 \Delta x} + \sigma_y \frac{\phi_{i,j+1} + \phi_{i,j-1}}{2 \Delta y} \right), \end{aligned} \quad (4)$$

$$\begin{aligned} \phi_{i,j,k}^{n+1} &= c \left\{ R - H \left(x, \frac{(p^+)^n + (p^-)^n}{2}, \frac{(q^+)^n + (q^-)^n}{2}, \frac{(r^+)^n + (r^-)^n}{2} \right) \right. \\ &+ \left. \sigma_x \frac{\phi_{i+1,j,k} + \phi_{i-1,j,k}}{2 \Delta x} + \sigma_y \frac{\phi_{i,j+1,k} + \phi_{i,j-1,k}}{2 \Delta y} + \sigma_z \frac{\phi_{i,j,k+1} + \phi_{i,j,k-1}}{2 \Delta z} \right\}, \end{aligned} \quad (5)$$

where $q = \frac{\partial \phi}{\partial y}$ and $r = \frac{\partial \phi}{\partial z}$,

$$c = \frac{1}{\frac{\sigma_x}{\Delta x} + \frac{\sigma_y}{\Delta y} + \frac{\sigma_z}{\Delta z}},$$

and σ_x , σ_y and σ_z are artificial viscosities satisfying

$$\sigma_x \geq \max \left| \frac{\partial H}{\partial p} \right|, \sigma_y \geq \left| \frac{\partial H}{\partial q} \right|, \text{ and } \sigma_z \geq \max \left| \frac{\partial H}{\partial r} \right|.$$

We remark that no nonlinear inversion is required in the above formulae, therefore the algorithm is simple to implement, no matter how complicated the Hamiltonian might be.

2.2 Computational Boundary Condition

There is a major difference between the Godunov Hamiltonian and the Lax-Friedrichs Hamiltonian. The Godunov Hamiltonian is an upwind Hamiltonian while the Lax-Friedrichs Hamiltonian is not. The Godunov Hamiltonian will choose the grid points automatically to give reasonable results on the computational boundary given that characteristics are assumed to flow out of the regions. However, the Lax-Friedrichs Hamiltonian gives a solution depending on all of its neighbors in all Cartesian dimensions. If we do not carefully specify the values of points outside of the computational domain, a huge error will be introduced for the points on the computational boundary and then propagate into the computational domain.

For simplicity, we describe the two dimensional case on the compact domain $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$. Suppose we have the uniform discretization (x_i, y_j) , $i = 0, 1, \dots, m_1, m_1 + 1$, and $j = 0, 1, \dots, m_2, m_2 + 1$ where $x_i = (i - 1) \Delta x + x_{min}$,

$y_j = (j - 1) \Delta y + y_{min}$, $\Delta x = \frac{x_{max} - x_{min}}{m_1 - 1}$ and $\Delta y = \frac{y_{max} - y_{min}}{m_2 - 1}$. We use the following formulas

$$\begin{cases} \phi_{0,j}^{new} = \min(\max(2\phi_{1,j} - \phi_{2,j}, \phi_{2,j}), \phi_{0,j}^{old}) \\ \phi_{m_1+1,j}^{new} = \min(\max(2\phi_{m_1,j} - \phi_{m_1-1,j}, \phi_{m_1-1,j}), \phi_{m_1+1,j}^{old}) \\ \phi_{i,0}^{new} = \min(\max(2\phi_{i,1} - \phi_{i,2}, \phi_{i,2}), \phi_{i,0}^{old}) \\ \phi_{i,m_2+1}^{new} = \min(\max(2\phi_{i,m_2} - \phi_{i,m_2-1}, \phi_{i,m_2-1}), \phi_{i,m_2+1}^{old}) \end{cases} \quad (6)$$

which combine extrapolation, maximization and minimization to calculate the values for points outside of computational domain. For a source point not on a computational boundary, it is very reasonable to do linear extrapolation. It means that $p^+ = p^-$ for the points on the computational boundary. For a source point on a computational boundary, we need to choose $\frac{\partial \phi}{\partial n} = 0$ in order to avoid inflow. For well-posed problems, in the absence of physically prescribed boundary conditions, we must always have outflow on the computational boundary. That is why we choose $(\frac{\partial \phi}{\partial x})^+ \simeq \frac{\phi_{2,j} - \phi_{1,j}}{\Delta x} = (\frac{\partial \phi}{\partial x})^- \simeq \frac{\phi_{1,j} - \phi_{0,j}}{\Delta x}$ when $(\frac{\partial \phi}{\partial x})^+ \simeq \frac{\phi_{2,j} - \phi_{1,j}}{\Delta x} > 0$ and $(\frac{\partial \phi}{\partial n})^c \simeq \frac{\phi_{2,j} - \phi_{0,j}}{\Delta x} = 0$ when $(\frac{\partial \phi}{\partial x})^+ \simeq \frac{\phi_{2,j} - \phi_{1,j}}{\Delta x} \leq 0$. We want to make sure that our numerical approximation is decreasing after each iteration so we update the value only when it is less than its old value. This is how our formula (6) is derived.

2.3 Algorithm for Lax-Friedrichs Sweeping

The Lax-Friedrichs sweeping algorithm is very easy to implement. There are three steps: initialization, alternating sweeps, and enforcing the computational boundary condition. We take the two dimensional case for the sake of exposition. Suppose we have the discretization (x_i, y_j) , $i = 0, 1, \dots, m_1, m_1 + 1$ and $j = 0, 1, \dots, m_2, m_2 + 1$ as mentioned before.

1. Initialization: We assign exact values or interpolated values for $\phi_{i,j}^0$ at grid points on or near Γ . These values are fixed in later iterations. At all other grid points, we assign large positive values for $\phi_{i,j}^0$. These values will be updated in later iterations.
2. Alternating sweeps: At iteration $n + 1$, we calculate $\phi_{i,j}^{n+1}$ according to (4) at all grid points (x_i, y_j) $1 \leq i \leq m_1$, $1 \leq j \leq m_2$ except for those which have assigned values and update $\phi_{i,j}^{n+1}$ only when it is less than its previous value $\phi_{i,j}^n$. Recall that this process needs to be done in alternating sweeping directions, which means that it needs 4 different sweeps in the two dimensional case. (1) From lower left to upper right $i = 1 : m_1$, $j = 1 : m_2$, (2) from lower right to upper left $i = m_1 : 1$, $j = 1 : m_2$, (3) from upper left to lower right $i = 1 : m_1$, $j = m_2 : 1$, and (4) from upper right to lower left $i = m_1 : 1$, $j = m_2 : 1$. In general, for dimension l , we need 2^l alternating sweeps.
3. Enforcing the computational boundary condition: After each sweep, we enforce the computational boundary condition by using formula (6), trivially modified depending on which boundary we are at.

3 Properties for One Dimensional Eikonal Equation

For the one dimensional eikonal equation, $H(p) = |p|$, $R = c(x)$, we may choose the optimal $\sigma_x = 1$ and $\Delta x = h$. Thus

$$\phi_i = h \left(c(x_i) - \frac{|\phi_{i+1} - \phi_{i-1}|}{2h} \right) + \frac{\phi_{i+1} + \phi_{i-1}}{2}$$

We have

$$\phi_i = \begin{cases} hc(x_i) + \phi_{i+1} & \text{if } \phi_{i+1} \leq \phi_{i-1} \\ hc(x_i) + \phi_{i-1} & \text{if } \phi_{i-1} < \phi_{i+1} \end{cases} .$$

It is very easy to see that the computational boundary condition does not affect the result. This gives exactly the same approximation of the solution as the upwind Godunov method does, no matter how many source points we have.

For general $\sigma_x > 1$, we have the update formula

$$\phi_i = \begin{cases} \frac{h}{\sigma_x} + \left(\frac{1}{2} + \frac{1}{2\sigma_x}\right)\phi_{i+1} + \left(\frac{1}{2} - \frac{1}{2\sigma_x}\right)\phi_{i-1} & \text{if } \phi_{i+1} \leq \phi_{i-1} \\ \frac{h}{\sigma_y} + \left(\frac{1}{2} + \frac{1}{2\sigma_x}\right)\phi_{i-1} + \left(\frac{1}{2} - \frac{1}{2\sigma_x}\right)\phi_{i+1} & \text{if } \phi_{i-1} < \phi_{i+1} \end{cases}$$

In this case, it is harder to prove that it converges. We first discuss the case with a single source point in the center of the domain $[-1, 1]$. Suppose we have the discretization $x_i = \frac{i}{m}$, $-m - 1 \leq i \leq m + 1$, x_{m+1} and x_{-m-1} are points outside of domain, and $\phi(0) = 0$, $x_0 = 0$. We sweep from left to right and then from right to left. The approximation after two sweeps on the left hand side of the center is symmetric to the approximation after one sweep on the right side. Without loss of generality, we can just discuss the approximations on the right hand side of the center. Denote $a = \left(\frac{1}{2} - \frac{1}{2\sigma_x}\right)$ and $b = \left(\frac{1}{2} + \frac{1}{2\sigma_x}\right)$. $(\phi^n)^+$ represents the sweep from left to right at the n th iteration while $(A^n)^-$ represents the sweep from right to left at the n th iteration. We can write down the update formula as the following linear system

$$A_+(\phi^{n+1})^+ = B_+(\phi^n)^- + C$$

and

$$A_-(\phi^n)^- = B_-(\phi^n)^+ + C$$

where

$$A_+ = \begin{bmatrix} 1 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ -b & 1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & -b & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & 0 & \cdot \\ \cdot & \cdot & \cdot & 0 & -b & 1 & 0 \\ 0 & \cdot & \cdot & 0 & 1 & -2 & 1 \end{bmatrix}$$

$$\begin{aligned}
B_+ &= \begin{bmatrix} 0 & a & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & 0 & a & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 & a & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 & a \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix} \\
A_- &= \begin{bmatrix} 1 & -a & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 1 & -a & 0 & \cdot & \cdot & \cdot \\ \cdot & 0 & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & -a & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 \\ 0 & \cdot & \cdot & \cdot & 1 & -2 & 1 \end{bmatrix} \\
B_- &= \begin{bmatrix} 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ b & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & b & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & b & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & b & 0 & a \\ 0 & \cdot & \cdot & \cdot & 0 & 0 & 0 \end{bmatrix} \\
\phi^n &= \begin{bmatrix} \phi_1^n \\ \phi_2^n \\ \cdot \\ \cdot \\ \cdot \\ \phi_m^n \\ \phi_{m+1}^n \end{bmatrix} \quad \text{and, } C = \begin{bmatrix} \frac{h}{\sigma_x} \\ \frac{h}{\sigma_x} \\ \cdot \\ \cdot \\ \frac{h}{\sigma_x} \\ \frac{h}{\sigma_x} \\ 0 \end{bmatrix}
\end{aligned}$$

Thus the update formula becomes

$$(\phi^{n+1})^+ = (A_+)^{-1}B_+(\phi^n)^- + (A_+)^{-1}C = \widehat{B}_+(\phi^n)^- + \widehat{C}_+$$

and

$$(\phi^n)^- = (A_-)^{-1}B_-(\phi^n)^+ + (A_-)^{-1}C = \widehat{B}_-(\phi^n)^+ + \widehat{C}_-$$

where

$$\widehat{B}_+ = \begin{bmatrix} 0 & a & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & ab & a & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & ab^2 & ab & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & ab^3 & ab^2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a & 0 & \cdot \\ 0 & ab^{m-2} & ab^{m-3} & \cdot & \cdot & \cdot & ab & a & 0 \\ 0 & ab^{m-1} & ab^{m-2} & \cdot & \cdot & \cdot & ab^2 & ab & a \\ 0 & -ab^{m-2} + 2ab^{m-1} & -ab^{m-3} + 2ab^{m-2} & \cdot & \cdot & \cdot & -ab + 2ab^2 & -a + 2ab & 2a \end{bmatrix}$$

$$\widehat{B}_- = \begin{bmatrix} ab & a^2b & a^3b & \cdot & \cdot & \cdot & a^{m-2}b & a^{m-1}b & 0 & a^m \\ b & ab & a^2b & a^3b & \cdot & \cdot & \cdot & a^{m-2}b & 0 & a^{m-1} \\ 0 & b & ab & a^2b & a^3b & \cdot & \cdot & a^{m-3}b & 0 & a^{m-2} \\ \cdot & 0 & b & \cdot \\ \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a^2b & a^3b & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ab & a^2b & \cdot & a^3 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & ab & 0 & a^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & b & 0 & a \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & -b & (2-a)b & 0 & (2-a)a \end{bmatrix}$$

$$\widehat{C}_+ = \begin{bmatrix} \frac{h}{\sigma_x} \\ \frac{h}{\sigma_x}(1+b) \\ \frac{h}{\sigma_x}(1+b+b^2) \\ \cdot \\ \cdot \\ \cdot \\ \frac{h}{\sigma_x}(1+b+\dots+b^{m-3}+b^{m-2}) \\ \frac{h}{\sigma_x}(1+b+\dots+b^{m-2}+b^{m-1}) \\ \frac{h}{\sigma_x}(2+b+\dots+b^{m-2}+2b^{m-1}) \end{bmatrix}$$

$$\widehat{C}_- = \begin{bmatrix} \frac{h}{\sigma_x}(1+a+\dots+a^{m-2}+a^{m-1}) \\ \frac{h}{\sigma_x}(1+a+\dots+a^{m-3}+a^{m-2}) \\ \frac{h}{\sigma_x}(1+a+\dots+a^{m-4}+a^{m-3}) \\ \cdot \\ \cdot \\ \cdot \\ \frac{h}{\sigma_x}(1+a+a^2) \\ \frac{h}{\sigma_x}(1+a) \\ \frac{h}{\sigma_x} \\ (1-a)\frac{h}{\sigma_x} \end{bmatrix}$$

We have convergence for any fixed m when the spectral radius $\rho(\widehat{B}_+\widehat{B}_-) < 1$ because

$$(\phi^{n+1})^+ = \widehat{B}_+\widehat{B}_-(\phi^n)^+ + \widehat{B}_+\widehat{C}_- + \widehat{C}_+$$

It is hard to calculate the spectral radius for $\widehat{B}_+\widehat{B}_-$ theoretically. We show the numerical calculation for $m = 100$ and $m = 200$ in the following. The spectral radius approaches to 1 when σ_x goes to infinity, but vanishes rapidly as σ_x goes down to 1.

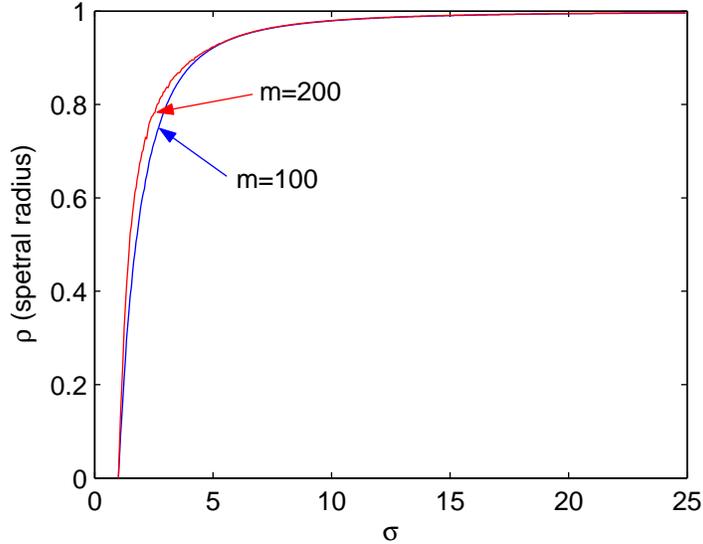


Figure 1: spectral radius of $\widehat{B}_+ \widehat{B}_-$

4 Numerical Simulation

We primarily test our Lax-Friedrichs sweeping method on Wulff crystal shape problems and the traveltime calculation of elastic waves. However, its usage is not limited to these two areas. Our method works for general static Hamilton-Jacobi equations (1). In each of the following example, we consider the iterations convergent if the L_1 norm of the difference of two successive iterations $\|\phi^{n+1} - \phi^n\|_{L_1}$ is less than 10^{-10} . Generally, we have convergence within a few hundred iterations. Even though our algorithm needs more iterations than the sweeping methods based on the Godunov Hamiltonian, it is still very fast. This is because the Lax-Friedrichs sweeping scheme does not involve any nonlinear inversion at all, let alone a complicated procedure involving many “if” statements.

4.1 The Wulff Crystal Shape

The level set formulation of the Wulff crystal shape problems [12] is

$$\begin{cases} \psi_t + \gamma \left(\frac{|\nabla \psi|}{|\nabla \psi|} \right) |\nabla \psi| = 0, & x \in R^d, t > 0 \\ \psi = 0, & x \in \Gamma \end{cases} \quad (7)$$

where γ is the normal speed, also known as the surface tension in the material science. The zero level set of the viscosity solution ψ of (7) at time t is the viscosity solution $\phi(x, y) = t$ of the following static Hamilton-Jacobi equation

[8]

$$\begin{cases} \gamma(\frac{\nabla\phi}{|\nabla\phi|})|\nabla\phi| = 1, & x \in R^d \\ \phi = 0, & x \in \Gamma \end{cases} \quad (8)$$

If Γ is a collection of closed surfaces of codimension one, there is no difference between these two formulations. If Γ is more complicated, (8) gives both inward and outward propagation.

In [12], γ is given as $\gamma(\nu)$ where ν is the angle between the outward normal direction $\frac{\nabla\phi}{|\nabla\phi|}$ and x-axis, $-\pi < \nu \leq \pi$ for two dimensional cases. Thus we have $\cos(\nu) = \frac{p}{\sqrt{p^2+q^2}}$ and $\sin(\nu) = \frac{q}{\sqrt{p^2+q^2}}$, where $p = \frac{\partial\phi}{\partial x}$ and $q = \frac{\partial\phi}{\partial y}$. For three dimensional cases, $\gamma = \gamma(\nu, \varphi) = \tilde{\gamma}(\nu)h(\varphi)$ is given, where ν and φ are the spherical coordinates, $-\pi < \nu \leq \pi$, $\frac{\pi}{2} < \varphi \leq \frac{\pi}{2}$. Thus we have

$$\begin{aligned} \cos(\nu) &= \frac{p}{\sqrt{p^2+q^2}} & \sin(\nu) &= \frac{q}{\sqrt{p^2+q^2}} \\ \cos(\varphi) &= \frac{\sqrt{p^2+q^2}}{\sqrt{p^2+q^2+r^2}} & \text{and } \cos(\varphi) &= \frac{\sqrt{p^2+q^2}}{\sqrt{p^2+q^2+r^2}} \end{aligned}$$

where $p = \frac{\partial\phi}{\partial x}$, $q = \frac{\partial\phi}{\partial y}$ and $r = \frac{\partial\phi}{\partial z}$. Applying these trigonometric equalities to a given surface tension $\gamma(\nu)$ or $\gamma = \gamma(\nu, \varphi)$, we obtain a corresponding Hamilton-Jacobi equation. For example, if $\gamma(\nu) = 1 + |\sin(\nu + \frac{\pi}{2})|$, then we have the corresponding Hamilton-Jacobi equation

$$\sqrt{p^2 + q^2} + |p| = 1.$$

In each Wulff crystal shape problem, we specify the normal speed, obtain the corresponding Hamilton-Jacobi equation, and choose the artificial viscosity as small as possible to make the scheme monotone. For Wulff crystal shape problems, we sometimes have terms such as $\sqrt{p^2 + q^2}$ and $\sqrt{p^2 + q^2 + r^2}$ in the denominator. We need to regularize them by adding a small quantity ϵ e.g. $\epsilon = 10^{-6}$ in order to avoid “dividing by zero”.

First we apply the scheme to some two dimensional problems with three different type of boundary conditions: (1) a single source point at the center of the domain, (2) Γ is a square, and (3) 100 random source points $\{(x_i, y_i), 1 \leq i \leq 100, \text{ s.t } \phi(x_i, y_i) = 0\}$.

In Figure 2, $\gamma(\nu) = 1 + |\sin(\nu + \frac{\pi}{2})|$, the Wulff crystal shape is an ellipse. In Figure 2-1, we obtain very nice ellipse contours with a single source point at the center of the domain. In Figure 2-2, we have inward and outward propagation. The outward propagation tends to smooth the kink and the contours gradually become ellipses. The inward propagation makes the contours become vertical ellipses. However, they look more like rectangle because there is not enough space to propagate. In Figure 2-3, 100 random source points interact with each other to give rise to complicated contours.

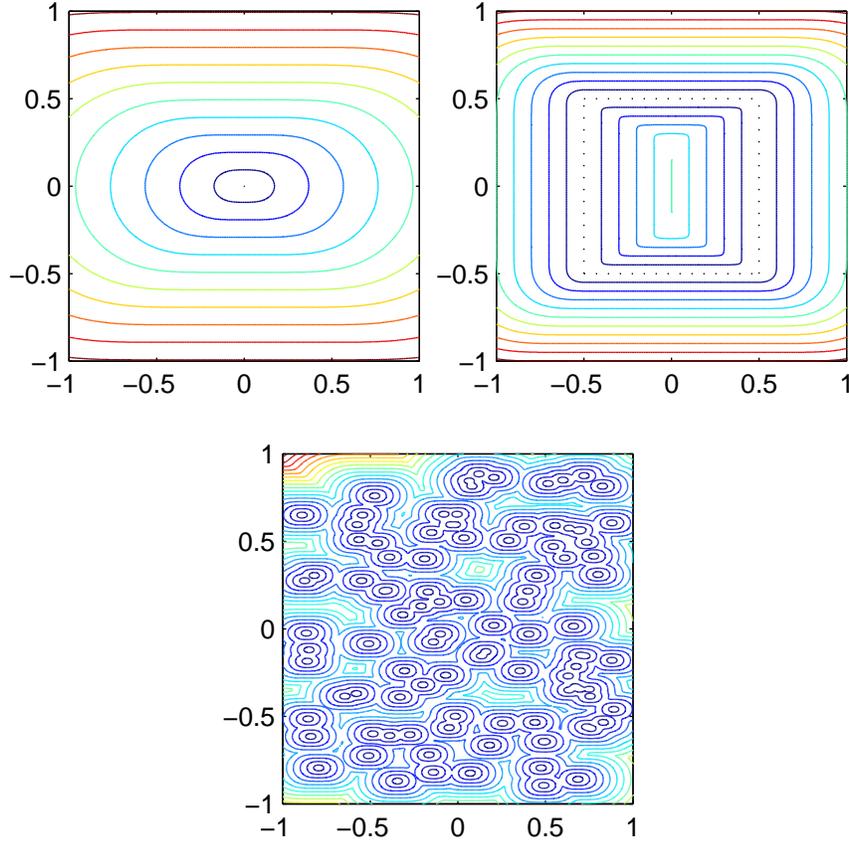


Figure 2: $|p| + \sqrt{p^2 + q^2} = 1$, $\sigma_x = 2$, $\sigma_y = 1$ (1) contour difference = 0.1, 200x200 grid, 55 iterations (2) contour difference = 0.05, 200x200 grid, 31 iterations (3) contour difference = 0.02, 400x400 grid, 36 iterations.

In Figure 3 and Figure 4, $\gamma(\nu) = 1 + |\sin(\frac{3}{2}(\nu + \frac{\pi}{2}))|$ and $\gamma(\nu) = 1 + 3|\sin(\frac{3}{2}(\nu + \frac{\pi}{2}))|$ respectively, and the Wulff crystal shapes are triangles in both cases. In Figure 5 and Figure 6, $\gamma(\nu) = |\cos(\nu)| + |\sin(\nu)|$ and $\gamma(\nu) = 1 + 3|\sin(2\nu)|$ respectively, and the Wulff crystal shapes are quadrilaterals in both cases. From these simulations, we see that the more non-convex the Hamiltonian, the sharper the facets are resolved numerically and the more iterations are needed for convergence.

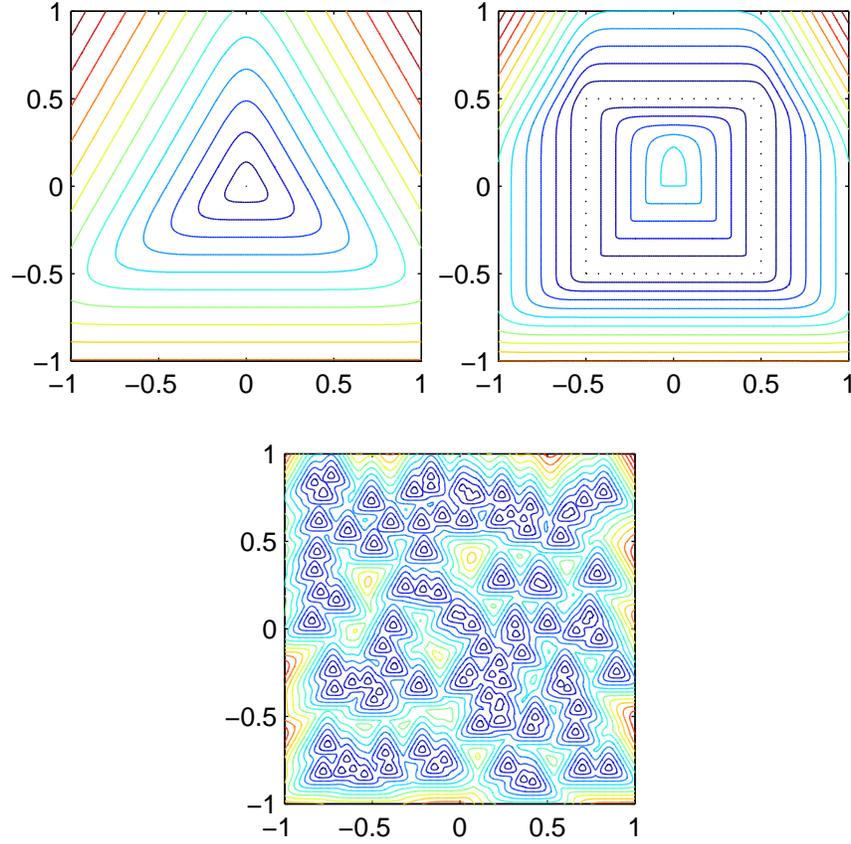


Figure 3: $\sqrt{p^2 + q^2} + \sqrt{\frac{(p^2+q^2)^{\frac{3}{2}} - (3p^2q - q^3)}{2\sqrt{p^2+q^2}}} = 1$, $\sigma_x = \sigma_y = 2$ (1) contour difference = 0.1, 200x200 grid, 100 iterations (2) contour difference = 0.05, 200x200 grid, 68 iterations (3) contour difference = 0.02, 400x400 grid, 62 iterations.

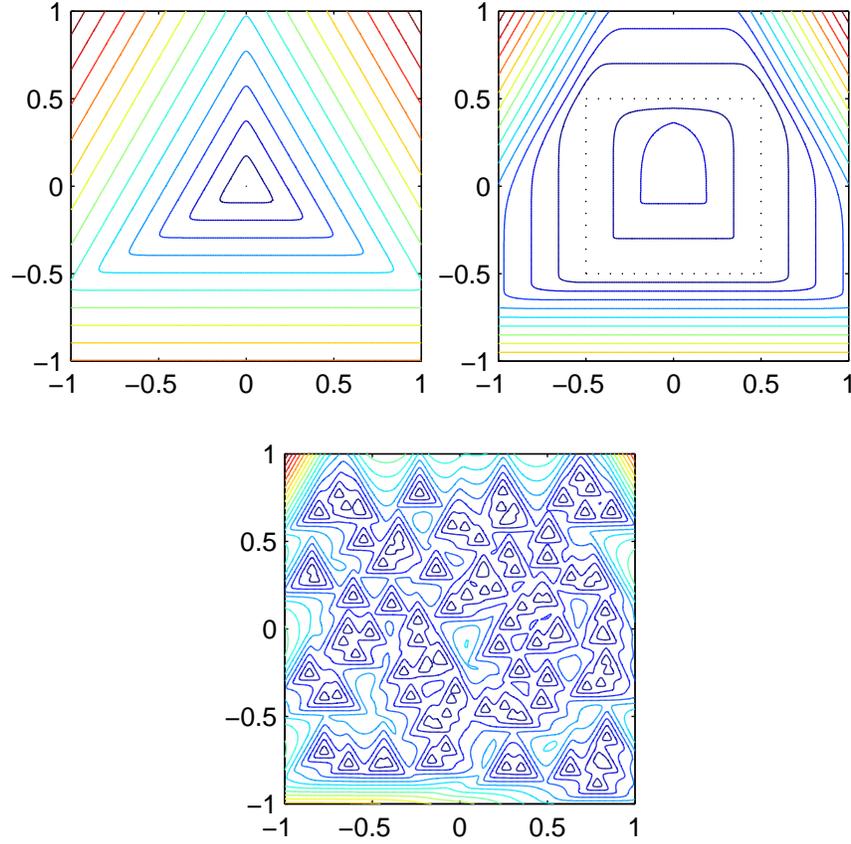


Figure 4: $\sqrt{p^2 + q^2} + 3\sqrt{\frac{(p^2+q^2)^{\frac{3}{2}} - (3p^2q - q^3)}{2\sqrt{p^2+q^2}}} = 1$, $\sigma_x = \sigma_y = 4(1)$ contour difference = 0.1, 200x200 grid, 222 iterations (2) contour difference = 0.05, 200x200 grid, 137 iterations (3) contour difference = 0.02, 400x400 grid, 130 iterations.

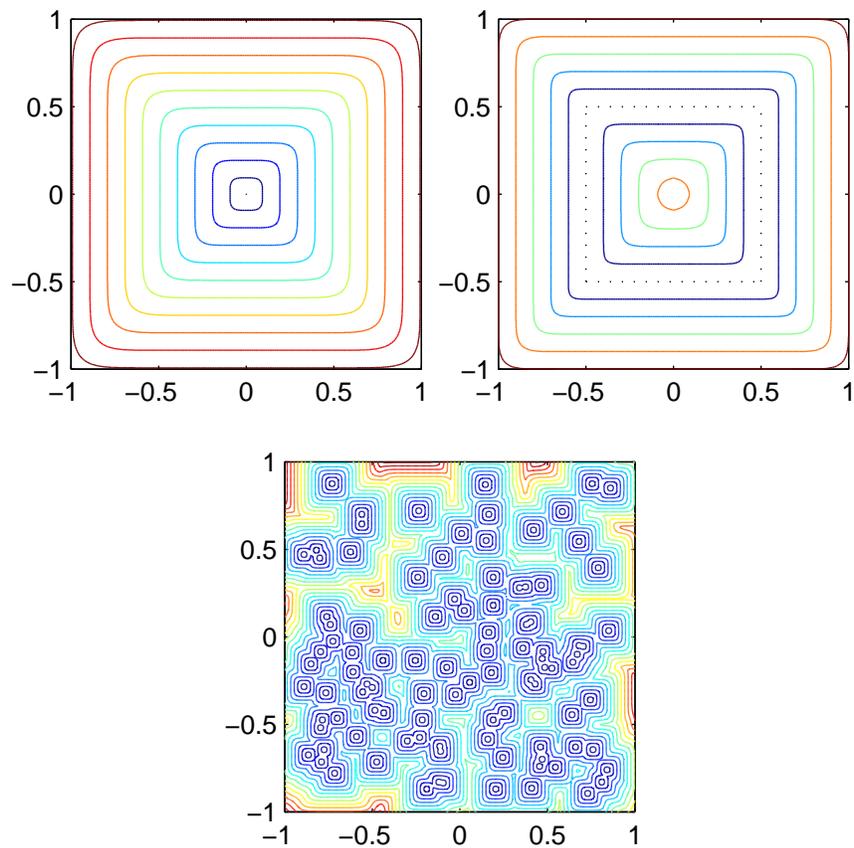


Figure 5: $|p| + |q| = 1$, $\sigma_x = \sigma_y = 1$ (1) contour difference = 0.1, 200x200 grid, 19 iterations (2) contour difference = 0.05, 200x200 grid, 3 iterations (3) contour difference = 0.02, 400x400 grid, 16 iterations.

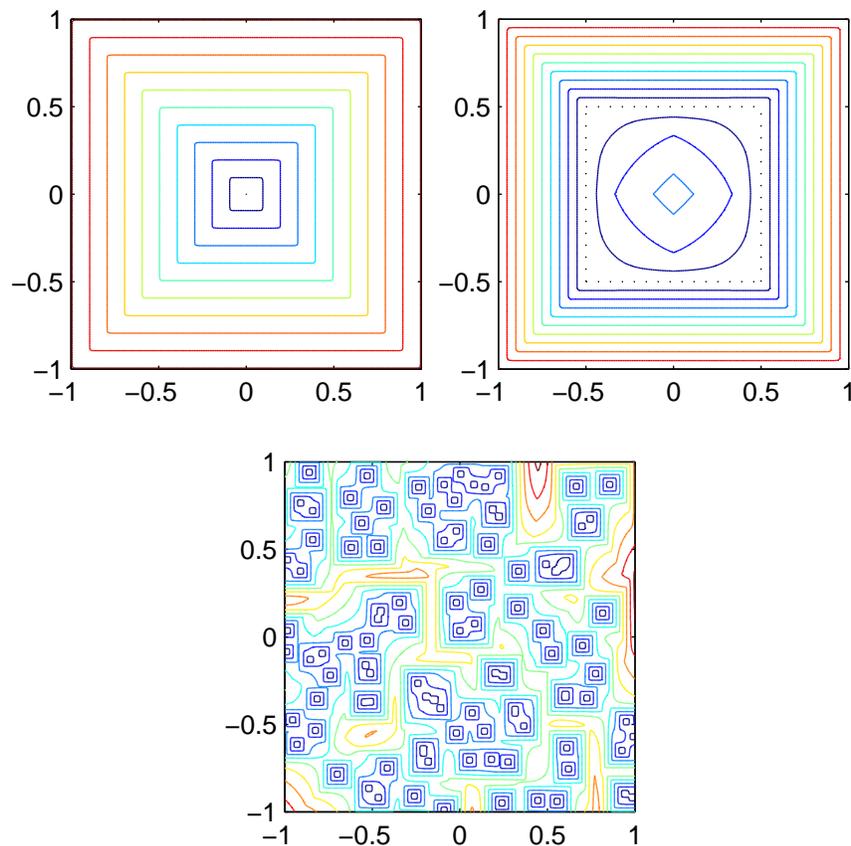


Figure 6: $\sqrt{p^2 + q^2} + \frac{6|pq|}{\sqrt{p^2 + q^2}} = 1$, $\sigma_x = \sigma_y = 4$ (1) contour difference = 0.1, 200x200 grid, 134 iterations (2) contour difference = 0.05, 200x200 grid, 120 iterations (3) contour difference = 0.02, 400x400 grid, 119 iterations.

In Figures 7, 8, 9, and 10, $\gamma(\nu) = 1 + |\sin(\frac{5}{2}(\nu + \frac{\pi}{2}))|$, $\gamma(\nu) = 1 + |\sin(3(\nu + \frac{\pi}{2}))|$, $\gamma(\nu) = 1 + |\sin(\frac{7}{2}(\nu + \frac{\pi}{2}))|$, and $\gamma(\nu) = 1 + |\sin(4\nu)|$ respectively, and the corresponding Wulff crystal shapes are pentagon, hexagon, seven multilateral, and octagon respectively.

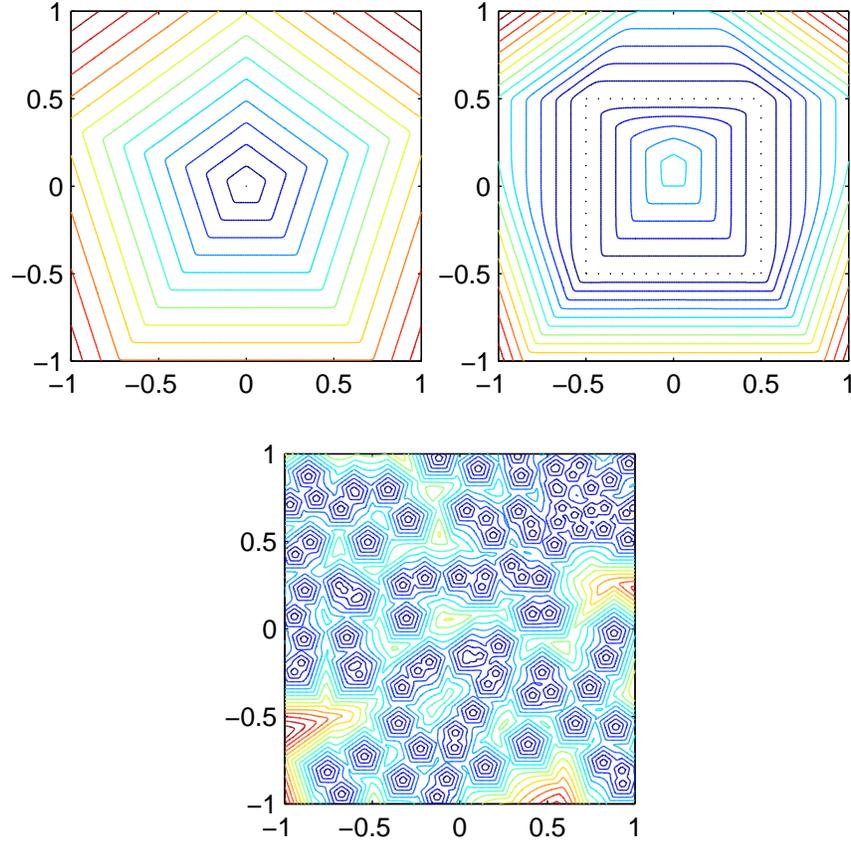


Figure 7: $\sqrt{p^2 + q^2} + \sqrt{\frac{(p^2 + q^2)^{\frac{5}{2}} - (-5qp^4 + 10q^3p^2 - q^5)}{2(p^2 + q^2)^{\frac{3}{2}}}} = 1$, $\sigma_x = \sigma_y = 2$ (1) contour difference = 0.1, 200x200 grid, 93 iterations (2) contour difference = 0.05, 200x200 grid, 58 iterations (3) contour difference = 0.02, 400x400 grid, 61 iterations.

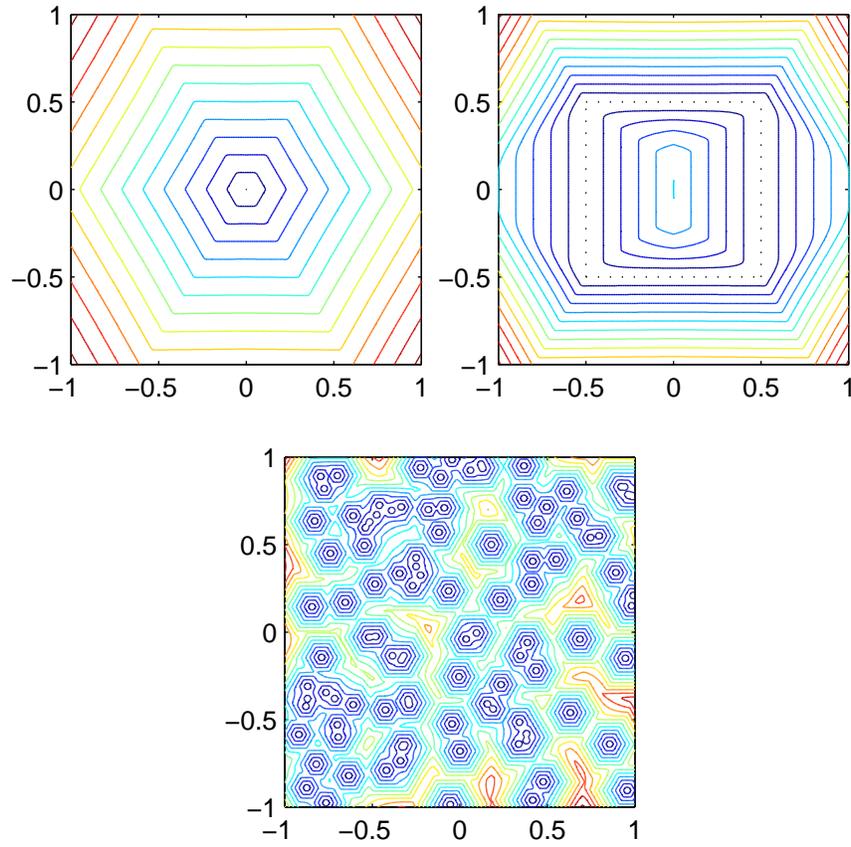


Figure 8: $\sqrt{p^2 + q^2} + \left| \frac{p^3 - 3pq^2}{p^2 + q^2} \right| = 1$, $\sigma_x = \sigma_y = 2$ (1) contour difference = 0.1, 200x200 grid, 88 iterations (2) contour difference = 0.05, 200x200 grid, 45 iterations (3) contour difference = 0.02, 400x400 grid, 58 iterations.

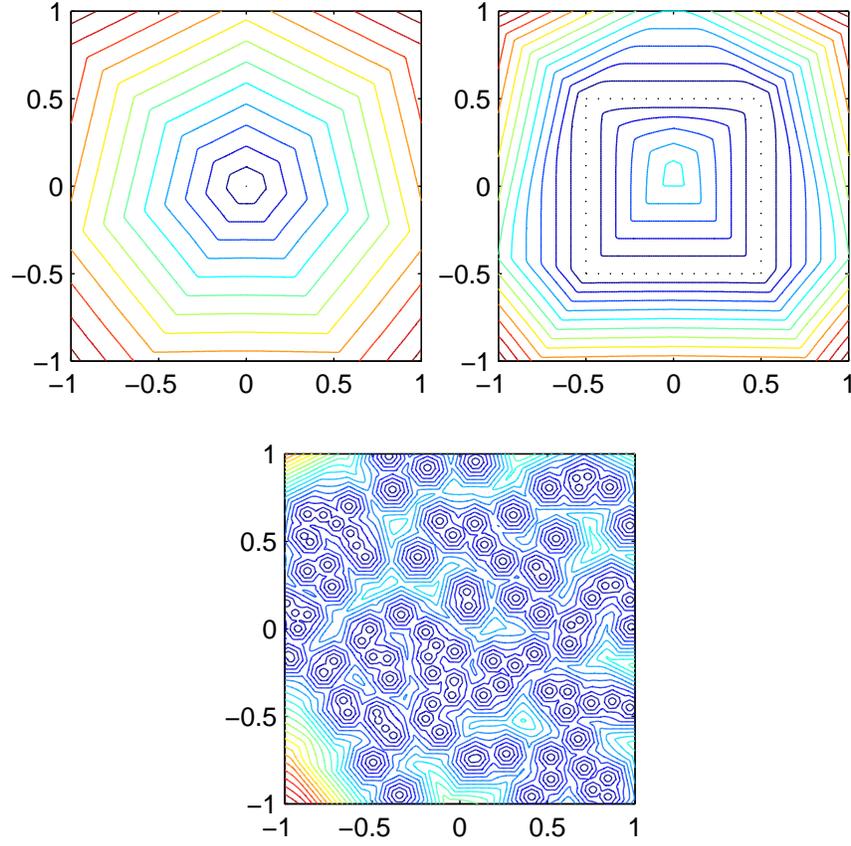


Figure 9: $\sqrt{p^2 + q^2} + \sqrt{\frac{(p^2+q^2)^{\frac{7}{2}} - (-q^7 + 21q^5p^2 - 35q^3p^4 + 7qp^6)}{2(p^2+q^2)^{\frac{5}{2}}}} = 1, \sigma_x = \sigma_y = 2$ (1)
 contour difference = 0.1, 200x200 grid, 167 iterations (2) contour difference =
 0.05, 200x200 grid, 53 iterations (3) contour difference = 0.02, 400x400 grid,
 121 iterations.

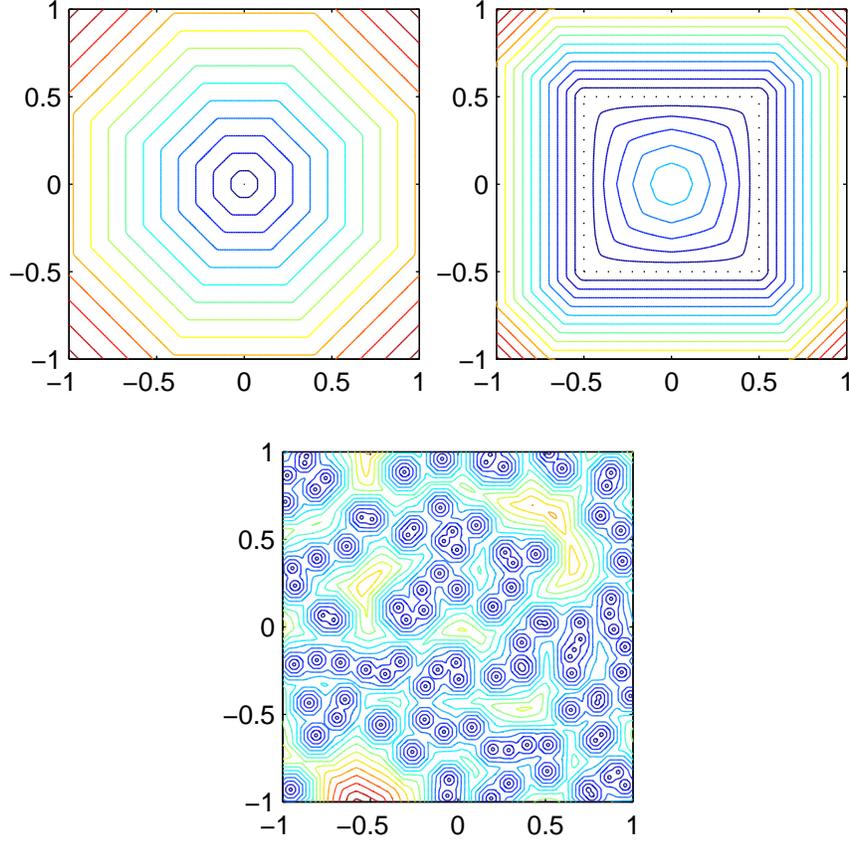


Figure 10: $\sqrt{p^2 + q^2} + \frac{|4pq(p^2 - q^2)|}{(p^2 + q^2)^{\frac{3}{2}}} = 1$, $\sigma_x = \sigma_y = 4$ (1) contour difference = 0.1, 200x200 grid, 231 iterations (2) contour difference = 0.05, 200x200 grid, 163 iterations (3) contour difference = 0.02, 400x400 grid, 195 iterations.

Next we apply the scheme to several three-dimensional examples. For ease of visualization, we only do simulations with a single source point in the center of the domain even though we can handle very complicated boundary conditions. The contours are plotted with specified differences or values.

In Figures 11, 12, 13, and 14, $h(\varphi) = (1 + 2|\sin(\varphi)|)$ and $\tilde{\gamma}$ are $(1 + |\sin(\frac{3}{2}(\nu + \frac{\pi}{2}))|)$, $(1 + |\sin(2(\nu + \frac{\pi}{2}))|)$, $(1 + |\sin(\frac{5}{2}(\nu + \frac{\pi}{2}))|)$, and $(1 + |\sin(3(\nu + \frac{\pi}{2}))|)$ respectively.

In Figure 15, $\gamma(\nu, \varphi) = (1 + 2|\sin(\frac{3}{2}(\varphi + \frac{\pi}{2}))|)(1 + |\sin(\frac{5}{2}(\nu + \frac{\pi}{2}))|)$, and the resulting Wulff crystal shapes are pyramids. In Figures 16 and 17, $\gamma(\nu, \varphi) = (1 + 2\sqrt{|\sin(|\varphi - \frac{\pi}{2}|)}) (1 + |\sin(\frac{3}{2}(\nu + \frac{\pi}{2}))|)$ and $\gamma(\nu, \varphi) = (1 + 2\sqrt{|\sin(|\varphi - \frac{\pi}{2}|)}) (1 + |\sin(\frac{5}{2}(\nu + \frac{\pi}{2}))|)$ respectively, and the Wulff crystal shapes are bi-pyramids in both cases.

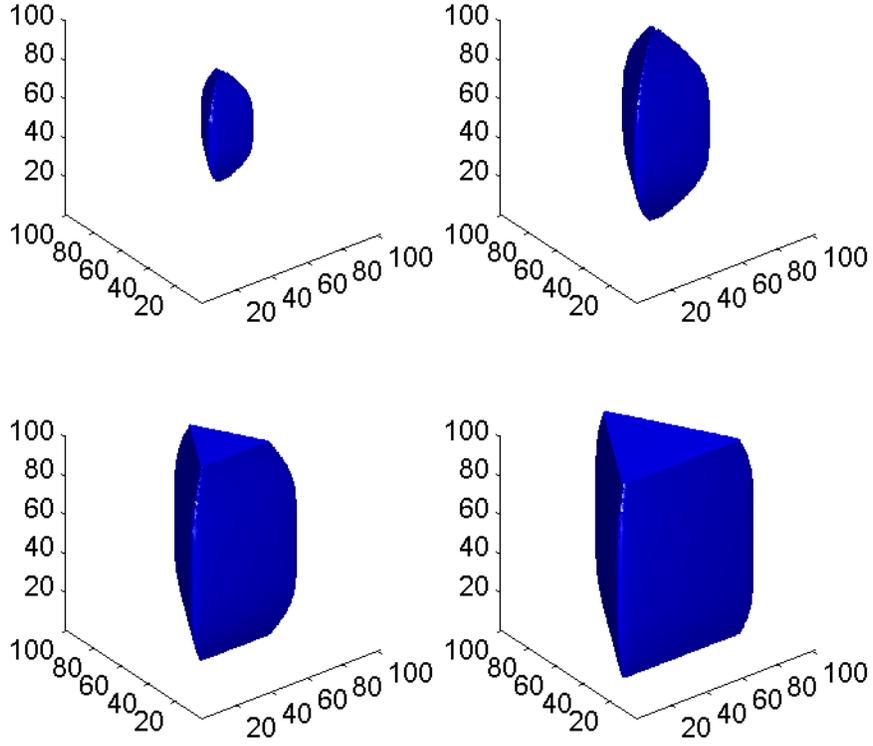


Figure 11: $(\sqrt{p^2 + q^2 + r^2} + 2|r|)(1 + 3\sqrt{\frac{(p^2+q^2)^{\frac{3}{2}} - (3p^2q - q^3)}{2(p^2+q^2)^{\frac{3}{2}}}}) = 1$, $\sigma_x = \sigma_y = \frac{9}{2}$, $\sigma_z = \frac{27}{4}$, 100x100x100 grid, 254 iterations, contour value = 0.2, 0.3, 0.4 and 0.5.

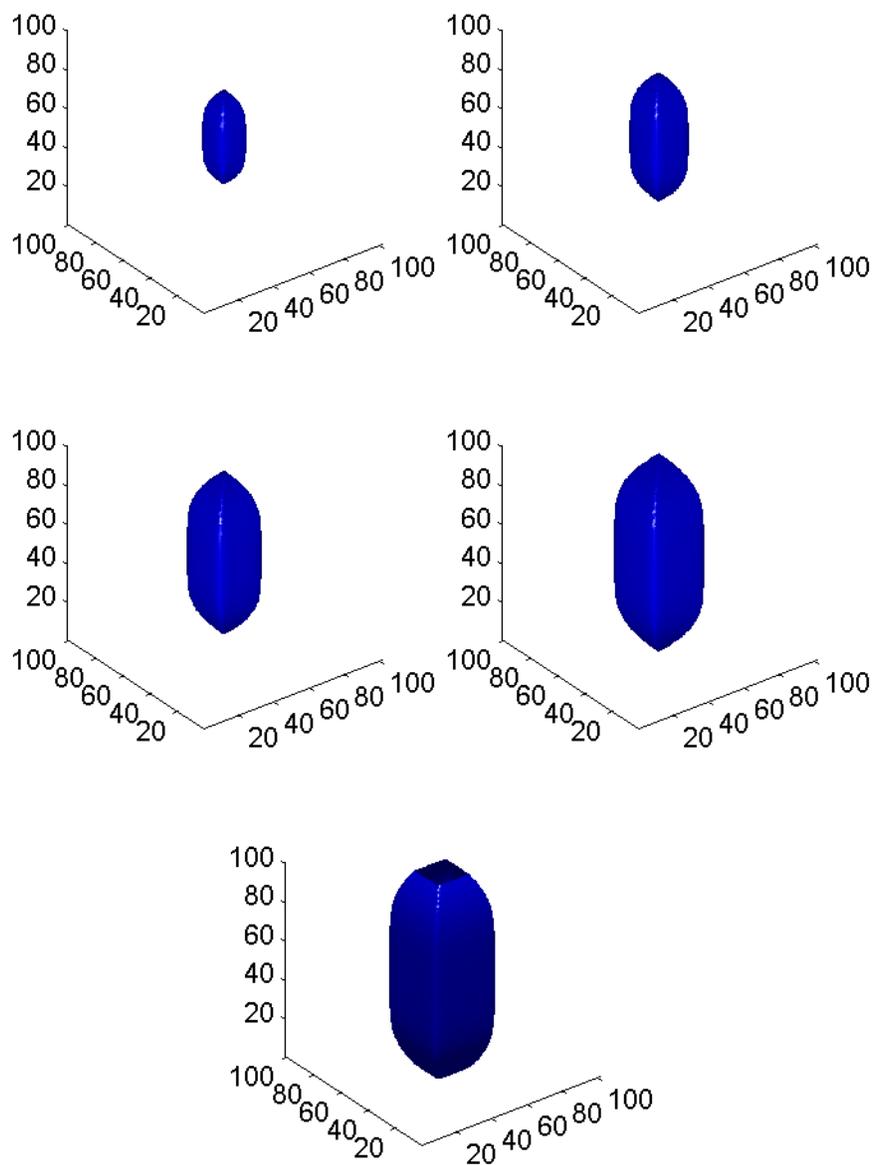


Figure 12: $(\sqrt{p^2 + q^2 + r^2} + 2|r|)(1 + \frac{|2pq|}{p^2+q^2}) = 1$, $\sigma_x = \sigma_y = \frac{5}{2}$, $\sigma_z = \frac{15}{2}$,
 100x100x100 grid, 97 iterations, contour value = 0.2, 0.25, 0.3, 0.35 and 0.4.

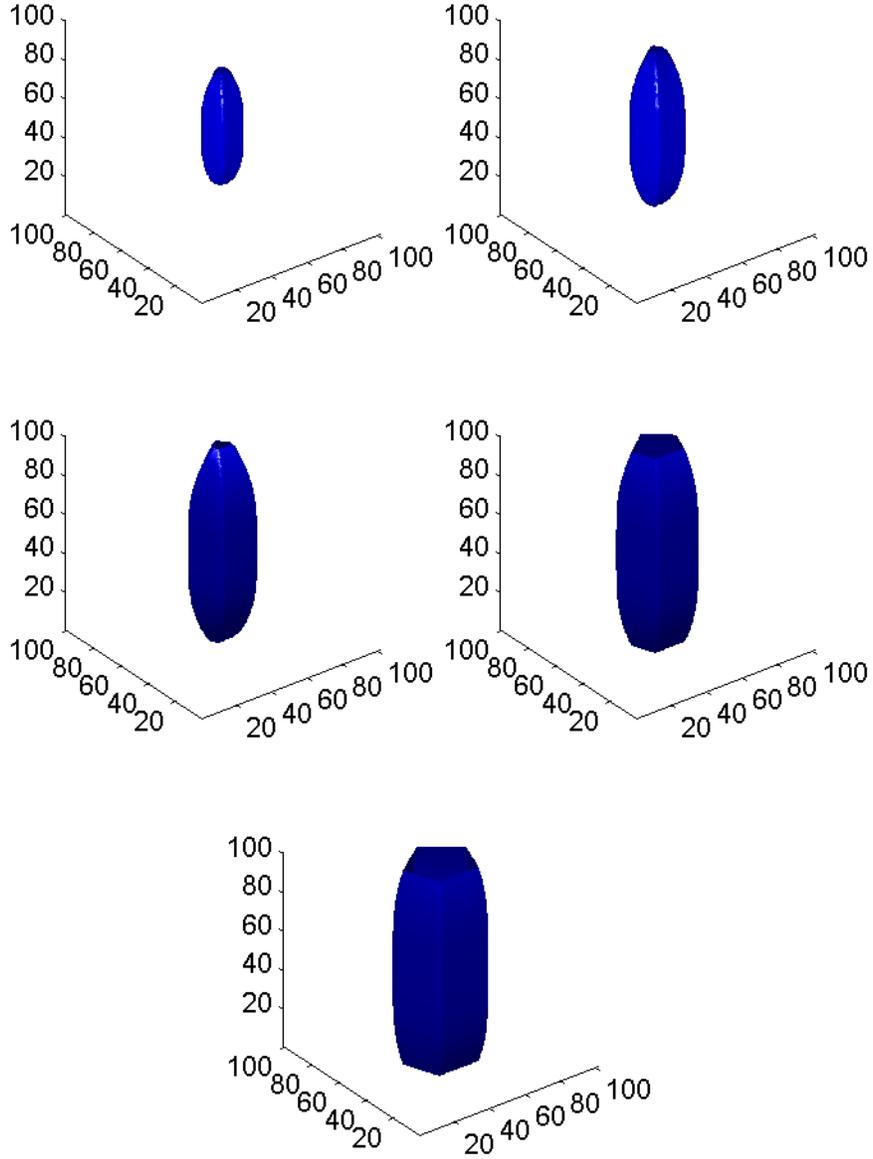


Figure 13: $(\sqrt{p^2 + q^2 + r^2} + 2|r|)(1 + \sqrt{\frac{(p^2+q^2)^{\frac{5}{2}} - (-5qp^4 + 10q^3p^2 - q^4)}{2(p^2+q^2)^{\frac{5}{2}}}}) = 1$, $\sigma_x = \sigma_y = 2$, $\sigma_z = 3$ 100x100x100 grid, 126 iterations, contour value = 0.2, 0.25, 0.3, 0.35 and 0.4.

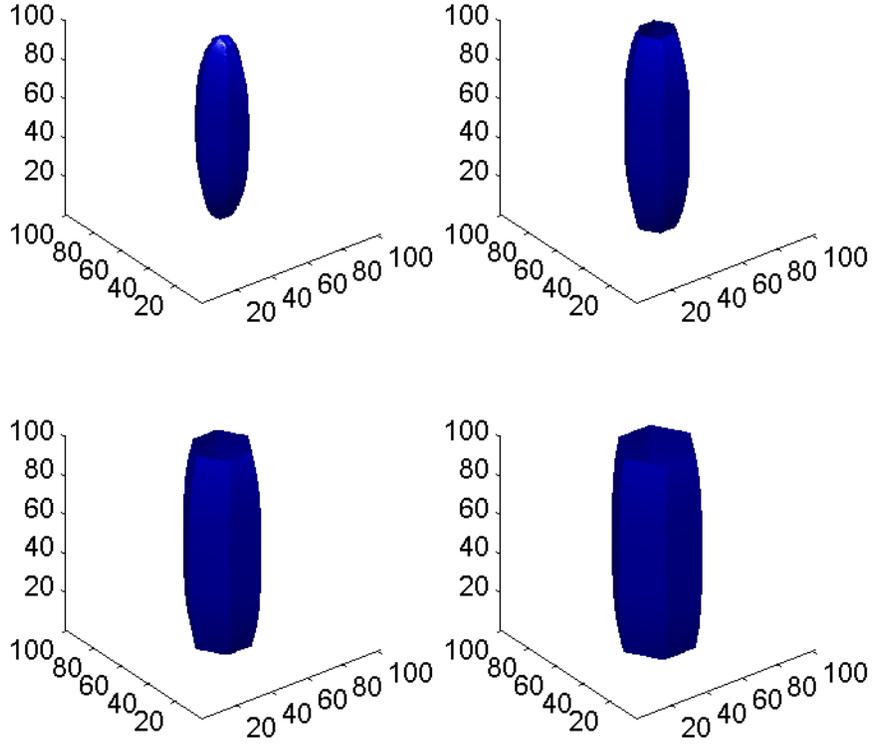


Figure 14: $(\sqrt{p^2 + q^2 + r^2} + 2|r|)(1 + \frac{|p^3 - 3pq^2|}{(p^2 + q^2)^{\frac{3}{2}}}) = 1$, $\sigma_x = \sigma_y = 2$, $\sigma_z = 6$,
 100x100x100 grid, 136 iterations, contour value = 0.25, 0.3, 0.35 and 0.4.

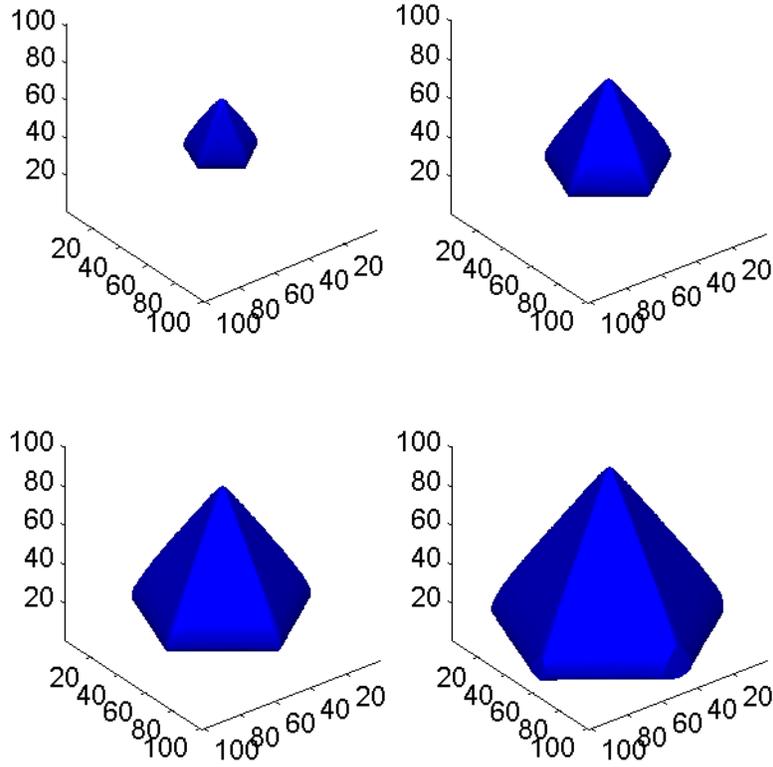


Figure 15: $(\sqrt{p^2 + q^2 + r^2})(1 + \sqrt{2(1 - \frac{3p^2r + 3q^2r - r^3}{(p^2 + q^2 + r^2)^{\frac{3}{2}}}})(1 + \sqrt{\frac{(p^2 + q^2)^{\frac{5}{2}} - (-5qp^4 + 10q^3p^2 - q^4)}{2(p^2 + q^2)^{\frac{5}{2}}}}) = 1$ $\sigma_x = \sigma_y = 3.5$, $\sigma_z = 4$, 100x100x100 grid, 179 iterations, contour value = 0.2, 0.3, 0.4 and 0.5.

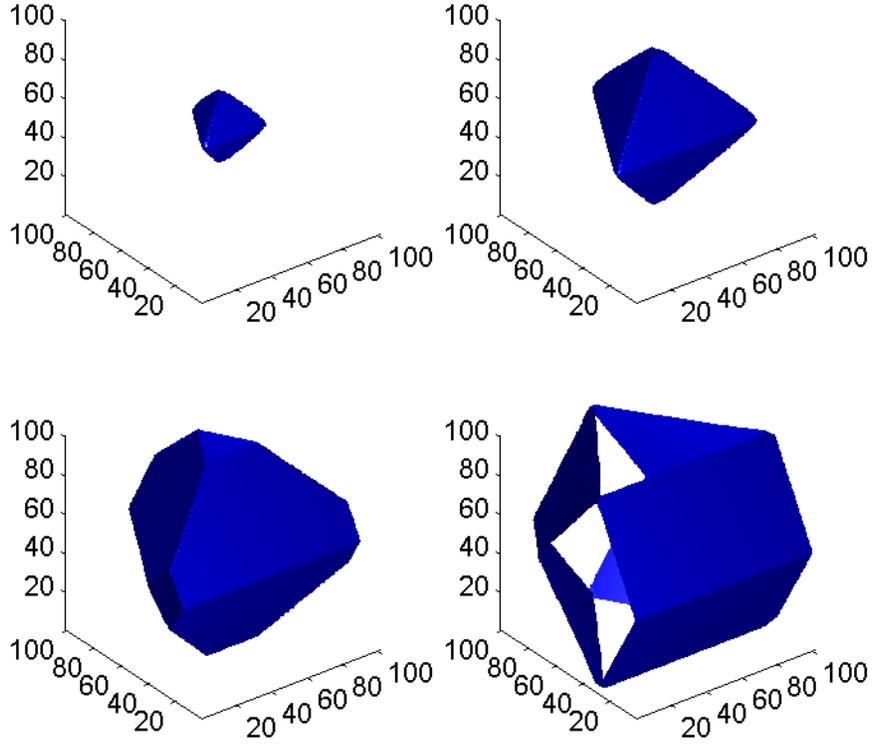


Figure 16: $(\sqrt{p^2 + q^2 + r^2} + \sqrt{2\sqrt{p^2 + q^2 + r^2}|\sqrt{3}|r| - \sqrt{p^2 + q^2}})(1 + 3\sqrt{\frac{(p^2+q^2)^{\frac{3}{2}} - (3p^2q - q^3)}{2(p^2+q^2)^{\frac{3}{2}}}}) = 1$, $\sigma_x = \sigma_y = \sigma_z = 6$, 100x100x100 grid, 91 iterations, contour value = 0.2, 0.4, 0.6 and 0.8.

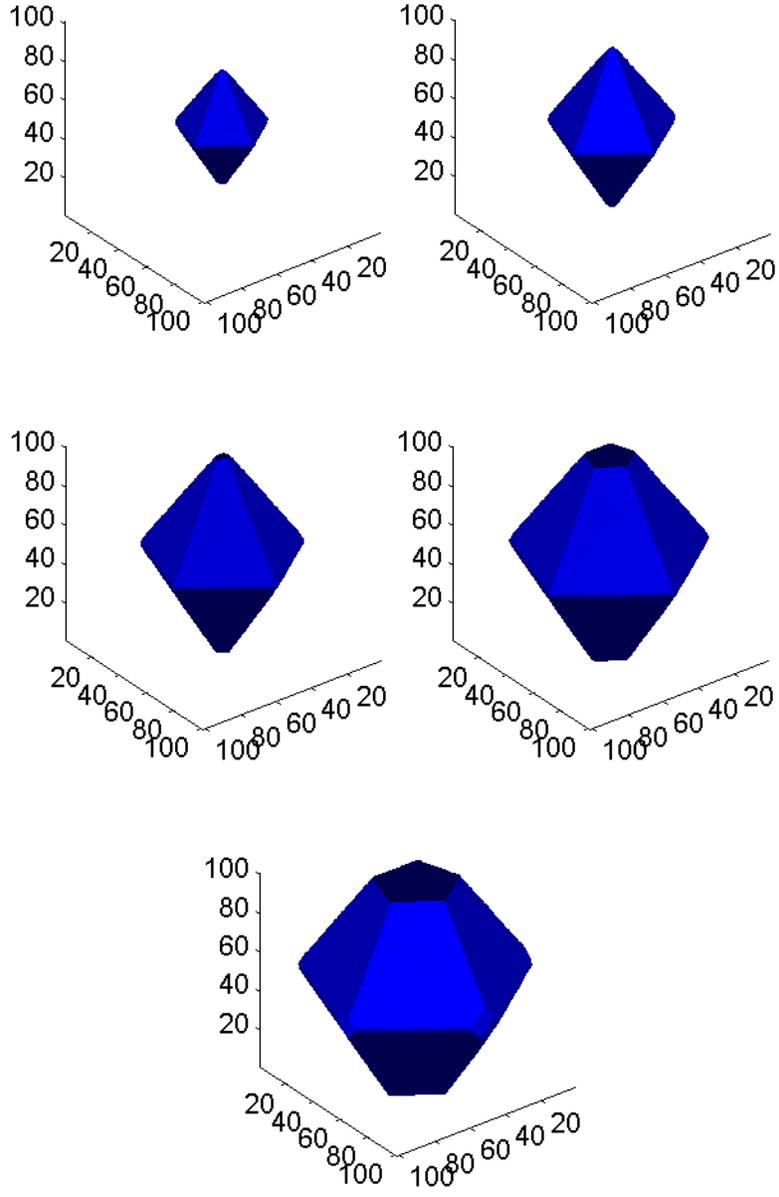


Figure 17: $(\sqrt{p^2 + q^2 + r^2} + \sqrt{2\sqrt{p^2 + q^2 + r^2}|\sqrt{3}|r| - \sqrt{p^2 + q^2}})(1 + \sqrt{\frac{(p^2+q^2)^{\frac{5}{2}} - (-5qp^4+10q^3p^2-q^4)}{2(p^2+q^2)^{\frac{5}{2}}}}) = 1$, $\sigma_x = \sigma_y = 3.5$, $\sigma_z = 4$, 100x100x100 grid, 54 iterations, contour value = 0.3, 0.4, 0.5, 0.6 and 0.7.

4.2 Traveltime Computation for The Elastic Wave Propagation

In the high frequency asymptotic for linear elastic wave propagation, we need to compute traveltime functions for three different wave modes: the quasi-P and two quasi-S waves; see [13] and reference therein for details. Here we consider a typical anisotropic elastic model, the transversely isotropic solid with horizontal symmetry. Then the quasi-P and the quasi-SV slowness surfaces are defined by the following quartic equation

$$c_1 p^4 + c_2 p^2 q^2 + c_3 q^4 + c_4 p^2 + c_5 q^2 + 1 = 0$$

where

$$\begin{aligned} c_1 &= a_{11}a_{44}, \quad c_2 = a_{11}a_{33} + a_{44}^2 - (a_{13} + a_{44})^2, \quad c_3 = a_{33}a_{44}, \\ c_4 &= -(a_{11} + a_{44}), \quad c_5 = -(a_{33} + a_{44}). \end{aligned}$$

Here a_{ij} s are given elastic parameters. Substituting $p = \frac{\partial \phi}{\partial x}$ and $q = \frac{\partial \phi}{\partial y}$ into the above equation, we have a nonlinear Hamilton-Jacobi equation for the function ϕ , the traveltime. Similarly, the quasi-SH slowness surface is defined by the equation

$$\frac{1}{2}(a_{11} - a_{12})p^2 + a_{44}q^2 = 1.$$

Since the model is transversely isotropic with horizontal symmetry, we may replace p^2 by $p^2 + r^2$ to obtain three dimensional Hamilton-Jacobi equations, where $r = \frac{\partial \phi}{\partial z}$. Furthermore, we remark that the Hamilton-Jacobi equation for quasi-SV wave traveltime in this model has a nonconvex Hamiltonian; see [13] for results related to multivalued traveltime computation.

Figure 18 shows wavefront contours for the three different wave modes in a homogeneous transversely isotropic solid. Since the Hamilton-Jacobi equation for quasi-SV traveltime is nonconvex, the quasi-SV wavefront has cusps, corresponding to a class of multivalued solutions; see [13] for capturing those multivalued wavefronts. However, the concept of the viscosity solution underlying the Lax-Friedrichs sweeping scheme allows single-valued solution only which essentially picks out the first-arrival traveltime and removes those cusps. This can be observed on Fig 18-2, where we can see that kinks appear along the two diagonals.

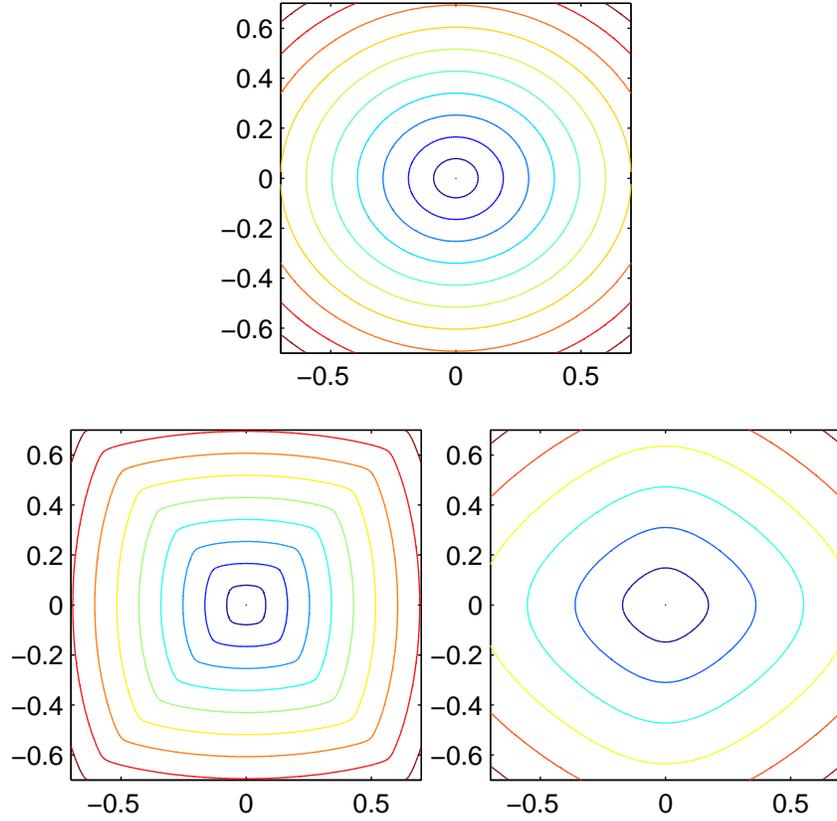


Figure 18: $a_{11} = 15.0638$, $a_{33} = 10.8373$, $a_{13} = 1.6381$, $a_{44} = 3.1258$, and $a_{12} = 6.5616$, contour difference = 0.05, 200x200 grid (1)quasi-SH: $\sigma_x = \sigma_y = 1.5$, 31 iterations (2)quasi-SV: $\sigma_x = \sigma_y = 2$, 44 iterations (3)quasi-P: $\sigma_x = \sigma_y = 3$, 50 iterations.

Figure 19 shows computational results for a model with two layers, so that the corresponding Hamilton-Jacobi equations have discontinuous coefficients; therefore, this model is used to test the stability and robustness of the sweeping scheme. As we can see from the figure, Snell's law for anisotropic media is well enforced.

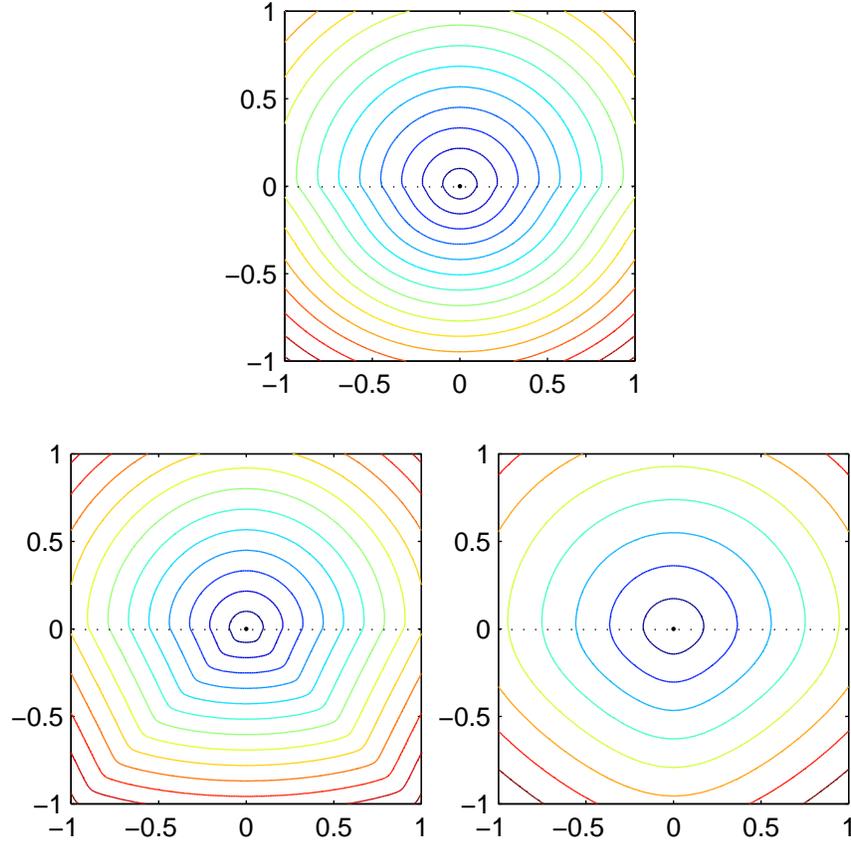


Figure 19: $a_{11} = 15.3871$, $a_{33} = 14.5161$, $a_{13} = 3.9321$, $a_{44} = 5.6074$, and $a_{12} = 3.4993$ for upper half domain, $a_{11} = 15.0638$, $a_{33} = 10.8373$, $a_{13} = 1.6381$, $a_{44} = 3.1258$, and $a_{12} = 6.5616$ for lower half domain, contour difference = 0.05, 200x200 grid (1)quasi-SH: $\sigma_x = \sigma_y = 2$, 56 iterations (2)quasi-SV: $\sigma_x = \sigma_y = 2$, 44 iterations, (3)quasi-P: $\sigma_x = \sigma_y = 3$, 48 iterations.

Figure 20 shows results for a three-dimensional transversely isotropic model with horizontal symmetry. As we can see from Figures 20-2 and 20-3, the wavefront profiles along y – direction are circles as expected from the horizontal symmetry of the model.

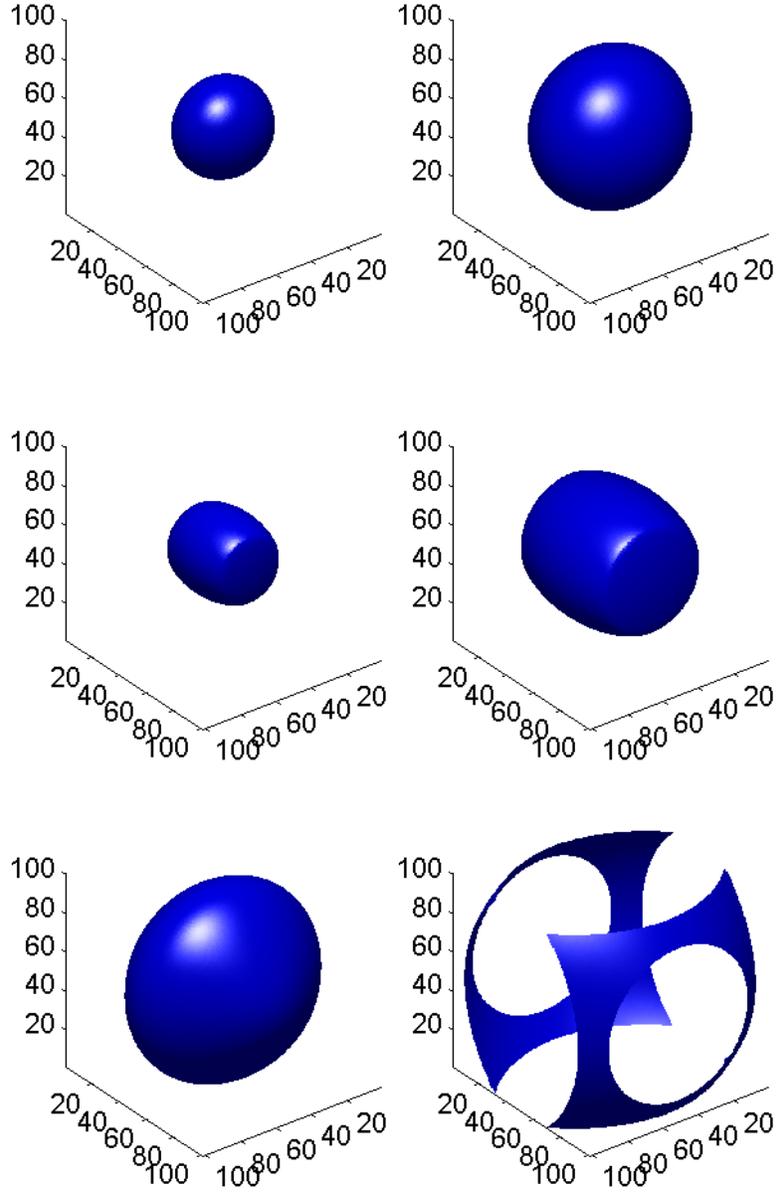


Figure 20: $a_{11} = 15.0638$, $a_{33} = 10.8373$, $a_{13} = 1.6381$, $a_{44} = 3.1258$, and $a_{12} = 6.5616$, contour value = 0.2 and 0.3, 100x100x100 grid (1)quasi-SH: $\sigma_x = \sigma_y = \sigma_z = 1.50$, 26 iterations (2)quasi-SV: $\sigma_x = \sigma_y = \sigma_z = 1.50$, 24 iterations, (3)quasi-P: $\sigma_x = \sigma_y = \sigma_z = 2.50$, 26 iterations.

4.3 Convergence Test

To validate the new Lax-Friedrichs sweeping scheme, we first apply the method to the eikonal equation $|\nabla\phi| = 1$ with a single source point at the center of the computational domain. In this case, we know the exact solution so that the convergence behavior of the method can be easily observed. Table 1 presents the L_∞ errors between computed and exact solutions for different mesh sizes. As we can see, the errors indicate the apparent first-order convergence as the mesh size approaches zero.

dx	L_∞ error	convergence order
$\frac{2}{50}$	0.062286	
$\frac{2}{100}$	0.036946	0.743490
$\frac{2}{200}$	0.019846	0.896570
$\frac{2}{400}$	0.010416	0.930047
$\frac{2}{800}$	0.005378	0.953660
$\frac{2}{1600}$	0.002750	0.967638

Table 1: The 2D eikonal case: errors and convergence order

Next we study the Lax-Friedrichs sweeping scheme for two dimensional Wulff crystal shape examples with source points located at the center of the computational domain. In these cases, we do not know the exact solutions; therefore, we consider the computed solution for the mesh size $\frac{2}{1600}$ as a good approximation of the true solution and observe the L_∞ error behaviors on coarser meshes. In Table 2, Figures 2-1, 3-1, 5-1, and 7-1 correspond to ellipse, triangle, quadrilateral, and pentagon Wulff crystal shapes, respectively. The errors listed in Table 2 also indicate first-order convergence.

example	Fig. 2-1	Fig. 3-1	Fig. 5-1	Fig. 7-1
$\ \phi_{\frac{2}{50}} - \phi_{\frac{2}{1600}}\ _\infty$	0.074627	0.142362	0.119700	0.037861
$\ \phi_{\frac{2}{100}} - \phi_{\frac{2}{1600}}\ _\infty$	0.042410	0.089958	0.073120	0.020970
$\ \phi_{\frac{2}{200}} - \phi_{\frac{2}{1600}}\ _\infty$	0.022857	0.053107	0.042770	0.013404
$\ \phi_{\frac{2}{400}} - \phi_{\frac{2}{1600}}\ _\infty$	0.011111	0.028248	0.022670	0.007039
$\ \phi_{\frac{2}{800}} - \phi_{\frac{2}{1600}}\ _\infty$	0.004121	0.011427	0.009180	0.002419

Table 2: The errors of two dimensional Wulff crystal shape examples

Table 3 presents the number of iteration for two dimensional Wulff crystal shape examples, and the result indicates that in some cases, for example, Fig 5-1, we have $O(N)$ complexity since the corresponding Hamiltonian is separable;

However, in general, we have an algorithmic complexity better than $O(N^2)$ but worse than $O(N)$.

dx	Fig. 2-1	Fig. 3-1	Fig. 5-1	Fig 7-1
$\frac{2}{50}$	28	65	16	75
$\frac{2}{100}$	39	79	18	80
$\frac{2}{200}$	55	100	19	93
$\frac{2}{400}$	86	140	20	117
$\frac{2}{800}$	146	229	20	202
$\frac{2}{1600}$	262	404	20	354

Table 3: The number of iterations for two dimensional Wulff crystal shape examples

Table 4 illustrates the computational results for some three-dimensional examples, and the errors also indicate first-order convergence.

example	Fig. 17	Fig. 20 quasi-SH	Fig. 20 quasi-P	Fig. 20 quasi-SV
$\phi_{\frac{2}{50}} - \phi_{\frac{2}{300}}$	0.139730	0.060470	0.061429	0.033068
$\phi_{\frac{2}{100}} - \phi_{\frac{2}{300}}$	0.091490	0.026058	0.027040	0.014548
$\phi_{\frac{2}{150}} - \phi_{\frac{2}{300}}$	0.067460	0.013513	0.014183	0.007627
$\phi_{\frac{2}{200}} - \phi_{\frac{2}{300}}$	0.055150	0.006913	0.007311	0.003930
$\phi_{\frac{2}{250}} - \phi_{\frac{2}{300}}$	0.045040	0.002810	0.002988	0.001606

Table 4: The errors of three dimensional examples

Table 5 shows the number of iterations for these 3-D examples; once again, the observed algorithmic complexity is much better than $O(N^2)$.

dx	Fig. 17	Fig. 20 quasi-SH	Fig. 20 quasi-P	Fig. 20 quasi-SV
$\frac{2}{50}$	42	21	21	19
$\frac{2}{100}$	52	26	26	24
$\frac{2}{150}$	90	31	31	29
$\frac{2}{200}$	109	36	36	34
$\frac{2}{250}$	131	41	41	39
$\frac{2}{300}$	206	46	47	44

Table 5: The number of iterations for three dimensional examples

5 Conclusion

In this paper, we proposed a simple, fast sweeping method based on the Lax-Friedrichs Hamiltonian to approximate the viscosity solution of static Hamilton-Jacobi equation. By using the Lax-Friedrichs Hamiltonian, we can handle much more complicated Hamiltonians than was previously done, including convex and general nonconvex cases. By using the sweeping method, we follow a group of characteristics at each iteration to speed up the algorithm. Unlike the Godunov Hamiltonian, we need to specify a simple computational boundary condition for our scheme. In order to have no inflow at the boundary, we have derived a min-max formula for the computational boundary. Some properties of the approximation for the one-dimensional eikonal equation were analyzed. We illustrated the efficiency and accuracy of the approach with extensive numerical examples in two and three dimensional cases. Currently, we are applying the method to differential game problems and we will report the result elsewhere.

References

- [1] M. Bardi and S. Osher. The nonconvex multi-dimensional Riemann problem for Hamilton-Jacobi equations. *SIAM J Math Anal*, 22(2):344–351, 1991.
- [2] M. Boué and P. Dupuis. Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control. *SIAM J Num Anal*, 36(3):667–695, 1999.
- [3] M. G. Crandall and P. L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Mathematics of Computation*, 43:1–19, 1984.
- [4] S. Gray and W. May. Kirchhoff migration using eikonal equation travel-times. *Geophysics*, 59(5):810–817, 1994.
- [5] 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.
- [6] G.-S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21(6):2126–2143 (electronic), 2000.
- [7] C. Y. Kao, S. Osher, and R. Tsai. Fast sweeping methods for Hamilton-Jacobi equations. *UCLA CAM Reports (02-66)*, *SINUM (submitted)*.
- [8] S. 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.
- [9] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

- [10] S. Osher and C.-W. Shu. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28(4):907–922, 1991.
- [11] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M. Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, 1999.
- [12] D. Peng, S. Osher, B. Merriman, and H.-K. 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.
- [13] J. Qian, L.T. Cheng, and S. Osher. A level set based Eulerian approach for anisotropic wave propagations. *Wave Motion*, 37:365–379, 2003.
- [14] J. Qian and W. W. Symes. Paraxial eikonal solvers for anisotropic quasi-P travel times. *J. Comp. Phys.*, 174(1):256–278, 2001.
- [15] J. Qian and W. W. Symes. Finite-difference quasi-P traveltimes for anisotropic media. *Geophysics*, 67:147–155, 2002.
- [16] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J Num Anal*, 29(3):867–884, 1992.
- [17] J. A. Sethian. Fast marching level set methods for three dimensional photolithography development. In *SPIE 2726*, pages 261–272, 1996.
- [18] 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.
- [19] P. E. Souganidis. Approximation schemes for viscosity solutions of Hamilton-Jacobi equations. *J. Differential Equations*, 59(1):1–43, 1985.
- [20] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, and H.-K. Zhao. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM J Num Anal*, 41(2):659–672, 2003.
- [21] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.
- [22] H.-K. Zhao. Fast sweeping method for eikonal equations I: Distance function. www.math.uci.edu/~zhao, 2002. Under review, SINUM.