

A compact upwind second order scheme for the Eikonal equation

J.-D. Benamou

INRIA, INRIA B.P. 105, 78153 Le Chesnay Cedex, France.

jean-david.benamou@inria.fr

Songting Luo

Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA.

luos@math.msu.edu

Hongkai Zhao

Department of Mathematics, University of California, Irvine, CA 92697, USA.

zhao@math.uci.edu

The research is partially supported by ONR Grant N00014-02-1-0090 ARO MURI Grant

W911NF-07-1-0185 NSF Grant DMS0811254.

Abstract

We present a compact upwind second order scheme for computing the viscosity solution of the Eikonal equation. This new scheme is based on:

1. the numerical observation that classical first order monotone upwind schemes for the Eikonal equation yield numerical upwind gradient which is also first order accurate up to singularities.
2. a remark that partial information on the second derivatives of the solution is known and given in the structure of the Eikonal equation and can be used to reduce the size of the stencil.

We implement the second order scheme as a correction to the well known sweeping method but it should be applicable to any first order monotone upwind scheme. Care is needed to choose the appropriate stencils to avoid instabilities. Numerical examples are presented.

keyword: Eikonal equation, Upwind scheme, Hamilton-Jacobi, Viscosity Solution. Sweeping method.

Contents

1	Introduction	2
2	Fast sweeping method and first order gradient	3
2.1	The 9 point stencil	4
2.2	Fast sweeping method	5
2.3	First order gradient	5
2.3.1	Uniform rectangular grid	6
2.3.2	Non-uniform rectangular grid	6
2.3.3	Triangular mesh	7
3	The compact upwind second order scheme	11
3.1	Adding the curvature of the wavefront in the discretization	11
3.2	A non linear ray direction weighted approximation	12
3.3	Simplification of the second order terms	12

4	General case and choice of stencils	14
4.1	The heterogeneous case: variable index of refraction	15
4.2	Rays on grid lines	18
4.3	Stencil near singularity	21
5	A one pass deferred correction scheme	23
6	Numerical examples	25
7	Conclusion	27

1. Introduction

The Eikonal equation:

$$\begin{cases} |\nabla\phi(x)| = n(x), & x \in \Omega/\Gamma \\ \phi(x) = \phi_0(x), & x \in \Gamma \end{cases} \quad (1.1)$$

is a special class of the Hamilton-Jacobi equations. It has wide applications in geometric optics, computer vision, optimal control and etc. This boundary value problem (1.1) is a first order hyperbolic partial differential equation (PDE). The classical method of characteristics can be applied to solve the problem. Solutions remain smooth until the characteristics cross and the fronts (level sets of the solution) intersect. Crandall and Lions [3] introduced the concept of the viscosity solutions for the Hamilton-Jacobi equations and a unique global weak solution can be defined in that sense. A weak solution remains smooth locally with the singularity in the gradient along some sub-manifold of codimension 1, 2 or 3 (in 3D).

It therefore makes sense to design high order schemes which have to remain high order away from the singularities called kinks. High order schemes are of particular importance in the high frequency wave propagation where the Eikonal equation is coupled to a transport equation through its gradient [26, 17].

Different first order numerical schemes have been developed to compute the viscosity solutions. There are two types of approaches to compute the viscosity solution of the Eikonal equation. One approach is to transform it to a time-dependent problem. For example, Osher [15] provided a natural link between the time-dependent and the static problems by using level-set ideas. Semi-Lagrangian schemes [7, 8] are obtained with the dynamical programming principle under the optimal control framework. Another approach is to treat the problem as a stationary problem and directly solve it with efficient numerical algorithms such as Dijkstra type of fast marching method (FMM) [6, 29, 9, 21, 22, 23] and iterative fast sweeping method [5, 20, 2, 31, 18, 19, 32, 28, 11, 12, 27].

However both approaches rely, in theory and in practice, on the idea of "upwind" or "causality". An efficient ordering of the application of the stencil on the grid must follow the traveltime or the level set propagation. First order upwind schemes can do it monotonically (the iterations converge monotonically to the solution) and the convergence is proven using the viscosity theory [1, 25, 4]. Even though the gradient may be singular, the method remains stable because the characteristics defining the upwind directions enter the kinks (exactly as in the case of the

shocks for the hyperbolic conservation laws).

On the other hand, second order schemes cannot be monotone [14] and in that case the viscosity theory to prove the convergence is inoperative. The popular high order ENO and WENO methods [16, 24, 10] use adaptive stencils (actually different stencils) to capture the smoothest possible approximation of the second derivatives and therefore avoid, in theory, the possible singularity of the solution. These ENO and WENO type of discretizations have been incorporated into fast sweeping method in [30]. Recently second order discontinuous Galerkin method has been developed for fast sweeping method for the Eikonal equation [13]. Also a second order fast marching method was proposed in [22]. In this approach both the solution and its gradient at accepted points, which are computed and stored during previous updates, are used to provide high order approximation of directional derivatives at a considered point during the marching process. The discretization is based on direction by direction approximation and accurate ∇u are needed near the boundary to start with the fast marching method.

Our approach is different and is based on the following two observations. The first observation is a superconvergence phenomena for first order monotone upwind methods. More precisely, the upwind numerical gradient of the solution obtained by these methods seems to remain first order accurate up to singularity, i.e., away from kinks. Apparently this phenomena has not been observed and studied in the literature. We substantiate this observation by a detailed numerical study and we are currently working on a proof. The second observation is that one only needs second derivative approximation tangential to the front, i.e., the curvature estimation, to achieve second order local truncation error by using the PDE and a decomposition of the Taylor expansion. This results in a compact and upwind second order stencil. For example, the stencil is 4 points in 2D (3 points is needed for the first order upwind scheme), which is more compact than direction by direction second order approximation. Moreover, if the second order accurate stencil can be placed upwind then we can avoid the singularity in the gradient and remain second order accurate. Our method can be regarded as an efficient one pass deferred correction to any first order monotone upwind method. It works in this way: after the first order solution is computed, the second order correction is applied to all grid points in one pass in the ascending order of the first order solution. The upwind gradient of the first order solution is used to determine the upwind stencil and compute the second order correction. In this paper we use the fast sweeping method to produce the first order solution.

Here is the outline of the paper. In Section 2 a brief summary of the fast sweeping method is presented. Numerical evidence for superconvergence of the numerical gradient for monotone upwind schemes is presented. In Section 3, the compact second order scheme is derived. Some stability issues for choosing the upwind stencil and discretization are discussed in Section 4. The full algorithm as a one pass deferred correction is prescribed in Section 5. Finally numerical results in homogeneous (constant $n(x)$) and heterogeneous cases are presented in Section 6.

2. Fast sweeping method and first order gradient

In this section, we first recall the fast sweeping method (FSM) on a rectangular mesh but with 9-point stencil (Figure 2.1) which is more accurate than the classical stencil [20, 31]. Then we show a numerical study of the superconvergence for the gradient.

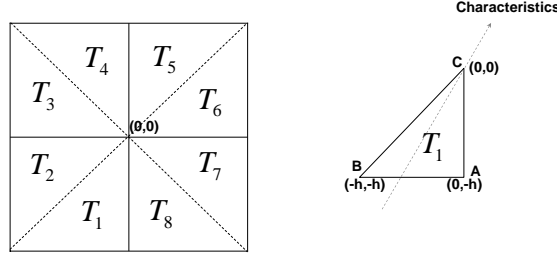


Fig. 2.1. 9-point stencil

2.1. The 9 point stencil

At each center point $(0,0)$, we do a piecewise linear approximation on each of the eight triangles in Figure 2.1. Let us focus on triangle T_1 . We denote $\nabla\phi(x) = (a_1, b_1)$, and do a first order Taylor expansion

$$\begin{cases} \phi_{0,-h} = \phi_{0,0} - b_1 h + O(h^2) \\ \phi_{-h,-h} = \phi_{0,0} - (a_1 + b_1)h + O(h^2) \end{cases} \quad (2.1)$$

then truncate the $O(h^2)$ terms to solve for (a_1, b_1) :

$$\begin{cases} a_1 = \frac{\phi_{0,-h} - \phi_{-h,-h}}{h} \\ b_1 = \frac{\phi_{0,0} - \phi_{0,-h}}{h} \end{cases} \quad (2.2)$$

We obtain a first order approximation of the Eikonal equation: $a_1^2 + b_1^2 = n^2$, or

$$(\phi_{0,-h} - \phi_{-h,-h})^2 + (\phi_{0,0} - \phi_{0,-h})^2 = n^2 h^2 \quad (2.3)$$

Set on all points, this is a large non-linear system. It is generally solved by an iterative method based on relaxation: fix the values at surrounding grid points and compute at the center point $\phi_{0,0}$ and iterate. To ensure the convergence of the iterative process and the convergence to the viscosity solution one must pick the triangle (or stencil) which satisfies a causality condition.

We now define the causality condition.

The discriminant of (2.3) is $\Delta = 4[n^2 h^2 - (\phi_{0,-h} - \phi_{-h,-h})^2]$. The nonnegativity of Δ requires

$$|\phi_{0,-h} - \phi_{-h,-h}| \leq n h \quad (2.4)$$

Two real roots are:

$$\phi_{0,0} = \phi_{0,-h} \pm \sqrt{n^2 h^2 - (\phi_{0,-h} - \phi_{-h,-h})^2}.$$

We choose

$$\phi_{0,0} = \phi_{0,-h} + \sqrt{n^2 h^2 - (\phi_{0,-h} - \phi_{-h,-h})^2} \quad (2.5)$$

according to the upwind property: $\phi_{0,0} \geq \phi_{0,-h}$. And we require $\nabla\phi$, the characteristic passing through C , intersects the base (AB) of the triangle (e.g., Figure 2.1) and is oriented from the

base (AB) toward C , that is, $\phi_{0,0} \geq \phi_{0,-h}$, $\phi_{0,0} \geq \phi_{-h,-h}$, $\phi_{0,-h} \geq \phi_{-h,-h}$, $\frac{\phi_{0,0} - \phi_{0,-h}}{h} \geq \frac{\phi_{0,-h} - \phi_{-h,-h}}{h}$. Therefore, the **Causality condition** is:

$$0 \leq \sqrt{2}(\phi_{0,-h} - \phi_{-h,-h}) \leq nh, \quad (2.6)$$

which also implies

$$\phi_{0,0} - \phi_{0,-h} \leq nh \leq \sqrt{2}(\phi_{0,0} - \phi_{0,-h}). \quad (2.7)$$

The causality condition is equivalent to the monotonicity of the fast sweeping method [18]. Therefore we can prove the convergence of the numerical solution to the viscosity solution [18, 1].

For this triangle we get a candidate solution ϕ_1 .

- if the causality condition (2.6) is satisfied, ϕ_1 is given by (2.5).
- else $\phi_1 = \min(\phi_A + n(C)|AC|, \phi_B + n(C)|BC|)$.

Here $|AC|$ (or $|BC|$) is the length of edge AC (or BC). The full 9-point stencil method consists in applying the same triangle solver to all 8 triangles and selecting the minimum value, i.e., $\phi_{0,0} = \min(\phi_1, \phi_2, \dots, \phi_8)$.

2.2. Fast sweeping method

The fast sweeping method consists in the Gauss-Seidel iterations of the 9-point solver over the grid in the following way,

Fast Sweeping Method:

- Initialization: To enforce the boundary condition, $\phi(x_{i,j}) = \phi_0$ for $x_{i,j} \in \Gamma$, assign exact values or interpolated values at those grid points on or near Γ . Assign large positive values elsewhere (start with a viscosity subsolution).
- Gauss-Seidel iteration with alternating orderings:
 - (1) $i = 1 : I, j = 1 : J$ (2) $i = 1 : I, j = J : 1$
 - (3) $i = I : 1, j = 1 : J$ (4) $i = I : 1, j = J : 1$

At each grid replace its old value with the newly computed value only if the old value is larger during the iterations.

2.3. First order gradient

The fast sweeping method is a monotone upwind scheme, so it is at most first order accurate [14]. We therefore expect the gradient to be $O(1)$. However, we observe here numerically that its gradient remains first order accurate away from singularities, e.g., kinks, in the maximum norm. This striking result has apparently not been reported or analyzed in the literature. We currently have partial theoretical results.

As can be seen from the following numerical results, both the solution and its numerical upwind gradient (at grid points) are first order in the maximum norm.

2.3.1. Uniform rectangular grid

In the following examples, we test both homogeneous and heterogeneous cases on a uniform rectangular mesh. The computational domain is $[0, 1] \times [0, 1]$.

Example 1: Homogeneous case: distance function to one source point outside a disk with fixed radius around the source. Boundary condition is assigned to the disk. Table 2.1 shows the results.

source point=(0.2, 0.2)						
Mesh	80x80	160x160	320x320	640x640	1280x1280	2560x2560
$E_{-\phi_x}$	0.0529669	0.0290819	0.0151099	0.0076870	0.0038753	0.0019454
Order	-	0.865	0.945	0.975	0.988	0.994
$E_{-\phi_y}$	0.0529669	0.0290819	0.0151099	0.0076870	0.0038753	0.0019454
Order	-	0.865	0.945	0.975	0.988	0.994
$E_{-\phi}$	0.0026306	0.0013020	0.0006480	0.0003237	0.0001617	0.0000808
Order	-	1.015	1.007	1.001	1.001	1.001
#iter	5	5	5	5	5	5
source point=(0.4999, 0.4997)						
Mesh	80x80	160x160	320x320	640x640	1280x1280	2560x2560
$E_{-\phi_x}$	0.0532476	0.0290716	0.0150992	0.0076804	0.0038716	0.0019476
Order	-	0.873	0.945	0.975	0.988	0.991
$E_{-\phi_y}$	0.0534723	0.0293191	0.0151945	0.0077276	0.0038951	0.0019470
Order	-	0.867	0.948	0.975	0.988	1.000
$E_{-\phi}$	0.0020922	0.0010269	0.0005093	0.0002537	0.0001266	0.0000632
Order	-	1.026	1.012	1.005	1.003	1.002
#iter	5	5	5	5	5	5

Table 2.1: Distance function on uniform rectangular meshes

Example 2: Heterogeneous case 1: Table 2.2 shows the results for $\phi(x, y) = 1 - e^{-[(x-x_0)^2+(y-y_0)^2]}$ where (x_0, y_0) is the source point.

Example 3: Heterogeneous case 2: Table 2.3 shows the results for $\phi(x, y) = \sin[\pi(1+x)]\sin(\pi y)$, where the source point is at $(0.5, 0.5)$.

2.3.2. Non-uniform rectangular grid

In the following examples, we test both homogeneous and heterogeneous cases on a non-uniform rectangular mesh. The non-uniform rectangular mesh is generated by perturbing a uniform rectangular mesh in the following way: given a uniform rectangular mesh of size h , then each grid point (x_i, y_i) is perturbed as $x_i = x_i + 0.3h * rand(-1, 1)$, $y_i = y_i + 0.3h * rand(-1, 1)$ where $rand(-1, 1)$ is a uniformly distributed random number. The computational domain is $[0, 1] \times [0, 1]$.

Example 1: Homogeneous case: distance function to one source point outside a disk with fixed radius around the source. Boundary condition is assigned to the disk. Table 2.4 shows the results.

Example 2: Heterogeneous case 1: Table 2.5 shows the results for $\phi(x, y) = 1 - e^{-[(x-x_0)^2+(y-y_0)^2]}$ where (x_0, y_0) is the source point.

$(x_0, y_0) = (0.2, 0.2)$						
Mesh	80x80	160x160	320x320	640x640	1280x1280	2560x2560
$E_{-\phi_x}$	0.0037678	0.0018813	0.0009400	0.0004698	0.0002348	0.0001174
Order	-	1.002	1.001	1.001	1.001	1.000
$E_{-\phi_y}$	0.0037678	0.0018813	0.0009400	0.0004698	0.0002348	0.0001174
Order	-	1.002	1.001	1.001	1.001	1.000
$E_{-\phi}$	0.0076833	0.0038544	0.0019304	0.0009660	0.0004831	0.0002416
Order	-	0.995	0.998	0.999	1.000	1.000
#iter	5	5	5	5	5	5
$(x_0, y_0) = (0.4999, 0.4997)$						
Mesh	80x80	160x160	320x320	640x640	1280x1280	2560x2560
$E_{-\phi_x}$	0.0124930	0.0062491	0.0031248	0.0015624	0.0007812	0.0003906
Order	-	0.999	1.000	1.000	1.000	1.000
$E_{-\phi_y}$	0.0124933	0.0062491	0.0031249	0.0015624	0.0007812	0.0003906
Order	-	0.999	1.000	1.000	1.000	1.000
$E_{-\phi}$	0.0075728	0.0037984	0.0019022	0.0009518	0.0004761	0.0001881
Order	-	0.995	0.998	0.999	0.999	1.340
#iter	5	5	5	5	5	5

Table 2.2: $\phi(x, y) = 1 - e^{-[(x-x_0)^2+(y-y_0)^2]}$ on uniform rectangular meshes

Mesh	80x80	160x160	320x320	640x640	1280x1280	2560x2560
$E_{-\phi_x}$	0.0613285	0.0307979	0.0154156	0.0077099	0.0038552	0.0019276
Order	-	0.994	0.998	1.000	1.000	1.000
$E_{-\phi_y}$	0.0613285	0.0307979	0.0154156	0.0077099	0.0038552	0.0019276
Order	-	0.994	0.998	1.000	1.000	1.000
$E_{-\phi}$	0.0216920	0.0111907	0.0056818	0.0028625	0.0014367	0.0007197
Order	-	0.955	0.978	0.989	0.995	0.997
#iter	5	5	5	5	5	5

Table 2.3: $\phi(x, y) = \sin[\pi(1+x)]\sin(\pi y)$, source (0.5, 0.5), on uniform rectangular meshes

Example 3: Heterogeneous case 2: Table 2.6 shows the results for $\phi(x, y) = \sin[\pi(1+x)]\sin(\pi y)$, where the source point is at (0.5, 0.5).

2.3.3. Triangular mesh

We test two cases on a triangular mesh to show the first order numerical gradient (computed from the upwind triangle) on triangular meshes. The sweeping strategies were designed in [18]. Figure 2.2 shows a triangular mesh generated by pdetool in matlab. The computational domain is $[0, 1] \times [0, 1]$. The following two examples show that the numerical gradient is still first order accurate in maximum norm.

Example 4: Homogeneous case: distance function to one source point outside a disk with fixed radius around the source. Boundary condition is assigned to the disk. Table 2.7 shows the results (N=nodes, T=triangles).

Example 5: Heterogeneous case: Table 2.8 shows the results for $\phi(x, y) = 1 - e^{-[(x-x_0)^2+(y-y_0)^2]}$. (N=nodes, T=triangles).

source point=(0.2, 0.2)						
Mesh	80x80	160x160	320x320	640x640	1280x1280	2560x2560
E_{ϕ_x}	0.0593022	0.0376018	0.0213904	0.0103294	0.0051558	0.0026291
Order	-	0.657	0.814	1.050	1.002	0.972
E_{ϕ_y}	0.0605857	0.0353263	0.0187774	0.0089362	0.0054762	0.0026653
Order	-	0.778	0.912	1.071	0.706	1.039
E_{ϕ}	0.0030526	0.0013952	0.0007047	0.0003471	0.0001720	0.0000860
Order	-	1.130	0.985	1.022	1.013	1.000
#iter	6	6	6	6	6	6
source point=(0.4999, 0.4997)						
Mesh	80x80	160x160	320x320	640x640	1280x1280	2560x2560
E_{ϕ_x}	0.0718876	0.0382733	0.0220579	0.0098473	0.0050713	0.0028083
Order	-	0.909	0.795	1.163	0.957	0.853
E_{ϕ_y}	0.0522860	0.0297888	0.0185513	0.0104084	0.0056949	0.0029713
Order	-	0.812	0.683	0.834	0.870	0.939
E_{ϕ}	0.0022527	0.0011291	0.0005544	0.0002729	0.0001353	0.0000676
Order	-	0.996	1.026	1.023	1.012	1.001
#iter	6	6	6	6	6	6

Table 2.4: Distance function on non-uniform rectangular meshes

$(x_0, y_0) = (0.2, 0.2)$						
Mesh	80x80	160x160	320x320	640x640	1280x1280	2560x2560
E_{ϕ_x}	0.0096266	0.0047490	0.0030360	0.0013689	0.0007363	0.0003752
Order	-	1.019	0.645	1.149	0.895	0.973
E_{ϕ_y}	0.0090814	0.0052429	0.0025928	0.0015664	0.0007839	0.0003728
Order	-	0.793	1.016	0.727	0.999	1.072
E_{ϕ}	0.0083566	0.0041136	0.0020730	0.0010290	0.0005137	0.0002573
Order	-	1.023	0.987	1.010	1.002	0.997
#iter	6	10	5	5	5	5
$(x_0, y_0) = (0.4999, 0.4997)$						
Mesh	80x80	160x160	320x320	640x640	1280x1280	2560x2560
E_{ϕ_x}	0.0108591	0.0044969	0.0020478	0.0010590	0.0005139	0.0003388
Order	-	1.272	1.135	0.951	1.043	0.601
E_{ϕ_y}	0.0100936	0.0056676	0.0022326	0.0014519	0.0005796	0.0002883
Order	-	0.833	1.344	0.621	1.325	1.007
E_{ϕ}	0.0082796	0.0041161	0.0020622	0.0010311	0.0005136	0.0002572
Order	-	1.008	0.997	1.000	1.005	0.998
#iter	5	6	6	6	6	6

Table 2.5: $\phi(x, y) = 1 - e^{-[(x-x_0)^2+(y-y_0)^2]}$ on non-uniform rectangular meshes

Mesh	80x80	160x160	320x320	640x640	1280x1280	2560x2560
E_{ϕ_x}	0.1265495	0.0543291	0.0310489	0.0164091	0.0089669	0.0047503
Order	-	1.220	0.807	0.920	0.872	0.917
E_{ϕ_y}	0.0943599	0.0608614	0.0303352	0.0165642	0.0090523	0.0047208
Order	-	0.633	1.005	0.873	0.872	0.939
E_{ϕ}	0.0245629	0.0122128	0.0061812	0.0030636	0.0015287	0.0007650
Order	-	1.008	0.982	1.013	1.003	0.999
#iter	6	6	6	6	6	6

Table 2.6: $\phi(x, y) = \sin[\pi(1+x)]\sin(\pi y)$, source (0.5, 0.5), on non-uniform rectangular meshes

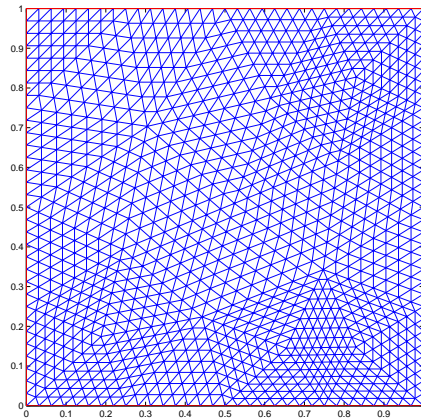


Fig. 2.2. 1217 nodes, 2304 triangles

source point=(0.5, 0.5)					
(N,T)	(321,576)	(1217,2304)	(4737,9216)	(18689,36864)	(74241,147456)
E_{ϕ_x}	0.2070550	0.1176064	0.0729189	0.0401607	0.0208485
Order	-	0.8489	0.7034	0.8691	0.9506
E_{ϕ_y}	0.2078222	0.1178848	0.0727418	0.0402041	0.0208150
Order	-	0.8509	0.7105	0.8640	0.9545
E_{ϕ}	0.0203922	0.0099214	0.0049357	0.0023715	0.0011487
Order	-	1.0812	1.0275	1.0681	1.0510
#iter	5	5	5	5	5

Table 2.7: Triangular mesh: distance function

$(x_0, y_0) = (0.5, 0.5)$					
(N, T)	(321,576)	(1217,2304)	(4737,9216)	(18689,36864)	(74241,147456)
E_{ϕ_x}	0.0578306	0.0444789	0.0216892	0.0110174	0.0055591
Order	-	0.3939	1.0569	0.9870	0.9918
E_{ϕ_y}	0.0577844	0.0444843	0.0216903	0.0110177	0.0055591
Order	-	0.3926	1.0570	0.9870	0.9918
E_{ϕ}	0.0414766	0.0203827	0.0098675	0.0048889	0.0024295
Order	-	1.0662	1.0676	1.0233	1.0139
#iter	5	5	5	5	5

Table 2.8: Triangular mesh: $\phi(x, y) = 1 - e^{-[(x-x_0)^2+(y-y_0)^2]}$

3. The compact upwind second order scheme

3.1. Adding the curvature of the wavefront in the discretization

The most popular high order schemes for hyperbolic type of PDEs are based on the ENO or WENO technique [16, 24, 10] which moves a second order stencil around a grid point to fetch the smoothest (kink free) approximation **direction by direction**. The approach followed here is different. Using the *Lagrangian structure* of the Eikonal equation we propose a pure upwind geometric second order correction to the first order scheme. The correction is based on a decomposition of the Hessian into normal and tangential components of the wavefront. By exploring the PDE, it turns out that only the tangential component, i.e. curvature of the wavefront, is needed for the correction. We will assume the first order solution ϕ^1 and numerical gradient (a_1, b_1) have been computed.

The second order Taylor expansion is:

$$\phi(X + \delta X) = \phi(X) + \delta X \cdot \nabla \phi(X) + \frac{1}{2} \langle \delta X, D^2 \phi(X) \cdot \delta X \rangle + O(\delta X^3). \quad (3.1)$$

We differentiate the Eikonal equation

$$D^2 \phi(X) \cdot \nabla \phi(X) = n(X) \nabla n(X)$$

and will use it to decompose the variation δX in a local coordinate system made of the Lagrangian direction and its orthogonal

$$\delta X = \frac{1}{n^2(X)} \{ (\delta X \cdot \nabla \phi) \nabla \phi + (\delta X \cdot \nabla \phi^\perp) \nabla \phi^\perp \}$$

Plugging this decomposition into the second order term of (3.1) yields

$$\begin{aligned} \frac{1}{2} \langle \delta X, D^2 \phi(X) \cdot \delta X \rangle = & \frac{1}{2n^4} (\delta X \cdot \nabla \phi)^2 \langle \nabla \phi, D^2 \phi \cdot \nabla \phi \rangle \\ & + \frac{1}{n^4} (\delta X \cdot \nabla \phi) (\delta X \cdot \nabla \phi^\perp) \langle \nabla \phi^\perp, D^2 \phi \cdot \nabla \phi \rangle \\ & + \frac{1}{2n^4} (\delta X \cdot \nabla \phi^\perp)^2 \langle \nabla \phi^\perp, D^2 \phi \cdot \nabla \phi^\perp \rangle \end{aligned} \quad (3.2)$$

We first restrict to the simpler homogeneous case $n \equiv 1$. Derivatives of n vanish and (3.1) becomes

$$\phi(X + \delta X) = \phi(X) + \delta X \cdot \nabla \phi(X) + \frac{1}{2} (\delta X \cdot \nabla \phi^\perp)^2 \langle \nabla \phi^\perp, D^2 \phi \cdot \nabla \phi^\perp \rangle + O(\delta X^3) \quad (3.3)$$

As $\langle \nabla \phi^\perp, D^2 \phi \cdot \nabla \phi^\perp \rangle$ is the curvature of the level sets of the phase (also called wavefronts), we will denote it as C . Neglecting the third order term, our new approximation of the phase (3.3) is of the form (denote $\nabla \phi = (a, b)$)

$$\phi(X + \delta X) = \phi(X) + \delta X \cdot (a, b) + \frac{C}{2n^4} (\delta X \cdot (-b, a))^2 \quad (3.4)$$

This is a parabolic "local curvature" correction to the piecewise linear approximation.

In the above second order approximation, there are three coefficients (a, b, C) to be determined. Suppose $a > 0, b > 0$, we choose the upwind quadrant and use the four grid points $(\{0, 0\}, \{-h, 0\}, \{0, -h\}, \{-h, -h\})$ for the expansion (we will come back on choice of stencils later).

$$\begin{cases} \phi_{-h,0} = \phi_{0,0} - h a + \frac{C}{2} (h b)^2 + O(h^3) \\ \phi_{0,-h} = \phi_{0,0} - h b + \frac{C}{2} (-h a)^2 + O(h^3) \\ \phi_{-h,-h} = \phi_{0,0} - h(a+b) + \frac{C}{2} (h(b-a))^2 + O(h^3) \end{cases} \quad (3.5)$$

Truncating the third order terms we can get a grid value approximation of the curvature C and eliminate the last equation for instance,

$$\text{line 1} + \text{line 2} - \text{line 3} \Rightarrow C = \frac{\phi_{0,-h} - \phi_{0,0} + \phi_{-h,0} - \phi_{-h,-h}}{h^2 a b} \quad (3.6)$$

The next natural steps should be

1. to solve for (a, b) as a function of $(\phi_{0,0}, \phi_{-h,0}, \phi_{0,-h}, \phi_{-h,-h})$ and replace in the Eikonal equation to write the numerical Hamiltonian.
2. relax: compute $\phi_{0,0}$ as a function of the values at grid points $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$.

This turns out to be a difficult nonlinear problem, involving the roots of a 6th order polynomial. Before explaining how to simplify this into a more tractable problem. We derive a natural and elegant interpretation of our scheme.

3.2. A non linear ray direction weighted approximation

Remembering that the Eikonal equation can be recast locally as $(a, b) = (n \cos \theta, n \sin \theta)$, we can use (3.5) (3.6) to write a local weighted centered finite difference (FD) formula in the upwind quadrant

$$\begin{cases} a = (1 - \frac{1}{2} \tan \theta) \frac{\phi_{0,0} - \phi_{-h,0}}{h} + \frac{1}{2} \tan \theta \frac{\phi_{0,-h} - \phi_{-h,-h}}{h} \\ b = (1 - \frac{1}{2} \tan^{-1} \theta) \frac{\phi_{0,0} - \phi_{0,-h}}{h} + \frac{1}{2} \tan^{-1} \theta \frac{\phi_{-h,0} - \phi_{-h,-h}}{h} \end{cases} \quad (3.7)$$

This is not a practical solution formula as θ depends on (a, b) but it shows that we average the local first order FD according to the Lagrangian direction. For instance if the ray makes a $\frac{\pi}{4}$ angle with the x axis, i.e. exactly cuts in half the quadrant, then we use an arithmetic average of the FD derivatives of the four grid points of this quadrant. If the ray approaches an axis, then the weight is non-linearly adjusted by the $\tan \theta$ functions.

If the ray direction is on a grid line (either the x or y axis) then the cotangent or tangent coefficient is infinite. In this case C is not guaranteed to be determined uniquely by system (3.1) (after truncation) as can be seen by replacing (a, b) by $(0, 1)$ or $(1, 0)$. If rays approach the grid lines the method is unstable. A natural way to avoid this instability is to use a different stencil according to the upwind direction. This is explained in section 4. We now focus on this particular stencil and assume the ray direction stays away from the grid lines ($\theta = 0, \frac{\pi}{2}$).

3.3. Simplification of the second order terms

In order to simplify our local solver and obtain a tractable scheme we remember that the first order scheme

1. determines the upwind direction.

2. does so with first order accuracy.

Assume (a_1, b_1) is in the positive x and y directions, we can therefore work with the upwind quadrant $(\{0, 0\}, \{-h, 0\}, \{0, -h\}, \{-h, -h\})$. Using remark that $(a_1, b_1) = (a, b) + O(h)$, we notice that replacing (a, b) in the quadratic terms of the Taylor expansions (3.5) with (a_1, b_1) preserves second order accuracy

$$\begin{cases} \phi_{-h,0} = \phi_{0,0} - h a + \frac{C}{2} (h b_1)^2 + O(h^3) \\ \phi_{0,-h} = \phi_{0,0} - h b + \frac{C}{2} (-h a_1)^2 + O(h^3) \\ \phi_{-h,-h} = \phi_{0,0} - h(a+b) + \frac{C}{2} (h(b_1 - a_1))^2 + O(h^3) \end{cases} \quad (3.8)$$

We can again eliminate C with formula (3.6) where (a, b) is replaced by (a_1, b_1) . Then, (3.7) simplifies to

$$\begin{cases} a = (1 - \frac{1}{2} \tan \theta_1) \frac{\phi_{0,0} - \phi_{-h,0}}{h} + \frac{1}{2} \tan \theta_1 \frac{\phi_{0,-h} - \phi_{-h,-h}}{h} \\ b = (1 - \frac{1}{2} \tan^{-1} \theta_1) \frac{\phi_{0,0} - \phi_{0,-h}}{h} + \frac{1}{2} \tan^{-1} \theta_1 \frac{\phi_{-h,0} - \phi_{-h,-h}}{h} \end{cases} \quad (3.9)$$

where θ_1 is the first order angle given by the first order scheme and therefore $\tan \theta_1 = \frac{b_1}{a_1}$. We can now write our second order numerical Hamiltonian (restricted to the upwind quadrant)

$$g(\phi_{0,0}, \phi_{0,-h}, \phi_{-h,0}, \phi_{-h,-h}) = \sqrt{a^2 + b^2} - n_{0,0} = 0 \quad (3.10)$$

where (a, b) is given by (3.9).

This numerical scheme has the following properties:

1. Consistency of second order local truncation error.
2. Stability as long as θ_1 stays away from grid lines.
3. Monotonicity, as a second order scheme it is of course non-monotone (it is easily checked that g cannot be simultaneously a decreasing function of $\phi_{0,-h}$ and $\phi_{-h,0}$ at the same time).

We need to address the solution of the discrete system. As for the first order monotone scheme we use relaxation. We need to be able to solve for $\phi_{0,0}^{k+1}$

$$g(\phi_{0,0}^{k+1}, \phi_{0,-h}^k, \phi_{-h,0}^k, \phi_{-h,-h}^k) = 0 \quad (3.11)$$

This boils down to finding the roots of a second order polynomial. For simplicity we omit the k superscripts, the equation is

$$a^2 + b^2 = n_{0,0}^2$$

where (a, b) are given in (3.9). This can be rewritten (carefully) as

$$c_1 \phi_{0,0}^2 + c_2 \phi_{0,0} + c_3 = 0 \quad (3.12)$$

where (c_1, c_2, c_3) are easily computed and depend on the values at other grid points and on (a_1, b_1) .

Let us study more precisely this quadratic equation:

$$\begin{aligned}
A &= \frac{b_1}{2a_1}(\phi_{0,-h} - \phi_{-h,-h}) - (1 - \frac{b_1}{2a_1})\phi_{-h,0} \\
B &= \frac{a_1}{2b_1}(\phi_{-h,0} - \phi_{-h,-h}) - (1 - \frac{a_1}{2b_1})\phi_{0,-h} \\
c_1 &= (1 - \frac{b_1}{2a_1})^2 + (1 - \frac{a_1}{2b_1})^2 \\
c_2 &= 2(1 - \frac{b_1}{2a_1})A + 2(1 - \frac{a_1}{2b_1})B \\
c_3 &= B^2 + A^2 - h^2 n^2
\end{aligned}$$

The discriminant can be reduced to

$$\Delta = c_1 h^2 n^2 - \{(2 - \frac{b_1}{a_1})B - (2 - \frac{a_1}{b_1})A\}^2 \quad (3.13)$$

So we get real roots when $\|(2 - \frac{b_1}{a_1})B - (2 - \frac{a_1}{b_1})A\| \leq \sqrt{c_1} h n$ which can be reduced to

$$\left\| \frac{(2 - \frac{b_1}{a_1})\phi_{-h,0} - (2 - \frac{a_1}{b_1})\phi_{0,-h} + (\frac{b_1}{a_1} - \frac{a_1}{b_1})\phi_{-h,-h}}{\sqrt{(1 - \frac{b_1}{2a_1})^2 + (1 - \frac{a_1}{2b_1})^2}} \right\| \leq n h \quad (3.14)$$

The roots are

$$roots = \frac{-c_2 \pm \sqrt{\Delta}}{2c_1}$$

However we find that the part

$$\begin{aligned}
\frac{-c_2}{2c_1} &= \frac{1}{(1 - \frac{b_1}{2a_1})^2 + (1 - \frac{a_1}{2b_1})^2} (\alpha_{0,-h}\phi_{0,-h} + \alpha_{-h,0}\phi_{-h,0} + \alpha_{-h,-h}\phi_{-h,-h}) \\
\alpha_{0,-h} &= -(1 - \frac{b_1}{2a_1})\frac{b_1}{2a_1} + (1 - \frac{a_1}{2b_1})^2 \\
\alpha_{-h,0} &= -(1 - \frac{a_1}{2b_1})\frac{a_1}{2b_1} + (1 - \frac{b_1}{2a_1})^2 \\
\alpha_{-h,-h} &= (1 - \frac{a_1}{2b_1})\frac{a_1}{2b_1} + (1 - \frac{b_1}{2a_1})\frac{b_1}{2a_1}
\end{aligned}$$

is a weighted average of the upwind values (remark $\alpha_{-h,0} + \alpha_{0,-h} + \alpha_{-h,-h} = (1 - \frac{b_1}{2a_1})^2 + (1 - \frac{a_1}{2b_1})^2$) and $\sqrt{\Delta}$ is of $O(h)$. So as we march in the upwind direction, we must pick the larger root $\frac{-c_2 + \sqrt{\Delta}}{2c_1}$, noting that $\frac{1}{(1 - \frac{b_1}{2a_1})^2 + (1 - \frac{a_1}{2b_1})^2} (\alpha_{-h,0}\{-h, 0\} + \alpha_{0,-h}\{0, -h\} + \alpha_{-h,-h}\{-h, -h\}) = \frac{-h}{(1 - \frac{b_1}{2a_1})^2 + (1 - \frac{a_1}{2b_1})^2} \{1 - \frac{b_1}{2a_1}, 1 - \frac{a_1}{2b_1}\}$ is under the line $y = -\frac{a_1}{b_1}x$ which is orthogonal to $\{a_1, b_1\}$, the wavefront reaches this point before reaching $\{0, 0\}$.

4. General case and choice of stencils

Here we discuss about more general cases and explain how to select different stencils when (1) either a_1 or b_1 vanishes (ray direction on a grid line), (2) there are shocks/kinks.

4.1. The heterogeneous case: variable index of refraction

We proceed as in the previous section with equation (5) and (6) to get

$$\begin{aligned}
\phi(x + \delta x) &= \phi(x) + \delta x \cdot \nabla \phi(x) \\
&+ \frac{1}{2n^4} (\delta x \cdot \nabla \phi)^2 < \nabla \phi, D^2 \phi \cdot \nabla \phi > \\
&+ \frac{1}{n^4} (\delta x \cdot \nabla \phi) (\delta x \cdot \nabla \phi^\perp) < \nabla \phi^\perp, D^2 \phi \cdot \nabla \phi > \\
&+ \frac{C}{2n^4} (\delta x \cdot \nabla \phi^\perp)^2
\end{aligned} \tag{4.1}$$

We note again $C = \langle \nabla \phi^\perp, D^2 \phi \cdot \nabla \phi^\perp \rangle$ is the curvature of the level set of the wavefront. This will be replaced in the formula above. The other terms involving the Hessian matrix of ϕ can be eliminated using the index of refraction n and its gradient $\nabla n = (n_x, n_y)$ by taking the gradient of the Eikonal equation

$$D^2 \phi \cdot \nabla \phi = n \nabla n$$

These terms were vanishing in the previous section where we assumed the constant index of refraction.

Denoting $\nabla \phi = (a, b)$, we proceed as in section 3 and remark that (4.1) has three unknowns: (a, b, C) which determine a parabolic second order approximation of the phase. Depending on the the upwind direction given by the first order gradient (a_1, b_1) we will use different stencils to compute (a, b, C) .

Let us again assume that $a_1 > 0$ and $b_1 > 0$. In the upwind quadrant, e.g. Figure 4.1(b) Region 1, with stencil $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$, we do second order Taylor expansion:

$$\left\{ \begin{aligned}
\phi_{-h,0} &= \phi_{0,0} - ah + \frac{h^2}{2n^3} [(a^3 + 2ab^2)n_x - a^2bn_y] + \frac{h^2b^2}{2n^4}C \\
\phi_{0,-h} &= \phi_{0,0} - bh + \frac{h^2}{2n^3} [-ab^2n_x + (b^3 + 2a^2b)n_y] + \frac{h^2a^2}{2n^4}C \\
\phi_{-h,-h} &= \phi_{0,0} - (a+b)h + \frac{h^2}{2n^3} [(a^3 + ab^2 + 2b^3)n_x + (b^3 + a^2b + 2a^3)n_y] \\
&\quad + \frac{h^2(b-a)^2}{2n^4}C
\end{aligned} \right. \tag{4.2}$$

An upwind scheme should use the neighboring values, e.g. at grid points $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$ to update the value at the center point $\{0, 0\}$. So assuming these values are given and already second order accurate we can solve system (4.2) with the Eikonal equation $a^2 + b^2 = n^2$ to get $(a, b, C, \phi_{0,0})$. In other words with the algebraic trick above, we only need one additional point to the first order stencil to get second order accuracy.

Similarly as in the case for the constant index of refraction (section 3) this system is difficult to solve and boils down to a 6th order polynomial in $\phi_{0,0}$. So we replace (a, b) in high order terms of (4.2) with first order approximation (a_1, b_1)

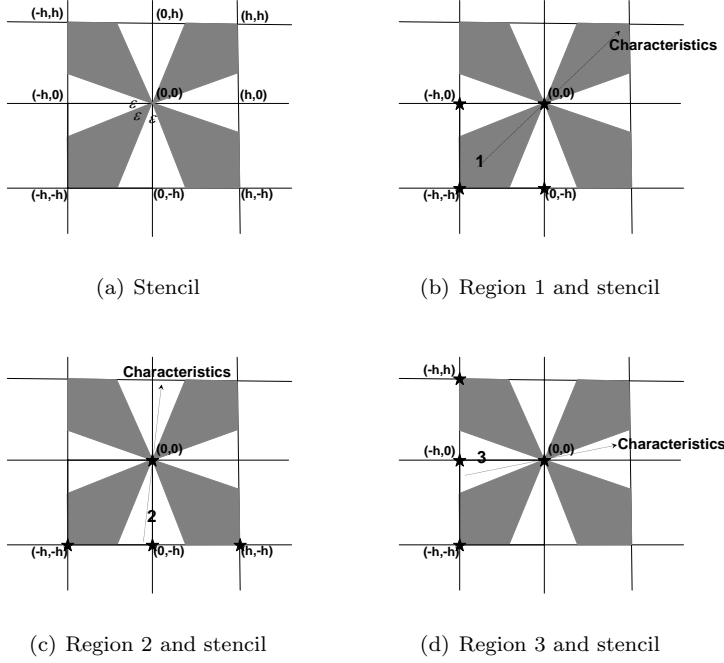


Fig. 4.1. Upwind quadrant, different regions and corresponding stencils (marked as \star)

$$\left\{ \begin{array}{l} \phi_{-h,0} = \phi_{0,0} - ah + \frac{h^2}{2n^3} [(a_1^3 + 2a_1b_1^2)n_x - a_1^2b_1n_y] + \frac{h^2b_1^2}{2n^4}C \\ \phi_{0,-h} = \phi_{0,0} - bh + \frac{h^2}{2n^3} [-a_1b_1^2n_x + (b_1^3 + 2a_1^2b_1)n_y] + \frac{h^2a_1^2}{2n^4}C \\ \phi_{-h,-h} = \phi_{0,0} - (a+b)h + \frac{h^2}{2n^3} [(a_1^3 + a_1b_1^2 + 2b_1^3)n_x + (b_1^3 + a_1^2b_1 + 2a_1^3)n_y] \\ \quad + \frac{h^2(b_1 - a_1)^2}{2n^4}C \end{array} \right. \quad (4.3)$$

Now, we can easily solve the new system (4.3) for (a, b, C) ,

$$\left\{ \begin{array}{l} a = \frac{1}{h} \left\{ \left(1 - \frac{b_1}{2a_1}\right) \phi_{0,0} - \phi_{-h,0} \right\} + \frac{hnn_x}{2a_1} + \frac{b_1}{2a_1} \frac{F0}{h} \\ b = \frac{1}{h} \left\{ \left(1 - \frac{a_1}{2b_1}\right) \phi_{0,0} - \phi_{0,-h} \right\} + \frac{hnn_y}{2b_1} + \frac{a_1}{2b_1} \frac{F0}{h} \\ C = \frac{(F0 - \phi_{0,0})n^4}{a_1b_1h^2} + \frac{n(b_1^3n_x + a_1^3n_y)}{a_1b_1} \end{array} \right. \quad (4.4)$$

and

$$\phi_{-h,0} = \left(1 - \frac{b_1}{2a_1}\right) \phi_{0,0} - ah + \frac{h^2nn_x}{2a_1} + \frac{b_1}{2a_1} F0 \quad (4.5a)$$

$$\phi_{0,-h} = \left(1 - \frac{a_1}{2b_1}\right) \phi_{0,0} - bh + \frac{h^2nn_y}{2b_1} + \frac{a_1}{2b_1} F0 \quad (4.5b)$$

$$\phi_{-h,-h} = \left\{ 1 - \frac{(b_1 - a_1)^2}{2a_1b_1} \right\} \phi_{0,0} - (a+b)h + \frac{h^2(b_1nn_x + a_1nn_y)}{2a_1b_1} + \frac{(b_1 - a_1)^2}{2a_1b_1} F0 \quad (4.5c)$$

Where $F0 = \phi_{-h,0} + \phi_{0,-h} - \phi_{-h,-h}$.

Then, the numerical Hamiltonian $a^2 + b^2 = n^2$ is only a quadratic polynomial for $\phi_{0,0}$:

$$\begin{aligned} & [(1 - \frac{b_1}{2a_1})\phi_{0,0} - \phi_{-h,0} + \frac{h^2nn_x}{2a_1} + \frac{b_1}{2a_1}F0]^2 \\ & + [(1 - \frac{a_1}{2b_1})\phi_{0,0} - \phi_{0,-h} + \frac{h^2nn_y}{2b_1} + \frac{a_1}{2b_1}F0]^2 = n^2h^2 \end{aligned} \quad (4.6)$$

Rewrite (4.6) as

$$c_1\phi_{0,0}^2 + c_2\phi_{0,0} + c_3 = 0$$

where

$$\begin{aligned} c_1 &= g_1^2 + g_3^2; \\ c_2 &= 2(g_1g_2 + g_3g_4); \\ c_3 &= g_2^2 + g_4^2 - n^2h^2; \\ g_1 &= 1 - \frac{b_1}{2a_1}; \\ g_2 &= -\phi_{-h,0} + \frac{h^2nn_x}{2a_1} + \frac{b_1}{2a_1}F0; \\ g_3 &= 1 - \frac{a_1}{2b_1}; \\ g_4 &= -\phi_{0,-h} + \frac{h^2nn_y}{2b_1} + \frac{a_1}{2b_1}F0. \end{aligned}$$

The discriminant of (4.6) is:

$$\begin{aligned} \Delta &= c_2^2 - 4c_1c_3 \\ &= 4[(g_1g_2 + g_3g_4)^2 - (g_1^2 + g_3^2)(g_2^2 + g_4^2 - n^2h^2)] \\ &= 4[(g_1^2 + g_3^2)n^2h^2 - (g_1g_4 - g_2g_3)^2] \end{aligned}$$

The nonnegativity of Δ implies $\frac{\|g_1g_4 - g_2g_3\|}{\sqrt{g_1^2 + g_3^2}} \leq nh$, which is

$$\left\| \frac{(1 - \frac{b_1}{2a_1})\phi_{-h,0} - (1 - \frac{a_1}{2b_1})\phi_{0,-h} + (\frac{b_1}{2a_1} - \frac{a_1}{2b_1})\phi_{-h,-h} + \frac{nh^2}{4}(\frac{2n_y - n_x}{b_1} - \frac{n_y - 2n_x}{a_1})}{\sqrt{(1 - \frac{b_1}{2a_1})^2 + (1 - \frac{a_1}{2b_1})^2}} \right\| \leq nh \quad (4.7)$$

Two real roots are:

$$roots = \frac{-c_2 \pm \sqrt{\Delta}}{2c_1}$$

with

$$\begin{aligned} \frac{-c_2}{2c_1} &= \frac{1}{(1 - \frac{b_1}{2a_1})^2 + (1 - \frac{a_1}{2b_1})^2} (\alpha_{-h,0}\phi_{-h,0} + \alpha_{0,-h}\phi_{0,-h} + \alpha_{-h,-h}\phi_{-h,-h} + \beta_{0,0}) \\ \alpha_{-h,0} &= -(1 - \frac{a_1}{2b_1})\frac{a_1}{2b_1} + (1 - \frac{b_1}{2a_1})^2 \\ \alpha_{0,-h} &= -(1 - \frac{b_1}{2a_1})\frac{b_1}{2a_1} + (1 - \frac{a_1}{2b_1})^2 \\ \alpha_{-h,-h} &= (1 - \frac{a_1}{2b_1})\frac{a_1}{2b_1} + (1 - \frac{b_1}{2a_1})\frac{b_1}{2a_1} \\ \beta_{0,0} &= \frac{h^2n}{2} [(1 - \frac{b_1}{2a_1})\frac{n_x}{a_1} + (1 - \frac{a_1}{2b_1})\frac{n_y}{b_1}] \end{aligned} \quad (4.8)$$

We choose the larger root: $\frac{-c_2 + \sqrt{\Delta}}{2c_1}$.

4.2. Rays on grid lines

In (4.4), $\frac{b_1}{a_1}$ or $\frac{a_1}{b_1}$ may blow up if the ray falls close to the grid lines, which results in instability.

In order to resolve this problem, we separate the neighborhood (local mesh) into eight regions (e.g. Figure 4.1(a), grey and blank, $\epsilon = \frac{\pi}{4}$). And choose different stencils according to where (a_1, b_1) falls to approximate (a, b, C) .

In region 1, e.g. Figure 4.1(b), we use stencil $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$, and the formulas are as above.

In region 2, e.g. Figure 4.1(c), we use stencil $(\{0, -h\}, \{-h, -h\}, \{h, -h\})$ instead.

$$\left\{ \begin{array}{l} \phi_{0,-h} = \phi_{0,0} - bh + \frac{h^2}{2n^3}[-a_1 b_1^2 n_x + (b_1^3 + 2a_1^2 b_1) n_y] + \frac{h^2 a_1^2}{2n^4} C \\ \phi_{-h,-h} = \phi_{0,0} - (a+b)h + \frac{h^2}{2n^3}[(a_1^3 + a_1 b_1^2 + 2b_1^3) n_x \\ \quad + (b_1^3 + a_1^2 b_1 + 2a_1^3) n_y] + \frac{h^2 (b_1 - a_1)^2}{2n^4} C \\ \phi_{h,-h} = \phi_{0,0} + (a-b)h + \frac{h^2}{2n^3}[(a_1^3 + a_1 b_1^2 - 2b_1^3) n_x \\ \quad + (b_1^3 + a_1^2 b_1 - 2a_1^3) n_y] + \frac{h^2 (b_1 + a_1)^2}{2n^4} C \end{array} \right. \quad (4.9)$$

Solve for (a, b, C) ,

$$\left\{ \begin{array}{l} a = \frac{\phi_{0,-h} - \phi_{-h,-h}}{h} + \frac{nhn_x}{b_1} - \frac{2a_1 - b_1}{2b_1} \frac{F0}{h} \\ b = \frac{\phi_{0,0} - \phi_{0,-h}}{h} + \frac{nh}{2} \left(-\frac{a_1 n_x}{b_1^2} + \frac{n_y}{b_1} \right) + \frac{a_1^2}{2b_1^2} \frac{F0}{h} \\ C = \frac{F0n^4}{b_1^2 h^2} - \frac{n[(a_1^3 + 2a_1 b_1^2) n_x - a_1^2 b_1 n_y]}{b_1^2} \end{array} \right. \quad (4.10)$$

and

$$\phi_{0,-h} = \phi_{0,0} - bh + \frac{nh^2}{2} \left(-\frac{a_1 n_x}{b_1^2} + \frac{n_y}{b_1} \right) + \frac{a_1^2}{2b_1^2} F0 \quad (4.11a)$$

$$\phi_{-h,-h} = \phi_{0,0} - (a+b)h + \frac{nh^2}{2} \left[\frac{(2b_1 - a_1)n_x}{b_1^2} + \frac{n_y}{b_1} \right] + \frac{(b_1 - a_1)^2}{2b_1^2} F0 \quad (4.11b)$$

$$\phi_{h,-h} = \phi_{0,0} + (a-b)h + \frac{nh^2}{2} \left[-\frac{(a_1 + 2b_1)n_x}{b_1^2} + \frac{n_y}{b_1} \right] + \frac{(a_1 + b_1)^2}{2b_1^2} F0 \quad (4.11c)$$

Where $F0 = \phi_{-h,-h} + \phi_{h,-h} - 2\phi_{0,-h}$.

Then the numerical Hamiltonian $a^2 + b^2 = n^2$ is,

$$\begin{aligned} & [\phi_{0,-h} - \phi_{-h,-h} + \frac{h^2 n n_x}{b_1} - \frac{2a_1 - b_1}{2b_1} F0]^2 \\ & + [\phi_{0,0} - \phi_{0,-h} + \frac{h^2 n}{2} \left(-\frac{a_1 n_x}{b_1^2} + \frac{n_y}{b_1} \right) + \frac{a_1^2}{2b_1^2} F0]^2 = n^2 h^2 \end{aligned} \quad (4.12)$$

Rewrite (4.12) as

$$c_1\phi_{0,0}^2 + c_2\phi_{0,0} + c_3 = 0$$

with

$$\begin{aligned} c_1 &= g_1^2 + g_3^2; \\ c_2 &= 2(g_1g_2 + g_3g_4); \\ c_3 &= g_2^2 + g_4^2 - n^2h^2; \\ g_1 &= 0; \\ g_2 &= \phi_{0,-h} - \phi_{h,-h} + \frac{h^2nn_x}{b_1} - \frac{2a_1 - b_1}{2b_1}F0; \\ g_3 &= 1; \\ g_4 &= -\phi_{0,-h} + \frac{h^2n}{2}\left(\frac{-a_1n_x}{b_1^2} + \frac{n_y}{b_1}\right) + \frac{a_1^2}{2b_1^2}F0. \end{aligned}$$

The discriminant of (4.12) is:

$$\begin{aligned} \Delta &= c_2^2 - 4c_1c_3 \\ &= 4[(g_1g_2 + g_3g_4)^2 - (g_1^2 + g_3^2)(g_2^2 + g_4^2 - n^2h^2)] \\ &= 4[(g_1^2 + g_3^2)n^2h^2 - (g_1g_4 - g_2g_3)^2] \\ &= 4[n^2h^2 - g_2^2] \end{aligned}$$

The nonnegativity of Δ implies $\|g_2\| \leq nh$, that is

$$\left\| \frac{2a_1}{b_1}\phi_{0,-h} - \left(\frac{a_1}{b_1} + \frac{1}{2}\right)\phi_{-h,-h} - \left(\frac{a_1}{b_1} - \frac{1}{2}\right)\phi_{h,-h} + \frac{h^2nn_x}{b_1} \right\| \leq nh \quad (4.13)$$

Two real roots are:

$$roots = \frac{-c_2 \pm \sqrt{\Delta}}{2c_1}$$

with

$$\begin{aligned} \frac{-c_2}{2c_1} &= \alpha_{0,-h}\phi_{0,-h} + \alpha_{-h,-h}\phi_{-h,-h} + \alpha_{h,-h}\phi_{h,-h} + \beta_{0,0} \\ \alpha_{0,-h} &= 1 + \frac{a_1^2}{b_1^2} \\ \alpha_{-h,-h} &= -\frac{a_1^2}{2b_1^2} \\ \alpha_{h,-h} &= -\frac{a_1^2}{2b_1^2} \\ \beta_{0,0} &= -\frac{h^2n}{2}\left(\frac{-a_1n_x}{b_1^2} + \frac{n_y}{b_1}\right) \end{aligned} \quad (4.14)$$

We choose the larger root: $\frac{-c_2 + \sqrt{\Delta}}{2c_1}$.

In region 3, e.g. Figure 4.1(d), we use stencil $(\{-h, 0\}, \{-h, -h\}, \{-h, h\})$ instead.

$$\left\{ \begin{array}{l} \phi_{-h,0} = \phi_{0,0} - ah + \frac{h^2}{2n^3}[(a_1^3 + 2a_1b_1^2)n_x - a_1^2b_1n_y] + \frac{h^2b_1^2}{2n^4}C \\ \phi_{-h,-h} = \phi_{0,0} - (a+b)h + \frac{h^2}{2n^3}[(a_1^3 + a_1b_1^2 + 2b_1^3)n_x \\ \quad + (b_1^3 + a_1^2b_1 + 2a_1^3)n_y] + \frac{h^2(b_1 - a_1)^2}{2n^4}C \\ \phi_{-h,h} = \phi_{0,0} + (b-a)h + \frac{h^2}{2n^3}[(a_1^3 + a_1b_1^2 - 2b_1^3)n_x \\ \quad + (b_1^3 + a_1^2b_1 - 2a_1^3)n_y] + \frac{h^2(b_1 + a_1)^2}{2n^4}C \end{array} \right. \quad (4.15)$$

Solve for (a, b, C) ,

$$\left\{ \begin{array}{l} a = \frac{\phi_{0,0} - \phi_{-h,0}}{h} + \frac{nh}{2} \left(\frac{n_x}{a_1} - \frac{b_1n_y}{a_1^2} \right) + \frac{b_1^2}{2a_1^2} \frac{F0}{h} \\ b = \frac{\phi_{-h,0} - \phi_{-h,-h}}{h} + \frac{nhn_y}{a_1} - \frac{2b_1 - a_1}{2a_1} \frac{F0}{h} \\ C = \frac{F0n^4}{a_1^2h^2} - \frac{n[-a_1b_1^2n_x + (b_1^3 + 2a_1^2b_1)n_y]}{a_1^2} \end{array} \right. \quad (4.16)$$

and

$$\phi_{-h,0} = \phi_{0,0} - ah + \frac{nh^2}{2} \left(\frac{n_x}{a_1} - \frac{b_1n_y}{2a_1^2} \right) + \frac{b_1^2}{2a_1^2} F0 \quad (4.17a)$$

$$\phi_{-h,-h} = \phi_{0,0} - (a+b)h + \frac{nh^2}{2} \left[\frac{n_x}{a_1} + \frac{(2a_1 - b_1)n_y}{a_1^2} \right] + \frac{(b_1 - a_1)^2}{2a_1^2} F0 \quad (4.17b)$$

$$\phi_{-h,h} = \phi_{0,0} + (b-a)h + \frac{nh^2}{2} \left[\frac{n_x}{a_1} - \frac{(2a_1 + b_1)n_y}{a_1^2} \right] + \frac{(a_1 + b_1)^2}{2a_1^2} F0 \quad (4.17c)$$

Where $F0 = \phi_{-h,-h} + \phi_{-h,h} - 2\phi_{-h,0}$.

Then the numerical Hamiltonian $a^2 + b^2 = n^2$ is,

$$\begin{aligned} & [\phi_{0,0} - \phi_{-h,0} + \frac{h^2n}{2} \left(-\frac{b_1n_y}{a_1^2} + \frac{n_x}{a_1} \right) + \frac{b_1^2}{2a_1^2} F0]^2 \\ & + [\phi_{-h,0} - \phi_{-h,-h} + \frac{h^2nn_y}{a_1} - \frac{2b_1 - a_1}{2a_1} F0]^2 = n^2h^2 \end{aligned} \quad (4.18)$$

Rewrite (4.18) as

$$c_1\phi_{0,0}^2 + c_2\phi_{0,0} + c_3 = 0$$

with

$$c_1 = g_1^2 + g_3^2;$$

$$c_2 = 2(g_1g_2 + g_3g_4);$$

$$c_3 = g_2^2 + g_4^2 - n^2h^2;$$

$$g_1 = 1;$$

$$g_2 = -\phi_{-h,0} + \frac{h^2n}{2} \left(-\frac{b_1n_y}{a_1^2} + \frac{n_x}{a_1} \right) + \frac{b_1^2}{2a_1^2} F0;$$

$$g_3 = 0;$$

$$g_4 = \phi_{-h,0} - \phi_{-h,-h} + \frac{h^2nn_y}{a_1} - \frac{2b_1 - a_1}{2a_1} F0.$$

The discriminant of (4.18) is:

$$\begin{aligned}
\Delta &= c_2^2 - 4c_1c_3 \\
&= 4[(g_1g_2 + g_3g_4)^2 - (g_1^2 + g_3^2)(g_2^2 + g_4^2 - n^2h^2)] \\
&= 4[(g_1^2 + g_3^2)n^2h^2 - (g_1g_4 - g_2g_3)^2] \\
&= 4[n^2h^2 - g_4^2]
\end{aligned}$$

The nonnegativity of Δ implies $\|g_4\| \leq nh$, that is

$$\left\| \frac{2b_1}{a_1}\phi_{-h,0} - \left(\frac{b_1}{a_1} + \frac{1}{2}\right)\phi_{-h,-h} - \left(\frac{b_1}{a_1} - \frac{1}{2}\right)\phi_{-h,h} + \frac{h^2nn_y}{a_1} \right\| \leq nh \quad (4.19)$$

Two real roots are:

$$roots = \frac{-c_2 \pm \sqrt{\Delta}}{2c_1}$$

with

$$\begin{aligned}
\frac{-c_2}{2c_1} &= \alpha_{-h,0}\phi_{-h,0} + \alpha_{-h,-h}\phi_{-h,-h} + \alpha_{-h,h}\phi_{-h,h} + \beta_{0,0} \\
\alpha_{-h,0} &= 1 + \frac{b_1^2}{a_1^2} \\
\alpha_{-h,-h} &= -\frac{b_1^2}{2a_1^2} \\
\alpha_{-h,h} &= -\frac{b_1^2}{2a_1^2} \\
\beta_{0,0} &= -\frac{h^2n}{2} \left(\frac{-b_1n_y}{a_1^2} + \frac{n_x}{a_1} \right)
\end{aligned} \quad (4.20)$$

We choose the larger root: $\frac{-c_2 + \sqrt{\Delta}}{2c_1}$.

In (4.4), (4.10) and (4.16), we avoid the instability near the grid lines, that is, neither $\frac{b_1}{a_1}$ nor $\frac{a_1}{b_1}$ blows up if we choose the appropriate stencil. Altogether there are eight different stencils depending on (a_1, b_1) in 2D. The other cases are similar to the derivations above.

The stencil $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$, $(\{0, -h\}, \{-h, -h\}, \{h, -h\})$ or $(\{-h, 0\}, \{-h, -h\}, \{-h, h\})$ chosen according to (a_1, b_1) is a good choice when the center point is in the smooth region. However, when on a shock/kink, it may not be a good choice.

4.3. Stencil near singularity

The above choice of stencils is valid away from singularities, e.g., shocks/kinks. Near a shock, one should avoid using stencils across the shock to approximate the second derivatives. Also when tracing back along the characteristic the current grid point value should be interpolated from neighboring stencils, i.e., the CFL (Courant-Friedrichs-Lewy) condition, for stability. In summary we need to choose an upwind stencil which (1) provides smoother approximations of second derivatives, and (2) satisfies the CFL condition. In particular we use numerical approximation of $|\phi_{xx}|$, $|\phi_{yy}|$ or $|\phi_{xx}| + |\phi_{yy}|$ from the upwind direction to measure the smoothness. Again assuming we have first order approximation of the gradient $a_1 > 0, b_1 > 0$, here is our general **Stencil-choosing criterion**: (see Figure 4.2),

1. when the ray direction is in region 1,

- if $\frac{b_1}{a_1} < 1$.
 - If $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$ has been updated,
 $Dc = |\phi_{0,-h}^1 - 2\phi_{-h,-h}^1 + \phi_{-2h,-h}^1| + |\phi_{-h,0}^1 - 2\phi_{-h,-h}^1 + \phi_{-h,-2h}^1|$.
 else, $Dc = \infty$.
 - If $(\{-h, 0\}, \{-h, -h\}, \{-h, h\})$ has been updated,
 $Da = |\phi_{0,0}^1 - 2\phi_{-h,0}^1 + \phi_{-2h,0}^1| + |\phi_{-h,h}^1 - 2\phi_{-h,0}^1 + \phi_{-h,-h}^1|$,
 else, $Da = \infty$.
 - If $Dc \leq Da$, choose $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$,
 else, choose $(\{-h, 0\}, \{-h, -h\}, \{-h, h\})$.
- if $\frac{b_1}{a_1} > 1$.
 - If $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$ has been updated,
 $Dc = |\phi_{0,-h}^1 - 2\phi_{-h,-h}^1 + \phi_{-2h,-h}^1| + |\phi_{-h,0}^1 - 2\phi_{-h,-h}^1 + \phi_{-h,-2h}^1|$.
 else, $Dc = \infty$.
 - If $(\{0, -h\}, \{-h, -h\}, \{h, -h\})$ has been updated,
 $Db = |\phi_{h,-h}^1 - 2\phi_{0,-h}^1 + \phi_{-h,-h}^1| + |\phi_{0,0}^1 - 2\phi_{0,-h}^1 + \phi_{0,-2h}^1|$,
 else, $Db = \infty$.
 - If $Dc \leq Db$, choose $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$,
 else, choose $(\{0, -h\}, \{-h, -h\}, \{h, -h\})$.
- if $\frac{b_1}{a_1} = 1$.
 - If $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$ has been updated,
 $Dc = |\phi_{0,-h}^1 - 2\phi_{-h,-h}^1 + \phi_{-2h,-h}^1| + |\phi_{-h,0}^1 - 2\phi_{-h,-h}^1 + \phi_{-h,-2h}^1|$.
 else, $Dc = \infty$.
 - If $(\{-h, 0\}, \{-h, -h\}, \{-h, h\})$ has been updated,
 $Da = |\phi_{0,0}^1 - 2\phi_{-h,0}^1 + \phi_{-2h,0}^1| + |\phi_{0,h}^1 - 2\phi_{0,0}^1 + \phi_{0,-h}^1|$,
 else, $Da = \infty$.
 - If $(\{0, -h\}, \{-h, -h\}, \{h, -h\})$ has been updated,
 $Db = |\phi_{h,0}^1 - 2\phi_{0,0}^1 + \phi_{-h,0}^1| + |\phi_{0,0}^1 - 2\phi_{0,-h}^1 + \phi_{0,-2h}^1|$,
 else, $Db = \infty$.
 - If $Dc \leq \min\{Da, Db\}$, choose $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$,
 else if $Da \leq Db$, choose $(\{-h, 0\}, \{-h, -h\}, \{-h, h\})$,
 else, choose $(\{0, -h\}, \{-h, -h\}, \{h, -h\})$.

2. when the ray is in region 2,

- If $(\{0, -h\}, \{-h, -h\}, \{h, -h\})$ has been updated,
 $D_1 = |\phi_{h,-h}^1 - 2\phi_{0,-h}^1 + \phi_{-h,-h}^1|$, else, $D_1 = \infty$.
- If $(\{0, -h\}, \{-h, -h\}, \{-2h, -h\})$ has been updated,
 $D_2 = |\phi_{0,-h}^1 - 2\phi_{-h,-h}^1 + \phi_{-2h,-h}^1|$, else, $D_2 = \infty$.
- If $D_1 \leq D_2$, choose $(\{0, -h\}, \{-h, -h\}, \{h, -h\})$,
 else, choose $(\{0, -h\}, \{-h, -h\}, \{-2h, -h\})$.

3. when the ray is in region 3,

- If $(\{-h, 0\}, \{-h, -h\}, \{-h, h\})$ has been updated,
 $D_1 = |\phi_{-h,h}^1 - 2\phi_{-h,0}^1 + \phi_{-h,-h}^1|$, else, $D_1 = \infty$.

- If $(\{-h, 0\}, \{-h, -h\}, \{-h, -2h\})$ has been updated,
 $D_2 = |\phi_{-h,0}^1 - 2\phi_{-h,-h}^1 + \phi_{-h,-2h}^1|$, else, $D_2 = \infty$.
- if $D_1 \leq D_2$, choose $(\{-h, 0\}, \{-h, -h\}, \{-h, h\})$,
else, choose $(\{-h, 0\}, \{-h, -h\}, \{-h, -2h\})$.

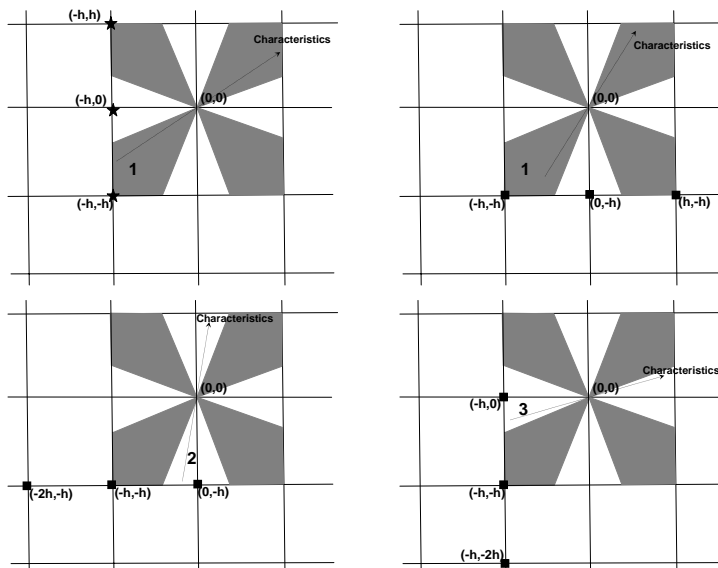


Fig. 4.2. Alternative stencil on a possible shock

The explanation of our strategy is the following: when the ray is in region 1 and $\frac{b_1}{a_1} < 1$, we choose between two stencils: $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$ and $(\{-h, 0\}, \{-h, -h\}, \{-h, h\})$ to update the value at $\{0, 0\}$ since characteristic in (a_1, b_1) direction intersects both stencils. We approximate the second derivative $|\phi_{xx}| + |\phi_{yy}|$ at points $\{-h, -h\}, \{-h, 0\}$ and choose the smaller one. The strategy for other cases is similar.

For the chosen stencil, we need to calculate (a, b, C) from system of equations as (4.5), (4.11) and (4.17), and the numerical Hamiltonian (4.6), (4.12) and (4.18). We skip the computation and discussion here since they are the same.

In the following section 5, we will present our scheme, with a detailed procedure incorporating the stencil-choosing criterion.

5. A one pass deferred correction scheme

Now we present our complete algorithm as a one pass deferred correction to the numerical solution, which we refer to as first order solution, to any upwind monotone scheme. The value of the first order solution is used for the ordering of all grid points. The upwind numerical gradient of the first order solution is used to determine upwind quadrant/stencils as well as the second order correction. The scheme goes through all grid points in the ascendent order of the first order solution and second order correction is propagated from sources or boundary conditions in an efficient one pass fashion. Note that only accurate values are needed at the boundary.

During the one pass correction process, the key issue is how to choose an upwind stencil that satisfies the **Stencil-choosing criterion**: (1) the stencil is composed of neighboring grid points that have been updated to second order accurate, (2) the stencil provides a smooth approximations of second derivatives and a stable interpolation. Once the stencil is chosen, second order correction can be implemented. When we make the second order correction sequentially according to the ordering of the first order solution, we are following the propagation of the wavefront (the level set of ϕ). Since each possible compact upwind stencil extends a angle at most 90° , there is no problem in finding a stencil that satisfies the above criterion during the one pass correction if the wavefront can be resolved by the underlying grid. Since characteristics go into shocks, our Stencil-choosing criterion will be able to pick a smooth updated stencil near shocks. In the case not all the values of the upwind stencil we choose have been updated to second order accurate or the nonnegativity of the discriminant is not satisfied, special treatment is needed. (see detailed description of the scheme below). This may happen, for instance, near a point source (a rarefaction wave) due to high-curvature wavefront.

A One Pass Second Order Correction Scheme:

Step 1. Use monotone upwind scheme (e.g., FSM, 9-point stencil) to compute a first order solution ϕ^1 and the corresponding upwind gradient (a_1, b_1) . Order all grid points in the ascendent order of ϕ^1 .

Step 2. Initialize boundary conditions. Put $mark = 1$ on points where the boundary values are prescribed.

Step 3. Go over each point according to ordering and make the second order correction.

Here is the detailed procedure for choosing the appropriate upwind stencil and making the second order correction. Denote the current grid as $\{0, 0\}$. We give the algorithm for the case $a_1 \geq 0, b_1 \geq 0$. Other cases (including $a_1 = 0$ or $b_1 = 0$) are treated similarly. The upwind quadrant and different regions are illustrated in Figure 4.1 and 4.2. The detail of Stencil-choosing criterion is described in Section 4.3.

★ if (a_1, b_1) is in region 1.

- Choose the stencil according to the "Stencil-choosing criterion".
- If the stencil has been updated.
 - If the numerical Hamiltonian has a real root, update $\phi(0, 0)$ and update $mark(0, 0) = 1$.
 - else,
 - * If the stencil is $(\{-h, 0\}, \{0, -h\}, \{-h, -h\})$,
 - if $\frac{b_1}{a_1} < 1$, use (4.5a) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
 - if $\frac{b_1}{a_1} > 1$, use (4.5b) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
 - if $\frac{b_1}{a_1} = 1$, use (4.5c) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
 - * If the stencil is $(\{0, -h\}, \{-h, -h\}, \{h, -h\})$, use (4.11a) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.

- * If the stencil is $(\{-h, 0\}, \{-h, -h\}, \{-h, h\})$, use (4.17a) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
- If the stencil hasn't been updated.
 - if $\frac{b_1}{a_1} < 1$,
 - * if $mark(-h, 0) = 1$, use (4.5a) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
 - * else, use (4.5c) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
 - if $\frac{b_1}{a_1} > 1$,
 - * if $mark(0, -h) = 1$, use (4.5b) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
 - * else, use (4.5c) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
 - if $\frac{b_1}{a_1} = 1$, use (4.5c) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
- ★ if (a_1, b_1) is in region 2.
 - Choose the stencil according to the "Stencil-choosing criterion".
 - If the stencil has been updated.
 - If the numerical Hamiltonian has a real root, update $\phi(0, 0)$ and update $mark(0, 0) = 1$.
 - else, use (4.11a) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
 - If the stencil has been updated, use (4.11a) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
- ★ if (a_1, b_1) is in region 3.
 - Choose the stencil according to the "Stencil-choosing criterion".
 - If the stencil has been updated.
 - If the numerical Hamiltonian has a real root, update $\phi(0, 0)$ and update $mark(0, 0) = 1$.
 - else, use (4.17a) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.
 - else, use (4.17a) to update $\phi(0, 0)$ with $(a, b) = (a_1, b_1)$, and update $mark(0, 0) = 1$.

6. Numerical examples

In this section, we test our scheme with both homogeneous and heterogeneous cases. We use the 9-point stencil fast sweeping method to compute the first order solution in all our tests. Error¹⁾ is recorded. The computational domain is $[0, 1] \times [0, 1]$.

¹⁾ $E.L_\infty$ and $E.L_1$ represent maximum norm error and L_1 norm error.

Example 1: Homogeneous case: we show some test results for the distance function to one source point and multiple source points in Table 6.1 and 6.2. A disk with fixed radius is wrapped up around each source, then the boundary condition is assigned to the disk(s). Figure 6.1 shows the plots of a multiple-source case on a 160×160 mesh.

Source point=(0.5,0.5)					
Mesh	80x80	160x160	320x320	640x640	1280x1280
E_{L_∞}	995.594825	222.604406	52.641615	12.992351	3.202858
Order	-	2.161	2.080	2.019	2.020
Source point=(0.4999,0.4997)					
Mesh	80x80	160x160	320x320	640x640	1280x1280
E_{L_∞}	999.521155	223.493026	53.710565	13.018126	3.209031
Order	-	2.161	2.057	2.045	2.020

Table 6.1: Distance function to one source. Error magnitude 10^{-7} .

Multiple sources					
Mesh	80x80	160x160	320x320	640x640	1280x1280
E_{L_∞}	5909.268831	5606.583983	3463.307601	1592.033549	469.877280
Order	-	0.076	0.695	1.121	1.761
E_{L_1}	1447.185635	312.339249	64.565416	16.185021	3.966551
Order	-	2.212	2.274	1.996	2.029

Table 6.2: Distance function to multiple sources: (0.2001,0.2001), (0.39985,0.19965), (0.5001,0.20039), (0.59985,0.24965), (0.2003,0.5002), (0.74985,0.49965), (0.25021,0.70019), (0.29985,0.74965), (0.49985,0.749965) and (0.749985,0.749965). Error magnitude 10^{-7} .

Example 2: Heterogeneous case 1: we build heterogeneous test cases by choosing a phase function $\phi(x, y) = \min_{(x_i, y_i)} \{1 - e^{-[(x-x_i)^2 + (y-y_i)^2]}\}$ with single source and multiple sources (the exact solution) and build the index of refraction by taking the norm of the gradient of this function. Table 6.3 and 6.4 show the results. Figure 6.2 shows the plots of the multiple-source case on a 160×160 mesh.

Source point= (0.5,0.5)					
Mesh	80x80	160x160	320x320	640x640	1280x1280
E_{L_∞}	1620.656952	257.349879	104.916781	26.361747	6.613370
Order	-	2.655	1.294	1.992	1.995
Source point= (0.4999,0.4997)					
Mesh	80x80	160x160	320x320	640x640	1280x1280
E_{L_∞}	1519.841986	369.526585	84.622353	17.952740	4.872976
Order	-	2.040	2.127	2.237	1.881

Table 6.3: $\phi(x, y) = 1 - e^{-[(x-x_0)^2 + (y-y_0)^2]}$: one source. Error magnitude 10^{-7} .

Example 3: Heterogeneous case 2: this example shows a test for $\phi(x, y) = \sin[\pi(1 + x)]\sin(\pi y)$. Table 6.5 shows the results. Figure 6.3 shows the plots on a 160×160 mesh.

The above examples show that the maximum error for the solution in a smooth region that is outside an $O(h)$ neighborhood of the shock/kink (away from shock/kink) is second order

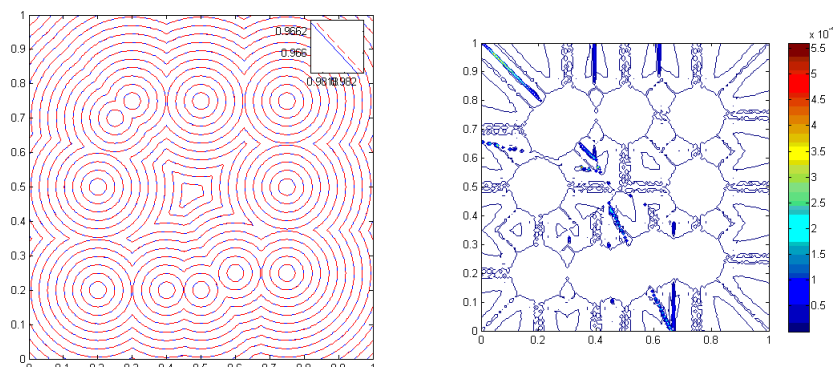


Fig. 6.1. Distance function to multiple sources. LEFT: solution plots (blue: exact solution, red-dashed: second order solution). RIGHT: error plot.

Multiple sources					
Mesh	80x80	160x160	320x320	640x640	1280x1280
E_{L_∞}	4145.408510	2946.903103	1158.282506	781.124027	526.569565
Order	-	0.492	1.347	0.568	0.569
E_{L_1}	21186.626686	4748.593298	997.761039	213.280224	49.575092
Order	-	2.158	2.251	2.226	2.105

Table 6.4: $\phi(x, y) = \min_{(x_i, y_i)} \{1 - e^{-[(x-x_i)^2 + (y-y_i)^2]}\}$. Multiple sources: (0.2001,0.2001), (0.39985,0.19965), (0.5001,0.20039), (0.59985,0.24965), (0.2003,0.5002), (0.74985,0.49965), (0.25021,0.70019), (0.29985,0.74965), (0.49985,0.74965) and (0.749985,0.749965). Error magnitude 10^{-7} .

accurate. When inside this $O(h)$ neighborhood of the shock/kink, it reduces to at most first order accurate. However the L_1 norm error is still second order accurate on the whole domain.

7. Conclusion

A compact upwind second order scheme is presented. The scheme is used as a one pass second order correction to first order monotone upwind schemes. The method uses the observed gradient superconvergence phenomena of first order monotone upwind schemes and the Lagrangian structure of the equation itself.

The method is simpler than the usual ENO methods but less general. It is doubtful that the same ideas can be used to construct schemes of order higher than 2.

The scheme also depends on the "Lagrangian structure" of the Eikonal equation and can be

Mesh	80x80	160x160	320x320	640x640	1280x1280
E_{L_∞}	10272.185787	2568.194937	642.510826	160.633922	40.158919
Order	-	2.000	2.000	2.000	2.000

Table 6.5: Phase function: $\phi(x, y) = \sin[\pi(1+x)]\sin(\pi y)$, source=(0.5, 0.5). Error magnitude 10^{-7} .

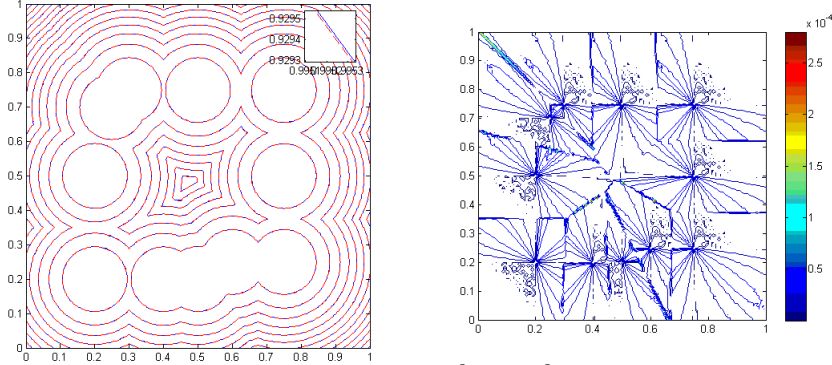


Fig. 6.2. $\phi(x, y) = \min_{(x_i, y_i)} \{1 - e^{-[(x-x_i)^2 + (y-y_i)^2]}\}$. LEFT: solution plots (blue: exact solution, red-dashed: second order solution). RIGHT: error plot.

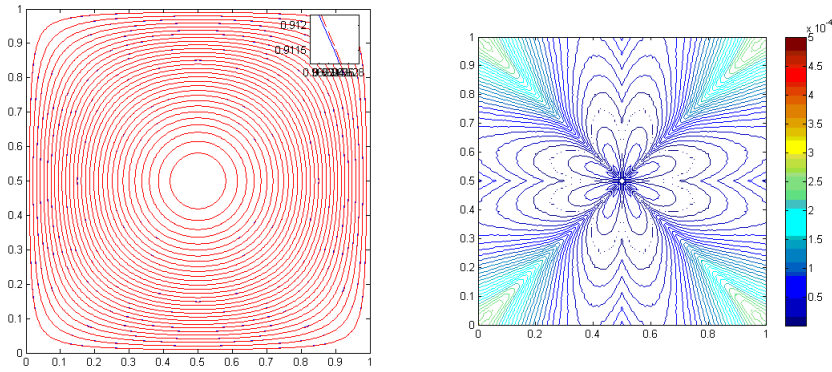


Fig. 6.3. LEFT: solution plots (blue: exact solution, red-dashed: second order solution). RIGHT: error plot.

extended to more general convex (or concave) Hamilton-Jacobi equations: $H(X, \nabla\phi(X)) = 0$. The Lagrangian structure then becomes:

$$D^2\phi \cdot H_p(X, \nabla\phi) = -H_X(X, \nabla\phi)$$

and the decomposition of the Hessian in the Taylor expansion is still done along the ray (of direction $H_p(X, \nabla\phi)$) and its orthogonal direction.

The same kind of idea can be applied to a transport equation coupled to the Eikonal equation. Amplitude equation can be simplified down to $\nabla Y_0 \cdot \nabla\phi = 0$ and taking the gradient we can use in a similar fashion the following "Lagrangian structure":

$$D^2 Y_0 \cdot \nabla\phi = -D^2\phi \nabla Y_0$$

to simplify the second order terms in the Taylor Expansion of Y_0 .

Future projects include extension to 3D and to unstructured mesh. We believe a similar approach can be used to construct the stencils, therefore the second order scheme.

References

- [1] G. Barles and P. E. Souganidis, Convergence of approximation schemes for fully nonlinear second order equations, *Asymptotic Analysis*, **4** (1991), 271-283, North-Holland.
- [2] M. Boué and P. Dupuis, Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control, *SIAM Journal on Numerical Analysis*, **36** (1999), 667-695.
- [3] M. G. Crandall and P. L. Lions, Viscosity solutions of Hamilton-Jacobi equations, *Tans. Amer. Math. Soc.*, **277** (1983), 1-42.
- [4] M. G. Crandall and P. L. Lions, Two approximations of solutions of Hamilton-Jacobi equations, *Math. of Comp.*, **43**:167 (1984), 1-19.
- [5] P. Danielsson, Euclidean distance mapping, *Computer Graphics and Image Processing*, **14** (1980), 227-248.
- [6] E. W. Dijkstra, A note on two problems in connection with graphs, *Numerische Mathematik*, **1** (1959), 269-271.
- [7] M. Falcone and R. Ferretti, Discrete time high-order schemes for viscosity solutions of Hamilton-Jacobi-Bellman equations, *Numer. Math.*, **67** (1994), 315-344.
- [8] M. Falcone and R. Ferretti, Semi-Lagrangian schemes for Hamilton-Jacobi equations, discrete representation formulae and Gobunov methods, *Journal of Computational Physics*, **175** (2002), 559-575.
- [9] J. Helmsen, E. Puckett, P. Colella and M. Dorr, Two new methods for simulating photolithography development in 3D, *Proc. SPIE*, **2726** (1996), 253-261.
- [10] G.-S. Jiang and D. Peng, Weighted ENO schemes for Hamilton-Jacobi equations, *SIAM Journal on Scientific Computing*, **21** (2000), 2126-2143.
- [11] C. Y. Kao, S. Osher and J. Qian, Lax-Friedrichs sweeping schemes for static Hamilton-Jacobi equations, *Journal of Computational Physics*, **196** (2004), 367-391.
- [12] C. Y. Kao, S. Osher and Y. H. Tsai, Fast sweeping method for static Hamilton-Jacobi equations, *SIAM Journal on Numerical Analysis*, **42** (2005), 2612-2632.
- [13] F. Li, C.-W. Shu, Y.-T. Zhang and H. Zhao, Second order discontinuous fast sweeping method for Eikonal equations, *J. Comp. Phys.*, **227**:17 (2008), 8191-8208.
- [14] A. M. Oberman, Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton-Jacobi equations and free boundary problems, *SIAM J. NUMER. ANAL.*, **44**:2 (2006), 879-895.
- [15] S. Osher, A level set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations, *SIAM J. Math. Anal.*, **24** (1983), 1145-1152.
- [16] S. Osher and C. W. Shu, High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations, *SIAM J. Numer. Anal.*, **28**:4 (1991), 907-922.
- [17] J.L. Qian and W. Symes. An adaptive finite difference method for travel-time and amplitudes. *Geophysics*, **67** (2002) 167-176.
- [18] J. Qian, Y. Zhang and H. Zhao, A fast sweeping method for static convex Hamilton-Jacobi equations, *Journal of Scientific Computing*, **31**:1/2 (2007), 237-271.
- [19] J. Qian, Y. Zhang and H. Zhao, Fast sweeping method for Eikonal equations on triangulated meshes, *SIAM Journal on Numerical Analysis*, **45** (2007), 83-107.
- [20] E. Rouy and A. Tourin, A viscosity solutions approach to shape-from-shading, *SIAM Journal on Numerical Analysis*, **29** (1992), 867-884.
- [21] J. A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proc. Nat. Acad. Sci.*, **93** (1996), 1591-1595.

- [22] J. A. Sethian and A. Vladimirsky, Ordered upwind methods for static Hamilton-Jacobi equations, *Proc. Natl. Acad. Sci.*, **98** (2001), 11069-11074.
- [23] J. A. Sethian and A. Vladimirsky, Ordered upwind methods for static Hamilton-Jacobi equations: theory and algorithms, *SIAM Journal on Numerical Analysis*, **41** (2003), 325-363.
- [24] C.-W. Shu, High order numerical methods for time dependent Hamilton-Jacobi equations, *IMS Lecture Notes Series, Mathematics and Computation in Imaging Science and Information Progressing*, World Scientific Publilshing, Singapore, **11** (2007), 47-91.
- [25] P. E. Souganidis, Approximation schemes for viscosity solutions of Hamilton-Jacobi equations, *J. Differential Equatioins*, **59** (1985), 1-43.
- [26] W. Symes, R. Versteeg, A. Sei, and Q. H. Tran. Kirchhoff simulation migration and inversion using finite-difference travel-times and amplitudes. TRIP tech. Report, Rice U., 1994.
- [27] Y. R. Tsai, Rapid and accurate computation of the distance function using grids, *J. Comp. Phys.*, **178**:1 (2002), 175-195.
- [28] Y.-H. R. Tsai, L.-T. Cheng, S. Osher and H.-K. Zhao, Fast sweeping algorithms for a class of Hamilton-Jacobi equations, *SIAM Journal on Numerical Analysis*, **41** (2003), 673-694.
- [29] J. N. Tsitsiklis, Efficient algorithms for globally optimal trajectories, *IEEE Transactions on Automatic Control*, **40** (1995), 1528-1538.
- [30] Y. Zhang, J. Qian and H. Zhao, High order fast sweeping methods for static Hamilton-Jacobi equations, *Journal of Scientific Computing*, **29** (2006), 25-56.
- [31] H. Zhao, A fast sweeping method for Eikonal equations, *Math. Comp.*, **74** (2005), 603-627
- [32] H. Zhao, Parallel implementations of the fast sweeping method, *Journal of Computational Mathematics*, **25**:4 (2007), 421-429.