The Fast Sweeping Method

Hongkai Zhao

Department of Mathematics
University of California, Irvine

Highlights of the fast sweeping mehtod

- Simple (A Gauss-Seidel type of iterative method).

- Optimal complexity.

- Local solver with compact and fixed stencils for convex Hamiltonian on arbitrary mesh.

- A truly nonlinear method.

- Idea can be applied to non-upwind or high order schemes.

A little background

- Danielson's algorithm (1980).
- Fast sweeping method for stochastic control (Boué and Dupuis, 1999).
- PDE based fast sweeping method for Eikonal equation (Zhao et al, 2000)
- Alternating sweeping to solve linearized steady compressible Euler equation (Jameson and Caughey, 2001).
- A lot of recent work on FSM by Fomel, Kao, Li, Luo, Osher, Qian, Renzi, Tsai, Zhang, Zhao, ...

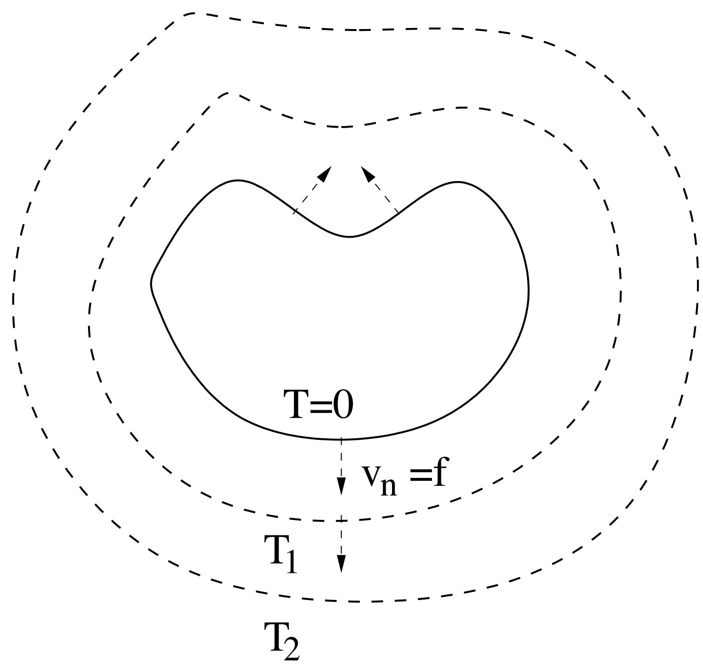
Introduction to fast sweeping method (FSM)

- FSM for Eikonal equation on rectangular grid.
- FSM for general convex Hamilton-Jacobi equations (Qian, Zhang, Z.).
- Understanding FSM in the framework of iterative methods.

The Eikonal equation

Isotropic case: $\|\nabla u(\boldsymbol{x})\| = c(\boldsymbol{x}), \quad u(\boldsymbol{x}) = 0, \ \boldsymbol{x} \in \mathcal{S}$

The viscosity solution $u(x)$ is the first arrival time for a front starting at $\mathcal{S}$ with a propagation speed $v(\boldsymbol{x}) = \dfrac{1}{c(\boldsymbol{x})}$.



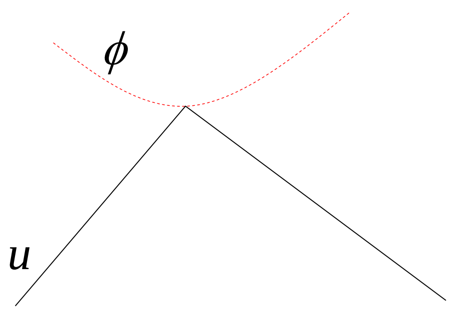
Anisotropic case: $[\nabla u(\boldsymbol{x}) M(x) \nabla u(\boldsymbol{x})]^{\frac{1}{2}} = 1, \quad M(\boldsymbol{x})$ is a SPD matrix

Different interpretation of viscosity solution (1)

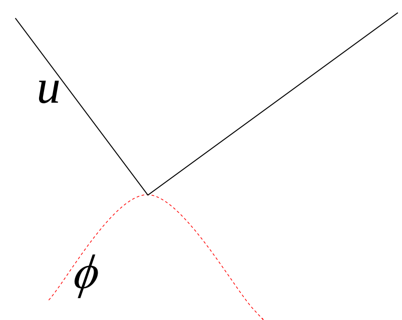
A function $u \in C^{0,1}(\overline{\Omega})$ is a viscosity *subsolution* (*supersolution*) if for all $\phi \in C_0^\infty(\Omega)$ with $u - \phi$ attaining a local maximum (minimum) at some $x_0 \in \Omega$ then

$$H(x_0, \nabla\phi(x_0)) \leq 0 \quad (\geq 0) \tag{1}$$

A viscosity solution is both a viscosity sub- and supersolution.



(a) subsolution



(b) supersolution

Remark: A general definition using test function. Not constructive and difficult to use in practice.

Different interpretation of viscosity solution (2)

Let u^ϵ be the solution to

$$H(x, \nabla u^\epsilon(x)) = \pm\epsilon\Delta u^\epsilon(x),$$

$u^\epsilon \to u$ as $\epsilon \to 0$. u is the viscosity solution.

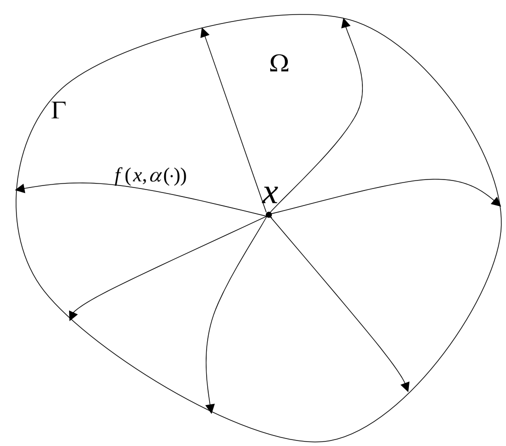
Singular perturbation and limit process using artificial viscosity.
Equation type is changed.
$\Rightarrow$ Lax-Friedrichs scheme using numerical viscosity.

Different interpretation of viscosity solution (3)

Control Interpretation: for convex Hamilton-Jacobi equation the viscosity solution can be constructed as

$$u(x) = \inf_{y \in \partial\Omega}(g(y) + L(x,y))$$

$$L(x,y) = \inf \left\{ \int_0^1 l(\xi(t), \xi'(t))dt : \xi \in C^{0,1}([0,1], \overline{\Omega}), \xi(0) = x, \xi(1) = y \right\}$$



Construction based on Lagrangian formulation.
For $\|\nabla u(x)\| = c(x) \Rightarrow l(\xi(t), \xi'(t)) = c(\xi(t))\|\xi'(t)\|$.

Numerical methods

Two crucial ingredients:

◇ Appropriate discretization (local solver).

Design numerical Hamiltonian that converges to the right viscosity solution, deals with non-smoothness, and has high order accuracy if possible.

◇ Solve the system of discretized equations.

• Time marching methods:

explicit but need many iterations due to finite speed of propagation and the CFL condition.

• Direct methods for the nonlinear boundary value problem:

efficient but need to solve a large system of non-linear equations.

The fast sweeping algorithm on rectgangular grids

Solve $|\nabla u(\boldsymbol{x})| = c(\boldsymbol{x})$, with $u(\boldsymbol{x}) = 0$, $\boldsymbol{x} \in \mathcal{S}$.

- Local solver (**nonlinear equation**): upwind and causal

$$[(u_{i,j}-u_{xmin})^+]^2+[(u_{i,j}-u_{ymin})^+]^2=h^2c_{i,j}^2, \qquad i,j = 1,2,\ldots \quad (2)$$

$u_{xmin}=\min(u_{i-1,j},\ u_{i+1,j}), u_{ymin}=\min(u_{i,j-1}, u_{i,j+1})$.

- Initial guess: $u_{i,j} = u(\boldsymbol{x}_{i,j}) = 0$, $\boldsymbol{x}_{i,j} \in \mathcal{S}$, $\mathbf{u_{i,j}} = \infty$, $\boldsymbol{x}_{\mathbf{i,j}} \notin \mathcal{S}$.

- G-S iterations with four alternating ordering:

$$\begin{array}{ll} (1)\ i=1:I, j=1:J & (2)\ i=I:1, j=1:J \\ (3)\ i=I:1, j=J:1 & (4)\ i=1:I, j=J:1 \end{array}$$

Let $\overline{u}$ be the solution to (1) using current values of $u_{i\pm1,j}, u_{i,j\pm1}$,

$$u_{i,j}^{new} = \min(u_{i,j}^{old}, \overline{u}), \quad \text{control interpretation is used.}$$

Gauss-Seidel method for a linear elliptic problem

Solve $\quad \Delta u(\boldsymbol{x}) = c(\boldsymbol{x}), \qquad u(\boldsymbol{x}) = 0,\ x \in S$

- Discretization at grid $\boldsymbol{x}_{i,j}$ (**linear equation**), $i, j = 1, 2, \ldots$

$$u_{i+1,j} - 2u_{i,j} + u_{i-1,j} + u_{i,j+1} - 2u_{i,j} + u_{i,j-1} = h^2 c_{i,j}$$

- Give an initial guess of u with $u(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in S$.

- Gauss-Seidel iteration: for $i = 1 : I, j = 1 : J$

$$u_{i,j}^{n+1} = \frac{u_{i+1,j}^{n} + u_{i-1,j}^{n+1} + u_{i,j+1}^{n} + u_{i,j-1}^{n+1} - h^2 c_{i,j}}{4}$$

1. **The iteration converges.**
2. **The number of iteration depends on the grid size.**
3. **Different ordering does not make much difference.**

Solve the numerical Hamiltonian in n dimensions

Order a_i's in the increasing order, assume $a_1 \leq a_2 \leq \ldots \leq a_n$, there is a unique solution $\overline{x}$ to

$$[(x-a_1)^+]^2 + [(x-a_2)^+]^2 + \cdots + [(x-a_n)^+]^2 = c_{i,j}^2 h^2$$

which satisfies,

$$(x-a_1)^2 + (x-a_2)^2 + \cdots + (x-a_p)^2 = c_{i,j}^2 h^2$$

$a_p < \overline{x} \leq a_{p+1}$ for some $p, 1 \leq p \leq n$.

Procedure:

1. Let $\tilde{x} = a_1 + hc_{i,j}$, if $\tilde{x} \leq a_2$ then $\overline{x} = \tilde{x}$; otherwise
2. Solve $(x-a_1)^2 + (x-a_2)^2 = c_{i,j}^2 h^2$. Let $\tilde{x}$ be the solution, if $\tilde{x} \leq a_3$ then $\overline{x} = \tilde{x}$, otherwise
3. Solve $(x-a_1)^2 + (x-a_2)^2 + (x-a_3)^2 = c_{i,j}^2 h^2$,

. . . ,

untill we reach p.

The monotonicity and Lipschitz continuity of the scheme

Lemma0: Let $\overline{x}$ be the solution to,

$$[(x-a_1)^+]^2 + [(x-a_2)^+]^2 + \cdots + [(x-a_n)^+]^2 = r^2$$

we have

$$1 > \frac{\partial \overline{x}}{\partial a_1} \geq \frac{\partial \overline{x}}{\partial a_2} \geq \ldots \geq \frac{\partial \overline{x}}{\partial a_n} \geq 0,$$

$$\frac{\partial \overline{x}}{\partial a_1} + \frac{\partial \overline{x}}{\partial a_2} + \ldots + \frac{\partial \overline{x}}{\partial a_n} = 1,$$

and

$$\frac{\partial \overline{x}}{\partial r} \leq 1$$

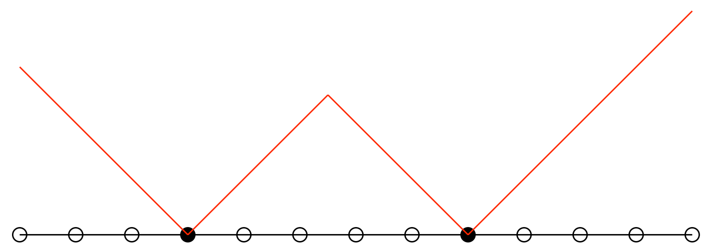
$\Rightarrow$

(1) the numerical solution converges to the viscosity solution.

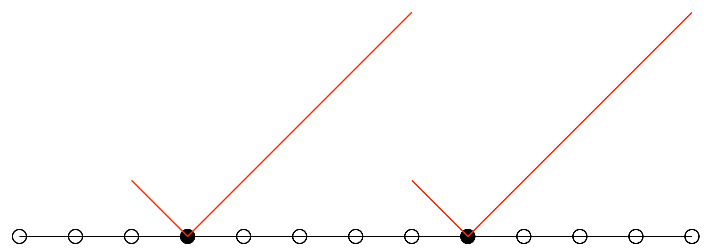
(2) maximum change principle + contraction of error for each update at a grid point during G-S iterations.

Motivation for the fast sweeping method ($c(x) = 1$)

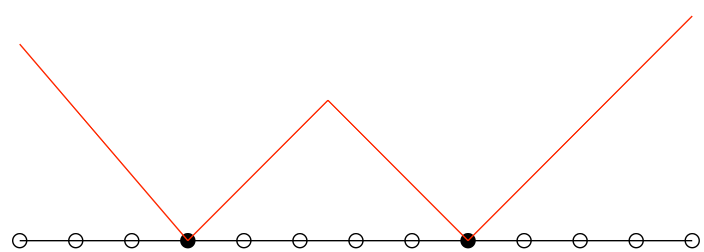
In 1D, there are two directions of characteristics, left or right. Two sweeps will find the exact solution.



(a) the exact distance function to a point



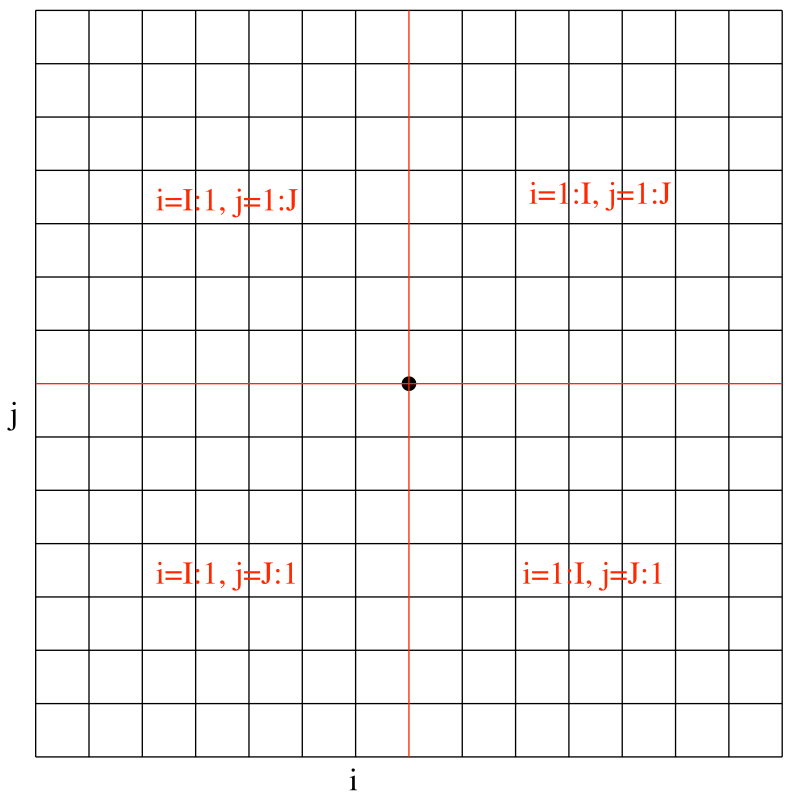
(b) the computed solution after first left to right sweeping



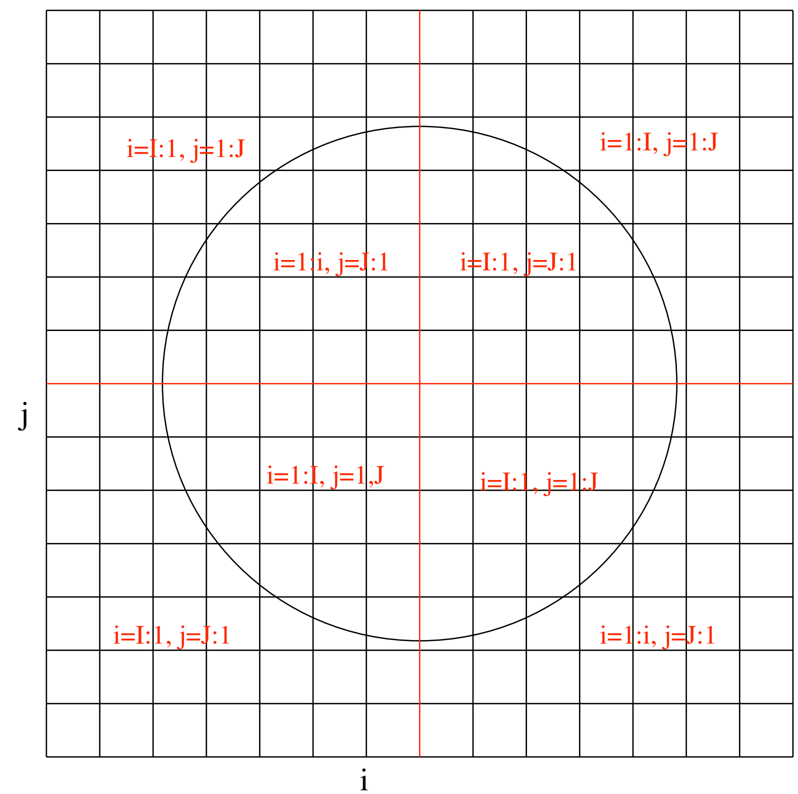
(c) the computed solution after second right to left sweeping

The fast sweeping algorithm in 2D

Facts: The characteristics have all directions. However all directions can be classified into four groups, up-right, up-left, down-left and down-right. Each sweep ordering covers one group of the characteristics simultaneously.



(a) one data point



(b) a circle

Convergence and error estimate on rectangular grid

Theorem1: The solution of the fast sweeping algorithm converges monotonically to the solution of the discretized system.

$\mathcal{S}$: the data set; $u^h(\boldsymbol{x}, \mathcal{S})$: the numerical solution; $d(\boldsymbol{x}, \mathcal{S})$: the distance function.

Theorem2: For a single data point $\mathcal{S} = \{\boldsymbol{x}_0\}$, $u^h(\boldsymbol{x}, \boldsymbol{x}_0)$ converges after 2^n sweeps in n dimensions and

$$d(\boldsymbol{x}, \boldsymbol{x}_0) \le u^h(\boldsymbol{x}, \boldsymbol{x}_0) \le d(\boldsymbol{x}, \boldsymbol{x}_0) + O(h|\log h|) \qquad \text{(sharp!)}$$

Theorem3: For arbitrary $\mathcal{S}$ in n dimensions, after 2^n iterations.

$$\overline{u}(\boldsymbol{x}_{i,j}, \mathcal{S}) \le u^h(\boldsymbol{x}_{i,j}, \mathcal{S}) \le \underline{u}^h(\boldsymbol{x}_{i,j}) \le d(\boldsymbol{x}_{i,j}, \mathcal{S}) + O(|h \log h|),$$

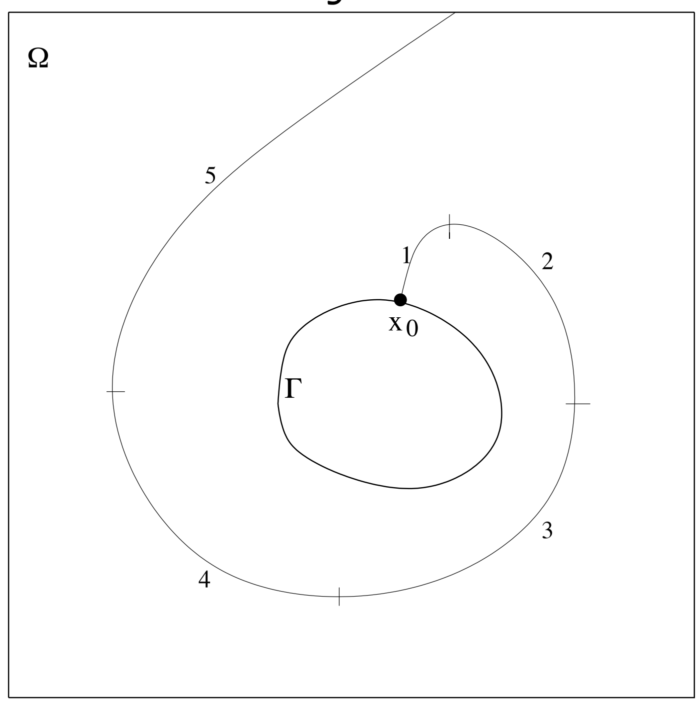
where $\overline{u}(\boldsymbol{x}_{i,j}, \mathcal{S})$ is the solution to the discretized system.

General Eikonal equation $|\nabla u(\boldsymbol{x})| = c(\boldsymbol{x})$

Denote $H(\mathbf{p}, \boldsymbol{x}) = |\mathbf{p}| - c(\boldsymbol{x})$, where $\mathbf{p} = \nabla u$.
The characteristic equations are:

$$\begin{cases} \dot{\boldsymbol{x}} = \nabla_{\mathbf{p}} H = \frac{\nabla u}{c(\boldsymbol{x})} \\ \dot{\mathbf{p}} = -\nabla_{\boldsymbol{x}} H = \nabla c(\boldsymbol{x}) \\ \dot{u} = \nabla u \cdot \dot{\boldsymbol{x}} = c(\boldsymbol{x}) \end{cases}$$

Each characteristic curve can be segmented into a finite number of pieces and can be covered by the G-S sweeps successively.



Total direction variation along a characteristic

$$\begin{cases} |\dot{\boldsymbol{x}}| = \left|\dfrac{\nabla u}{c(\boldsymbol{x})}\right| = 1, \\ \ddot{\boldsymbol{x}} = \dfrac{\nabla \dot{u}}{c(\boldsymbol{x})} - \dfrac{\nabla u}{c} \cdot \dfrac{\nabla c}{c} \cdot \dot{\boldsymbol{x}} = (I - P_{\boldsymbol{n}})\dfrac{\nabla c(\boldsymbol{x})}{c(\boldsymbol{x})}. \end{cases}$$

$P_{\boldsymbol{n}}$ is the projection on $\boldsymbol{n} = \frac{\nabla u}{|\nabla u|}$. $\ddot{\boldsymbol{x}}$ is the curvature.

Let $K = \max_{\boldsymbol{x} \in \Omega} \frac{|\nabla c(\boldsymbol{x})|}{c(\boldsymbol{x})}$, $D =$ radius of the domain, and $c_M(c_m)$ is the maximum (minimum) of $c(\boldsymbol{x})$. The total direction variation is bounded by

$$\int_\Gamma |\ddot{\boldsymbol{x}}| ds \le \int_\Gamma \frac{|\nabla c(\boldsymbol{x})|}{c(\boldsymbol{x})} ds \le K \int_\Gamma ds \le K \text{length}(\Gamma) \le DK \frac{c_M}{c_m}$$

Maximum length of a characteristic

Let the characteristics Γ joins $\boldsymbol{x}_0 \in \mathcal{S}$ and a $\boldsymbol{x} \in \Omega$.

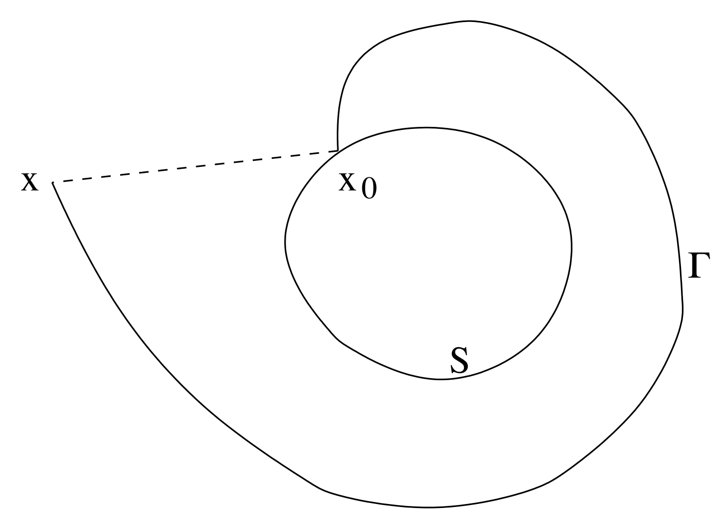
$$c_m \int_\Gamma ds \le \underline{\int_\Gamma c(s)ds = u(\boldsymbol{x})} \le \int_{\boldsymbol{x}_0}^{\boldsymbol{x}} c(s)ds \le c_M \|\boldsymbol{x} - \boldsymbol{x}_0\|$$

Hence

$$\text{length}(\Gamma) = \int_\Gamma ds \le \frac{Dc_M}{c_m}$$

where D is the radius of the domain and $c_M(c_m)$ is the maximum (minimum) of $c(\boldsymbol{x})$.

The maximum number of turns is bounded by $\frac{DK}{2\pi}\frac{c_M}{c_m}$.



Discrete version of interpretation

The monotone upwind scheme on a rectangular grid admits a unique solution. All grid points can be divided into a finite number of simply connected regions. In each region the value at a grid point depends on two of its neighbors in four ways:
(1) left and down neighbors; (2) left and above neighbors;
(3) right and down neighbors; (4) right and above neighbors.

Using GS iteration each connected region can be covered by one of the orderings simultaneously when the ordering is in the upwind direction of the dependence pattern. Moreover, a correct causality enforcement guarantee a correct value will not be changed by later iterations, e.g., no interference among different orderings.

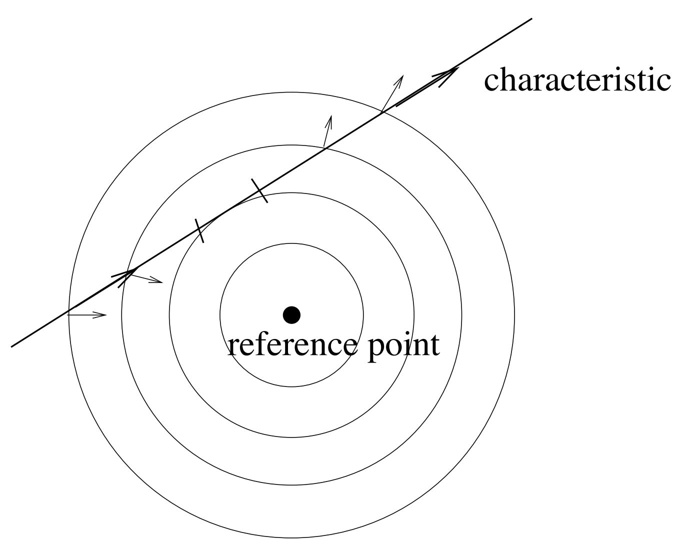
An appropriate upwind scheme + GS with systematic alternating orderings + causality enforcement are crucial.

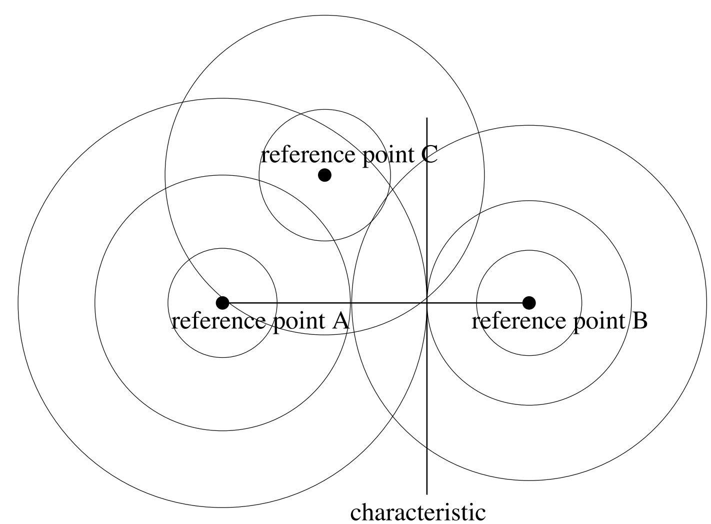
Ordering strategy on unstructured mesh

The Key point: Systematic ordering such that each characteristics can be covered in a finite number of sweep orderings.

Solution 1: Order all vertices accroding to the distance to a few reference points and sweep back and forth.

Solution 2: Using a background rectangular grid. (Bagnerini et al.)

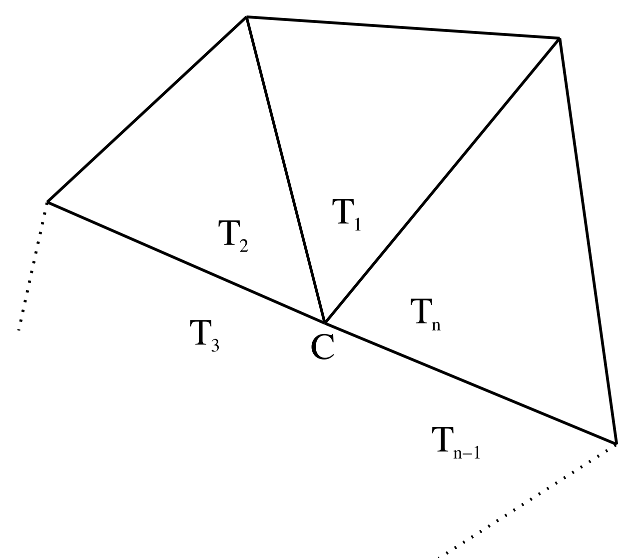




Local solver

At a vertex C, compute $u(C)$ from its neighbors
$\Rightarrow$ A nonlinear equation for $u(C)$ and its neighbors.

First order approximation: solve a HJ equation with local Hamiltonian independent of $\boldsymbol{x}$ on the polygon with C as the central vertex with piecewise linear interpolation of the neighbors at the boundary.



Convex HJ equation

- When the Hamiltonian is convex, optimal control interpretation can be used to design the local solver.

- The optimal path is a straight line from boundary to C.

- One can solve for each triangle and take the minimum one.

$$u(C) = \min\{u_1(C), u_2(C), \ldots\}$$

- The scheme is upwind and is the counter part to the Godunov scheme in hyperbolic conservation law.

- To develop an upwind local solver is as difficult as solving a spatial homogenous HJ equation $H(\mathbf{p}) = 0$.

A PDE based local solver for general convex Hamiltonian

Key points:

- Separate the consistency condition from causality check using the local relation from PDE in each triangle.
- Pick the right viscosity solution from all possible candidates using the control interpretation.

Local solver

Step 1: Find consistent u_C given u_A and u_B.

$$\nabla u(C) \approx \mathbf{P}^{-1} \begin{pmatrix} \frac{u_C - u_A}{b} \\ \frac{u_C - u_B}{a} \end{pmatrix}, \tag{3}$$

using linear approximation, where

$$\mathbf{P} = \begin{pmatrix} \frac{x_C - x_A}{b}, & \frac{y_C - y_A}{b} \\ \frac{x_C - x_B}{a}, & \frac{y_C - y_B}{a} \end{pmatrix} \tag{4}$$

Plug $\nabla u(C)$ into the PDE and solve for u_C

$$H(C, \nabla u(C)) = 0, \qquad \text{or} \qquad \widehat{H}(C, u_C, u_A, u_B) = 0.$$

Local solver

- No solution for $u_C \Rightarrow u_A, u_B$ do not support a consistent u_C in this triangle $\Rightarrow$ choose one of the two sides as the optimal path.

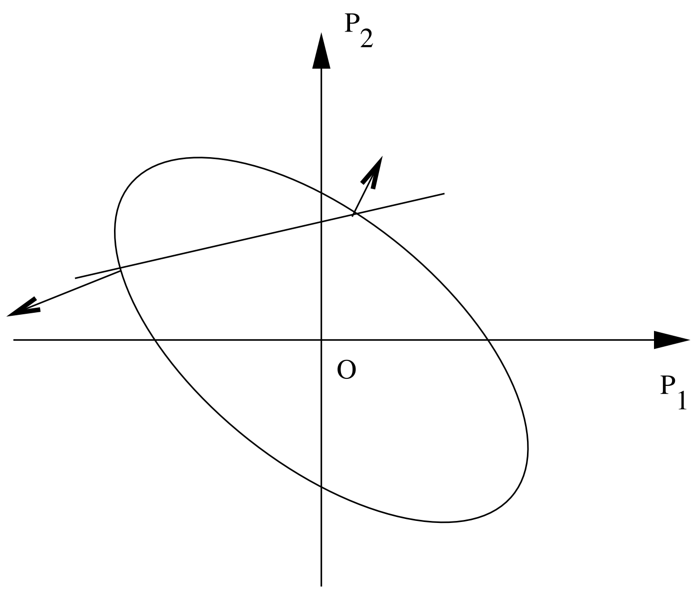
- There are one or more solutions for $u_C \Rightarrow$ causality check.

Step 2: **Causality check:** the characteristic starting from C against the direction $\nabla_{\mathbf{p}} H(C, \nabla u(C))$ intersects the side AB.

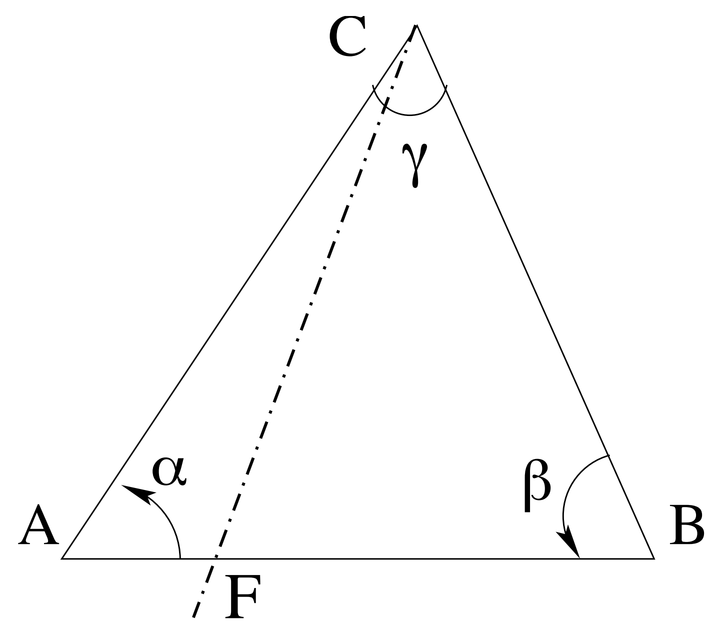
$u(C)$ takes the minimum among all consistent and causality satisfying candidates from each triangle and sides.

Geometric interpretation

Step 1



Step 2



Remark: If the Hamiltonian is convex $\Rightarrow$ there are at most two solutions for u_C

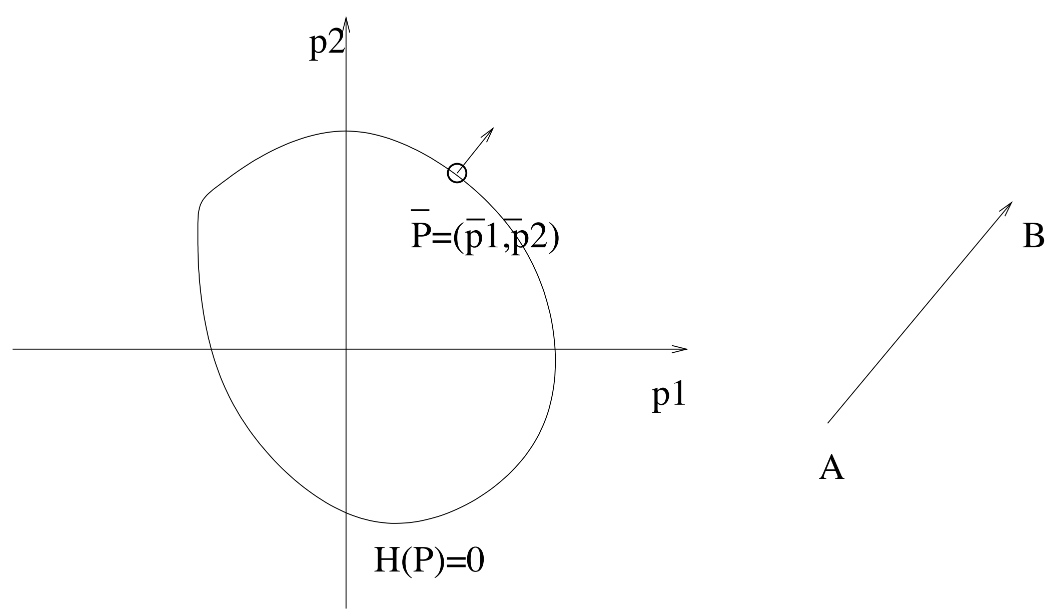
Compute the cost along a fixed path

Find $\overline{\mathbf{p}}$ such that $\dot{\mathbf{x}} = \nabla_{\mathbf{p}} H(\overline{\mathbf{p}}) // \vec{AB}$,

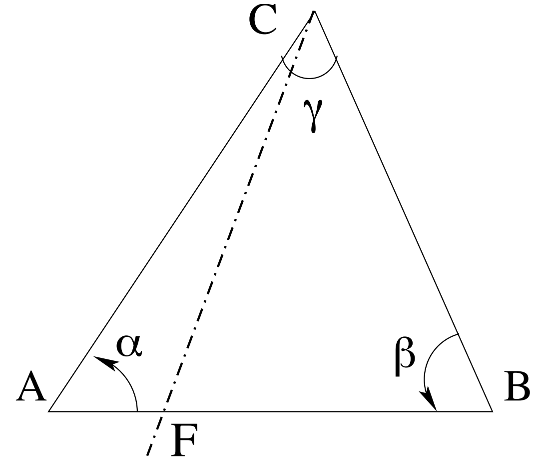
$$v_g = \|\nabla_{\mathbf{p}} H(\overline{\mathbf{p}})\|$$

The cost from A to B is

$$u_B - u_A = \frac{\overline{AB}}{v_g}$$



Local solver in one triangle



$$u_C = \min_{s\in[0,1]} \left\{ s u_B + (1-s) u_A + \frac{d(s)}{v_g(C;s)} \right\}.$$

$$F = sA + (1-s)B, \qquad d(s) = \overline{CF},$$

$v_g(C;s)$: the group velocity in the direction of $\vec{FC}$.

We can prove the equivalence of the PDE based local solver to the control based local solver.

Examples

1. For isotropic Eikonal equation, $\|\nabla u(\mathbf{x})\| = c(\mathbf{x})$

$$\dot{\mathbf{x}} = \nabla_{\mathbf{p}} H = \frac{\nabla u}{\|\nabla u\|}$$

   - $v_g(C; s) = v_g(C)$.
   - $u_C >$ max$\{u_A, u_B\}$.

2. For anisotropic Eikonal equation, $[\nabla u(\mathbf{x}) M(\mathbf{x}) \nabla u(\mathbf{x})]^{1/2} = 1$

$$\dot{\mathbf{x}} = \nabla_{\mathbf{p}} H = \frac{M \nabla u}{[\nabla u M \nabla u]^{1/2}}$$

   - $v_g(C; s)$ depends on s,
   - $u_C >$ min$\{u_A, u_B\}$.

Consistency and monotonicity of the scheme

Lemma The numerical Hamiltonian $\widehat{H}$ is consistent:

$$\widehat{H}\left(C, \frac{u_C - u_A}{b}, \frac{u_C - u_B}{a}\right) \;=\; H(C, \mathbf{p}) \qquad (5)$$

(if $\nabla u_h = \mathbf{p} \in \mathcal{R}^2$). The numerical Hamiltonian $\widehat{H}$ is monotone if the causality condition holds.

$$0 \le \frac{\partial u_C}{\partial u_A}, \; \frac{\partial u_C}{\partial u_B} \le 1, \qquad \frac{\partial u_C}{\partial u_A} + \frac{\partial u_C}{\partial u_B} = 1$$

Remark: For isotropic case, if $\gamma < \frac{\pi}{2}$ and

$$u_C = s u_B + (1-s) u_A + \frac{d(s)}{v_g(C)}, \qquad 0 < s < 1$$

we have

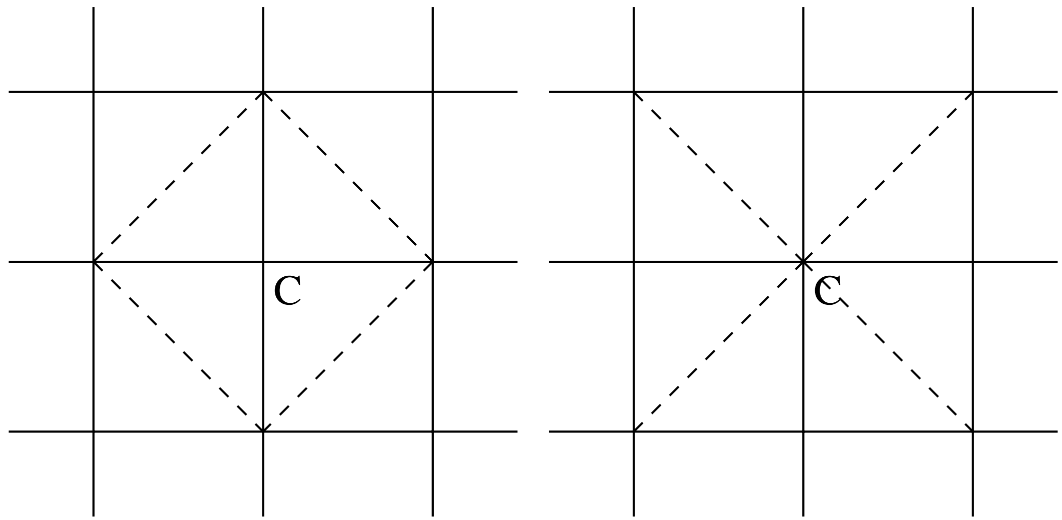
$$u_C > \max\{u_A, u_B\} + \delta(\gamma) h \;\; \text{and} \;\; \delta(\gamma) \overset{\gamma \to \frac{\pi}{2}}{\longrightarrow} 0$$

Convergence

Theorem 4 There exists a unique solution for the discretized nonlinear system and the fast sweeping iteration converges.

Theorem 5 The numerical solution converges to the viscosity solution as $h \to 0$

Two special cases for rectangular grids.



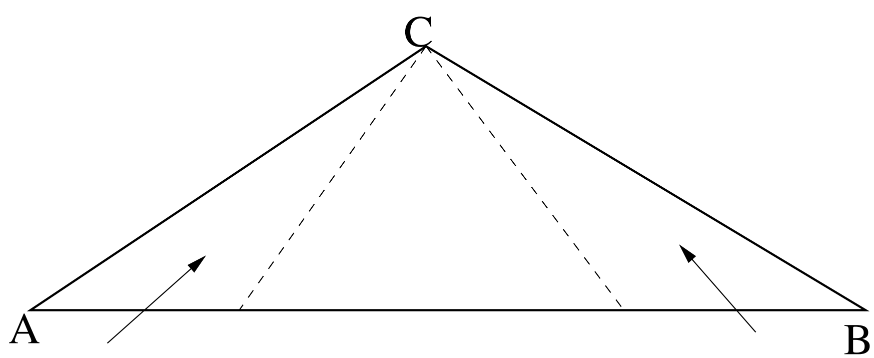
(a) Five point stencils. (b) Nine point stencils.

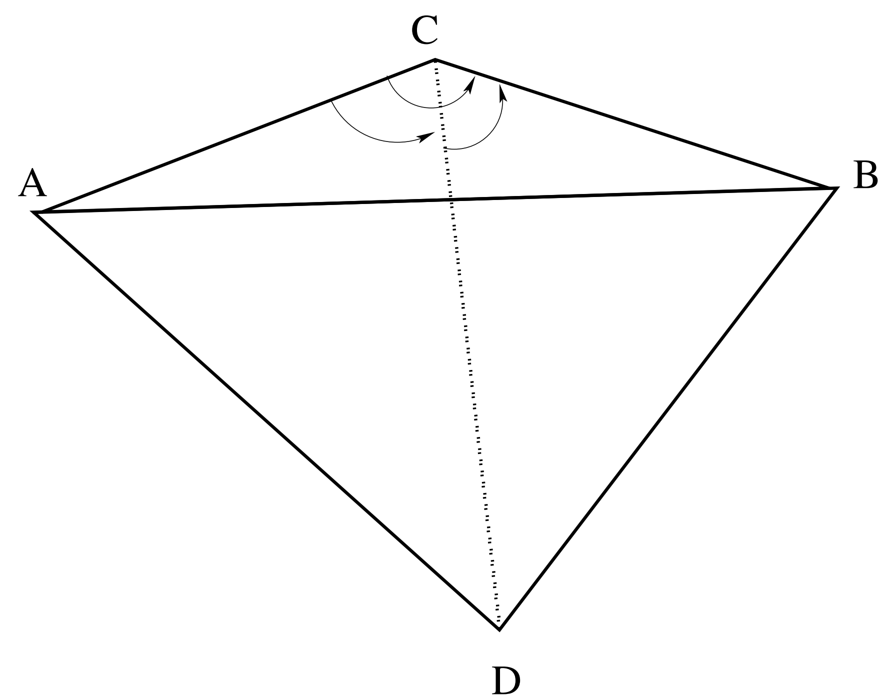
Case (b) will yield more accurate solution due to better angle resolution.

Obtuse angle

Obtuse angle may cause problems by using information in the wrong direction. Acute angle will not.

Treatment of obtuse angle: virtual splitting and connection can be used to take care of this.





Remarks on local solvers

- Lax-Friedirchs scheme uses centered difference + artificial viscosity.

$$H(\frac{u_x^+ + u_x^-}{2}, \frac{u_y^+ + u_y^-}{2}) + \alpha(u_x^+ - u_x^-) + \beta(u_y^+ - u_y^-)$$

- Implicit unconditionally stable scheme can be easily derived for time dependent HJ equation:

$$u_t + H(x, \nabla u) = 0$$

Remarks on the complexity of FSM

Let M be the total number of nodes, the complexity of FSM is $O(M)$;

Remark :

- The constant in the complexity formula depends on inhomogeneity but does not depend on the anisotropy of the Hamiltonian.

- Locking and dynamic queue strategy (Bak, McLaughlin and Renzi, 2009) can be used to reduce unnecessary updates, i.e., reduce the constant in O.

Anisotropic eikonal equations

$$[\nabla u(\mathbf{x}) M(\mathbf{x}) \nabla u(\mathbf{x})]^{\frac{1}{2}} = 1, \qquad \mathbf{x} \in R^d, \tag{6}$$

where $M(\mathbf{x})$ is a $d \times d$ symmetric positive definite matrix. In two dimensions

$$H = \sqrt{a(\mathbf{x})\, p_1^2 - 2c(\mathbf{x})\, p_1\, p_2 + b(\mathbf{x})\, p_2^2} = 1, \tag{7}$$

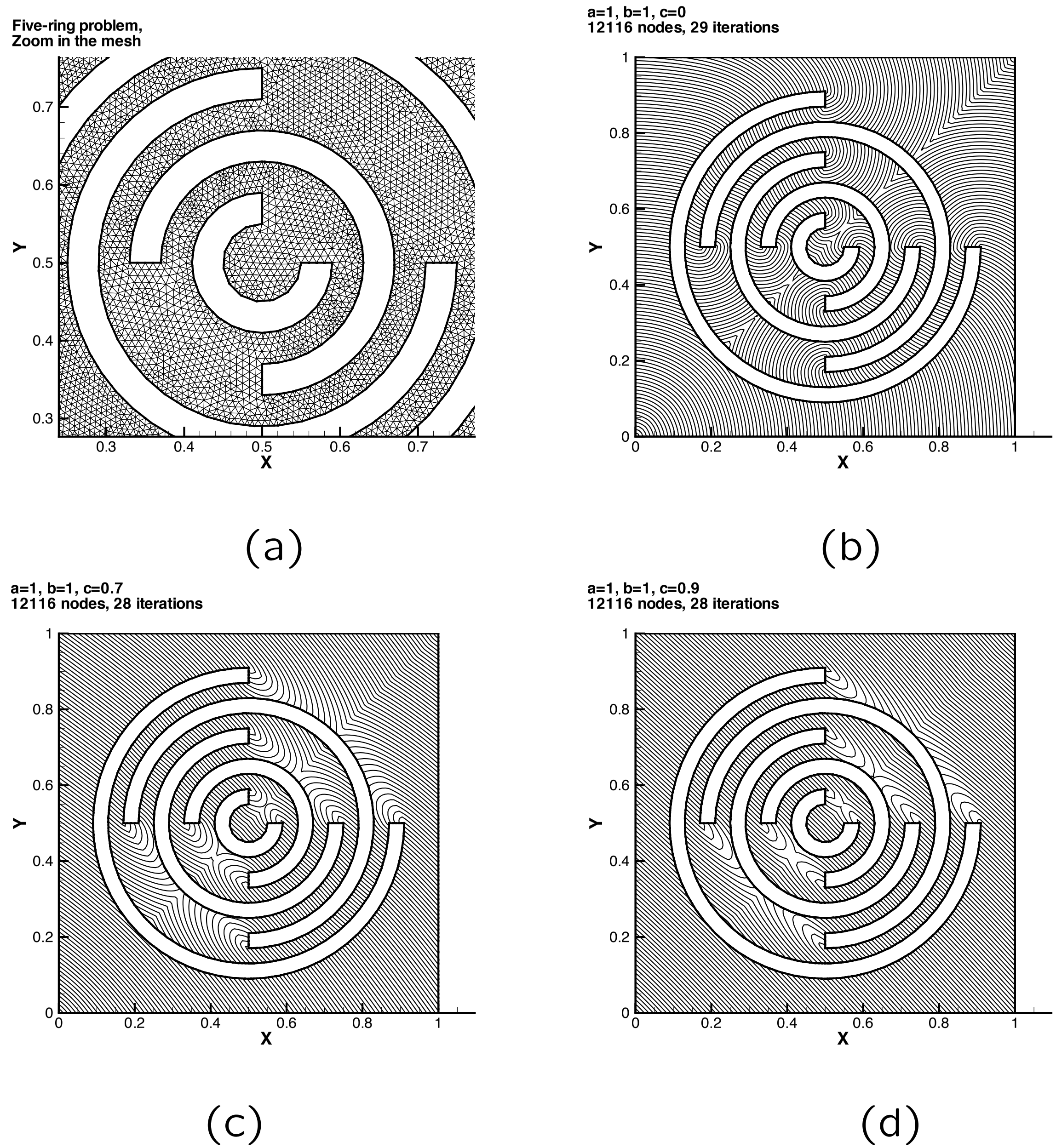
where $a > 0$, $b > 0$ and $c^2 - ab < 0$, e.g.,

$$\mathbf{M} = \begin{pmatrix} a, & -c \\ -c, & b \end{pmatrix}$$

The anisotropy of **M** (the metric) is characterized by

$$\eta = \sqrt{\frac{\lambda_{\max}(\mathbf{M})}{\lambda_{\min}(\mathbf{M})}},$$

where $\lambda_{\max}(\mathbf{M})$ and $\lambda_{\min}(\mathbf{M})$ are the larger and smaller eigenvalues of **M**, respectively.



(a): the mesh; (b): a=1, b=1, c=0; $\eta = \sqrt{\frac{1+c}{1-c}} = 0$; 29 sweepings; (c): a=1, b=1, c=0.7; 28 sweepings; (d): a=1, b=1, c=0.9; 28 sweepings.

		Wrapping $[-0.2, 0.2] \times [-0.2, 0.2]$			No wrapping		
Nodes	Elements	L^1 error	order	iter	L^1 error	order	iter
1473	2816	8.78E-3	–	4	8.87E-3	–	4
5716	11264	4.04E-3	1.12	4	5.38E-3	0.72	4
22785	45056	2.10E-3	0.94	4	3.22E-3	0.74	5
90625	180224	1.04E-3	1.02	4	1.88E-3	0.77	5

The order of convergence; $a = 150.25, b = 50.75, c = 86.16953, \eta = \sqrt{200}$.

		Wrapping $[-0.2, 0.2] \times [-0.2, 0.2]$			No wrapping		
Nodes	Elements	L^1 error	order	iter	L^1 error	order	iter
1473	2816	6.23E-3	–	4	5.72E-3	–	4
5716	11264	2.89E-3	1.11	4	3.33E-3	0.78	4
22785	45056	1.53E-3	0.92	4	1.93E-3	0.79	5
90625	180224	7.66E-4	1.00	4	1.10E-3	0.80	5

The order of convergence; $a = 1500.25, b = 500.75, c = 865.5924, \eta = \sqrt{2000}$.

	five point stencils			nine point stencils		
Mesh	L^1 error	order	iter	L^1 error	order	iter
40×40	1.17E-1	–	4	1.57E-2	–	4
80×80	6.35E-2	0.88	4	8.18E-3	0.94	4
160×160	3.39E-2	0.90	4	4.18E-3	0.97	4
320×320	1.78E-2	0.93	4	2.12E-3	0.98	4

Comparison between the five point stencils and the nine point stencils. $a = 1, b = 1, c = 0.9$.

Contraction property for linear problem

upwind scheme + GS iteration + right ordering

Linear example: $au_x + bu_y = 0, \qquad a > b > 0$

The first order monotone upwind scheme is

$$u^h_{i,j} = \frac{a}{a+b} u^h_{i-1,j} + \frac{b}{a+b} u^h_{i,j-1}$$

Define

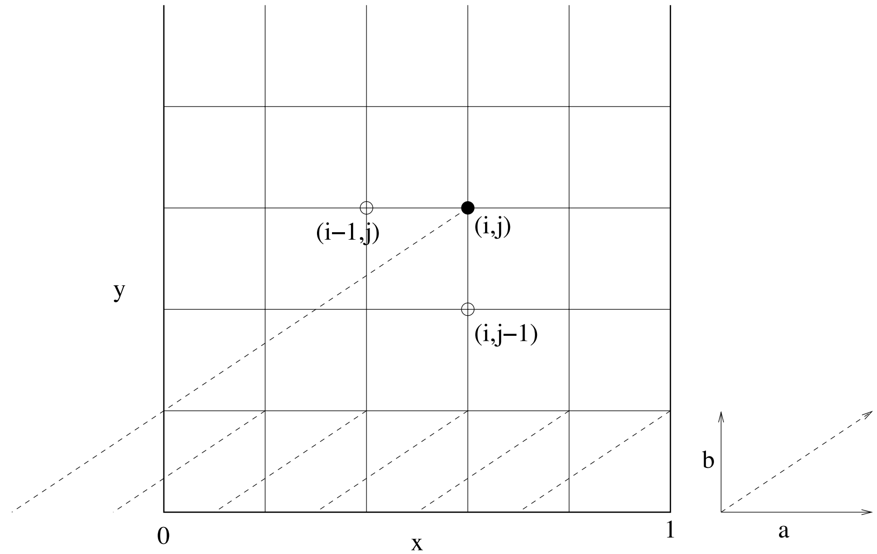
$$e^n_{i,j} = u^n_{i,j} - u^h_{i,j},$$

with **right ordering** $(i = 1 : I, j = 1 : J)$ for GS iteration, the update at every grid point gives

$$e^n_{i,j} = \frac{a}{a+b} e^n_{i-1,j} + \frac{b}{a+b} e^n_{i,j-1}$$

Remark: Similar relation is true for monotone upwind scheme on triangular mesh.

Two convergence scenarios (1)



Case 1: Information propagates directly from boundary. Convergence in finite number of iterations independent of mesh size. For the linear example with constant coefficient, one sweep with right ordering is needed if boundary condition is given at left and bottom.

Two convergence scenarios (2)

Case 2: There is circular dependent, e.g. with given boundary condition at the bottom and periodic boundary condition in x. With right ordering, at each grid the GS update gives an error contraction. For example, for the first grid line about the bottom

$$e_{i,1}^{n} = \frac{a}{a+b} e_{i-1,1}^{n}$$

Hence, in each sweep with the right ordering, the error is contracted by a factor of $\alpha^{\frac{1}{h}}$ after one iteration ($\alpha(a,b)<1$). Very few iterations n is needed for $\left(\alpha^{\frac{1}{h}}\right)^{n} \sim h^{p}$ for any p. The smaller h the faster.

Convergence for nonlinear problem

Our local solver for convex HJ equation is upwind and has the following monotonicity + contraction property on general mesh

$$0 \le \frac{\partial u_C}{\partial u_A}, \ \frac{\partial u_C}{\partial u_B} \le 1, \qquad \frac{\partial u_C}{\partial u_A} + \frac{\partial u_C}{\partial u_B} = 1$$

error is reduced at each update when the ordering is correct.

GS+fast sweeping $\Rightarrow$ fast convergence.

Differences between linear and nonlinear problem

Linear Problem:

- Ordering and contraction rate is known á priori (depending on the coefficients).
- The largest error may influence all points and the solution may get worse at some points during iterations.

Nonlinear problem:

- Ordering and contraction rate is unknown á priori (depending on the solution).
- Nonlinear stability: The largest error may not influence any other points and the solution at all grids gets better during iterations due to causality enforcement.
- Information propagates through edges very fast at the begining.

Linear example

$$a u_x + b u_y = 0, \quad \Omega = [0,1] \times [0,1].$$

B.C. $u(x,0) = \sin(2\pi x), u(0,y) = \sin(-2\pi \frac{a}{b} y)$

	a=1, b=2	a=2, b=1	a=4, b=1	a=8, b=1
h=1/20	2	2	2	2
h=1/40	2	2	2	2
h=1/80	2	2	2	2
h=1/160	2	2	2	2

B.C. $u(x,0) = \sin(2\pi x)$, periodic in x.

	a=1, b=2	a=2, b=1	a=4, b=1	a=8, b=1
h=1/20	6	13	23	46
h=1/40	5	9	16	32
h=1/80	4	7	12	23
h=1/160	4	6	10	18

$$[\nabla u M_\theta \nabla u]^{\frac{1}{2}} = 1, \quad u(x,0) = (\alpha+1) + \alpha \sin(2\pi x), \quad \text{periodic in } x$$

where θ is the rotation angle and $M_0 = \begin{pmatrix} 1/16 & 0 \\ 0 & 1 \end{pmatrix}$.

	$\theta=0$	$\theta=\pi/6$	$\theta=\pi/3$	$\theta=\pi/2$	$\theta=2\pi/3$	$\theta=5\pi/6$
h=1/20	2	12	20	2	18	10
h=1/40	2	12	17	2	15	10
h=1/80	2	12	16	2	14	10
h=1/160	2	8	16	2	14	6
h=1/320	2	8	12	2	10	6

$\alpha = 0$, tolerance=10^{-7}

	$\theta=0$	$\theta=\pi/6$	$\theta=\pi/3$	$\theta=\pi/2$	$\theta=2\pi/3$	$\theta=5\pi/6$
h=1/20	6	12	20	6	19	10
h=1/40	6	12	16	6	15	10
h=1/80	6	9	16	6	14	10
h=1/160	6	8	12	6	10	6
h=1/320	6	8	12	6	10	6

$\alpha = 0.15$, tolerance=10^{-7}

tolerance=machine 0

	$\theta = 0$	$\theta = \pi/6$	$\theta = \pi/3$	$\theta = \pi/2$	$\theta = 2\pi/3$	$\theta = 5\pi/6$
h=1/20	2	20	32	2	30	18
h=1/40	2	16	23	2	22	14
h=1/80	2	12	20	2	18	14
h=1/160	2	8	16	2	14	10

$\alpha = 0$

	$\theta = 0$	$\theta = \pi/6$	$\theta = \pi/3$	$\theta = \pi/2$	$\theta = 2\pi/3$	$\theta = 5\pi/6$
h=1/20	6	16	32	6	26	18
h=1/40	6	16	21	6	26	14
h=1/80	6	12	20	6	18	10
h=1/160	6	12	20	6	14	10

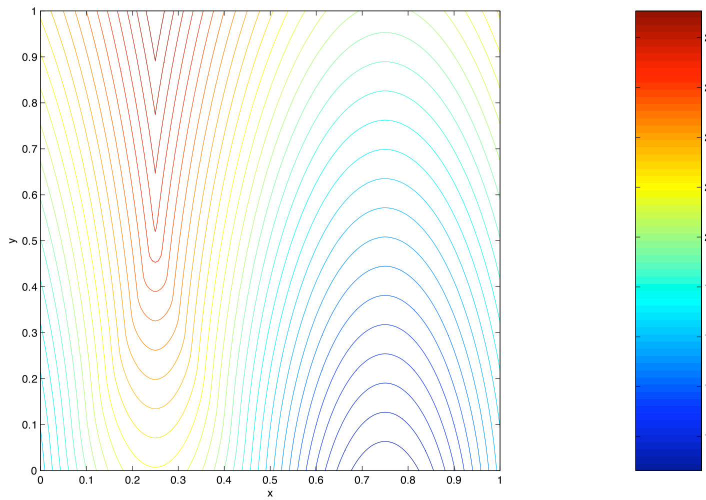
$\alpha = 0.15$

5 point stencil

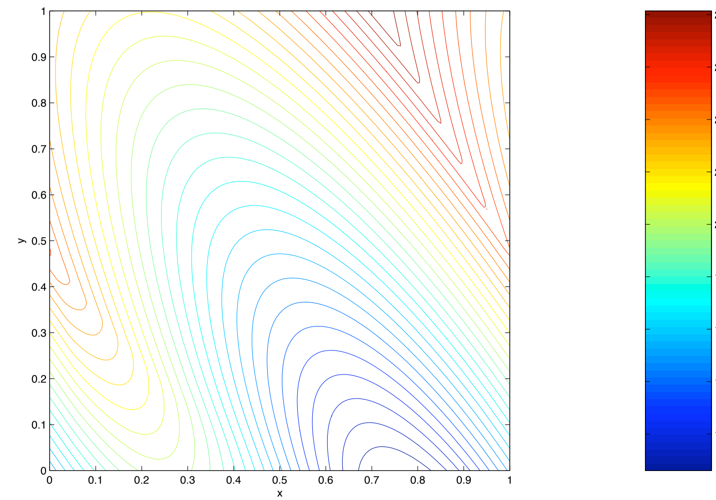
	$\alpha = 0.6$ $\theta = 0$	$\alpha = 0.5$ $\theta = \pi/6$	$\alpha = 0.3$ $\theta = \pi/3$	$\alpha = 0.15$ $\theta = \pi/2$	$\alpha = 0.3$ $\theta = 2\pi/3$	$\alpha = 0.5$ $\theta = 5\pi/6$
h=1/20	6	16	28	6	30	18
h=1/40	6	16	23	6	23	13
h=1/80	6	12	20	6	18	10
h=1/160	6	12	16	6	14	10

9 point stencil

	$\alpha = 0.6$ $\theta = 0$	$\alpha = 0.5$ $\theta = \pi/6$	$\alpha = 0.3$ $\theta = \pi/3$	$\alpha = 0.15$ $\theta = \pi/2$	$\alpha = 0.3$ $\theta = 2\pi/3$	$\alpha = 0.5$ $\theta = 5\pi/6$
h=1/20	6	8	8	6	7	10
h=1/40	6	12	8	6	6	6
h=1/80	6	12	8	6	6	6
h=1/160	6	12	8	6	7	6



(a) $\theta = 0$



(b) $\theta = \pi/3$

Can an iterative method converge in finite iterations?

Elliptic problem: no!
Every point is coupled with all other points $\Rightarrow$ the discretized system can not be a triangular system.
Convergence mechanism: the iteration is a contraction map.
Wish: a contraction rate that is bounded away from 1.

Hyperbolic problem: yes!
Information propagates along characteristics $\Rightarrow$ if an appropriate upwind scheme and ordering is used, the discretized system can be (or almost) put into a triangular system.
Convergence mechanism: capture propagation of information.
Wish: ordering or the nodes follows the characteristics.

- Linear problem: ordering can be determined á *priori*.
- Nonlinear problem: ordering depends on the solution. GS iteration with sweeping and upwind difference is a must to cover all characteristics **blindly and efficiently.**

Interpretation in the framework of iterative methods

- In theory, convergence in a finite number of iterations $\equiv$ the nonlinear system can be put in a triangular (banded) system according to the causality. FSM achieves this by enforcing causality in the scheme and using Gauss-Seidel iterations with alternating sweeping.

- Time marching method is Jacobi type iteration which uses information from previous iteration (step). Information propagates with finite speed between iterations (steps). Gauss-Seidel iteration uses newest information. If causality is enforced correctly information can be propagated much more efficiently.

- Alternating sweeping does not affect elliptic problems but may help hyperbolic problems.

- Potential difficulties for more general hyperbolic problems: (1) causality, (2) upwind schemes and local solver.

Error in the gradient

$$(u_x^h)_{j,i} = \frac{u_{j,i} - u_{j,i-1}}{h} - \frac{e^h_{j,i} - e^h_{j,i-1}}{h}$$

The error is first order if the the error $e^h_{j,i}$ is **O(h) and smooth**.
For linear example, $au_x + bu_y = f$ with $a, b > 0$,

$$e^h_{j,i} = \frac{a}{a+b} e^h_{j-1,i} + \frac{b}{a+b} e^h_{j,i-1} + \frac{he^T_{j,i}}{a+b}$$

$e^T_{j,i}$ is truncation error. Denote $\alpha = \frac{a}{a+b}, \beta = \frac{b}{a+b}, e^T = max|e^T_{j,i}|$.

$$\begin{aligned} e_{j,i} - e_{j,i-1} &= \tfrac{b}{a+b}(e_{j-1,i} - e_{j,i-1}) + \tfrac{h}{a+b} T_{j,i} \\ &= \tfrac{b}{a+b}((e_{j-1,i} - e_{j-1,i-1}) - (e_{j,i-1} - e_{j-1,i-1})) + \tfrac{h}{a+b} T_{j,i} \end{aligned}$$

Analysis continued

$$\frac{e_{j,i}-e_{j,i-1}}{h}$$
$$= \sum_{s=1}^{j+i-1} \sum_{k=0}^{s-1} \binom{s-1}{k} \beta^{s-1-k} \alpha^{k+1} \frac{T_{j-k,i-(s-k)} - T_{j-k-1,i-(s-k)+1}}{a+b} + \frac{T_{j,i}}{a+b}$$

$\Rightarrow$

$$\left|\frac{e_{j,i} - e_{j,i-1}}{h}\right| \leq \alpha(i+j-1)\frac{|\triangle T|}{a+b} + \frac{|T_{j,i}|}{a+b}$$

with $|\triangle T| = max_{0\leq k\leq j, 0\leq l\leq i}|T_{k,l-1} - T_{k-1,l}|$.

If $|\triangle T| = O(h^2), |T_{j,i}| = O(h)$, the gradient is first order.

Eikonal equation

$$\begin{cases} \left(\frac{u^h_{j,i}-u^h_{j,i-1}}{h}\right)^2 + \left(\frac{u^h_{j,i}-u^h_{j-1,i}}{h}\right)^2 & = 1 \\ \left(\frac{u_{j,i}-u_{j,i-1}}{h}\right)^2 + \left(\frac{u_{j,i}-u_{j-1,i}}{h}\right)^2 & = T_{j,i}+1 \end{cases}$$

Let

$$a_{j,i} = \frac{(u_{j,i}-u_{j,i-1}) + (u^h_{j,i}-u^h_{j,i-1})}{h} = \frac{2(u_{j,i}-u_{j,i-1}) + (e_{j,i-1}-e_{j,i})}{h}.$$

$$b_{j,i} = \frac{(u_{j,i}-u_{j-1,i}) + (u^h_{j,i}-u^h_{j-1,i})}{h} = \frac{2(u_{j,i}-u_{j-1,i}) + (e_{j-1,i}-e_{j,i})}{h}.$$

$$\alpha_{j,i} = \frac{a_{j,i}}{a_{j,i}+b_{j,i}}, \beta_{j,i} = \frac{b_{j,i}}{a_{j,i}+b_{j,i}}, \gamma_{j,i} = a_{j,i}+b_{j,i}.$$

$$\Rightarrow \begin{cases} e_{j,i}-e_{j,i-1} & = \beta_{j,i}(e_{j-1,i}-e_{j,i-1}) + \frac{T_{j,i}h}{\gamma_{j,i}} \\ e_{j,i}-e_{j-1,i} & = \alpha_{j,i}(e_{j,i-1}-e_{j-1,i}) + \frac{T_{j,i}h}{\gamma_{j,i}} \end{cases}$$

Convergence

With careful and clever analysis, one may establish first order in gradient away from shocks and source singularities.

Remark: Gradient of u is important information in path planing and transport of information.

Recent Developements

- Remove point source singularity based on factorized equation (Fomel, Luo, and Z., JCP 09)

- High order schemes
  - using WENO (Zhang, Qian, and Z., JSC 06)
  - using DG (Li, Shu, Zhang, and Z., JCP10, JCP to appear)
  - deferred correction based on a compact scheme (Benamou, Luo, and Z., JCM 10)

Factored eikonal equation

We want to solve

$$|\nabla T(\mathbf{x})|^2 = S^2(\mathbf{x}).$$

Suppose we know a good solution of

$$|\nabla T_0(\mathbf{x})|^2 = S_0^2(\mathbf{x}).$$

Consider a factored decomposition

$$\begin{aligned} S(\mathbf{x}) &= S_0(\mathbf{x})\,\alpha(\mathbf{x})\,, \\ T(\mathbf{x}) &= T_0(\mathbf{x})\,\tau(\mathbf{x}) \end{aligned}$$

Assume $\alpha(\mathbf{x})$ is known and smooth.

Goal: $\tau(\mathbf{x})$ can be solved efficiently and accurately and so is $T(\mathbf{x})$.

Factored eikonal equation

$\tau(\mathbf{x})$ satisfies the factored eikonal equation:

$$T_0^2(\mathbf{x})\, |\nabla\tau|^2 + 2\, T_0(\mathbf{x})\, \tau(\mathbf{x})\, \nabla T_0 \cdot \nabla\tau + \left[\tau^2(\mathbf{x}) - \alpha^2(\mathbf{x})\right]\, S_0^2(\mathbf{x}) = 0$$

Take $S_0(\mathbf{x}) = 1$, $T_0(\mathbf{x}) = |\mathbf{x} - \mathbf{x_0}|$.

$$|\mathbf{x} - \mathbf{x_0}|^2\, |\nabla\tau|^2 + 2\, \tau(\mathbf{x})\, (\mathbf{x} - \mathbf{x_0}) \cdot \nabla\tau + \tau^2(\mathbf{x}) - S^2(\mathbf{x}) = 0$$

Remarks:

- Remove singularity at a source point $\mathbf{x}_0$ using distance function so $\tau(\mathbf{x})$ does not have singularity at the source point $\mathbf{x}_0$.

- We need to design an efficient numerical algorithm to solve the factored eikonal equation.

Fast Sweeping Algorithm For The Factored Equation

Key point: design an upwind scheme and Guass-Seidel iteration with causality condition based on $T(\mathbf{x})$.

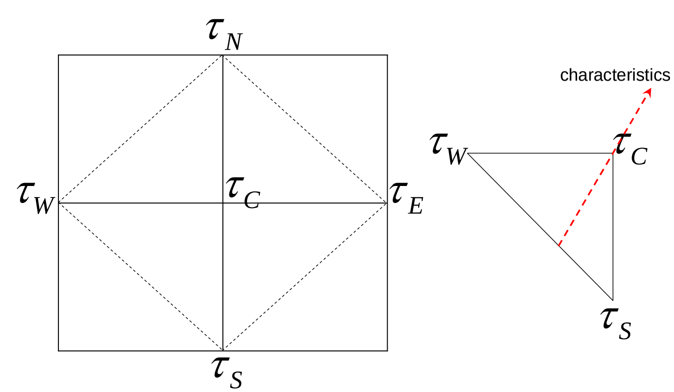
The upwind scheme: assume

$\tau_W T_0(W) = \min(\tau_W T_0(W), \tau_E T_0(E))$, $\tau_S T_0(S) = \min(\tau_S T_0(S), \tau_N T_0(N))$

The factored eikonal equation is discretized as

$$T_0^2(C)|(\frac{\tau_C-\tau_W}{h}, \frac{\tau_C-\tau_S}{h})|^2 + 2T_0(C)\tau_C \nabla T_0(C) \cdot (\frac{\tau_C-\tau_W}{h}, \frac{\tau_C-\tau_S}{h})$$

$$+\{\tau_C^2 - \alpha^2(C)\}S_0^2(C) = 0$$



Gauss-Seidel Iteration

A solution τ_C satisfies the causality condition if

$$\tau_C T_0(C) \geq \tau_W T_0(W) \quad \text{and} \quad \tau_C T_0(C) \geq \tau_S T_0(S).$$

- Initialization: assign boundary values.

- Gauss-Seidel iteration: sweep the domain with four 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 gird point, solve the discretized equation for two possible roots, $\tau_{C,1}$ and $\tau_{C,2}$.

- If there are two real roots, $\tau_{C,1}$ and $\tau_{C,2}$, then
  - if both $\tau_{C,1}$, $\tau_{C,2}$ satisfy causality condition, then $T_C = \text{min}(\tau_{C,1} T_0(C), \tau_{C,2} T_0(C), T_C)$.
  - else if $\tau_{C,1}$ satisfies the causality condition, then $T_C = \text{min}(\tau_{C,1} T_0(C), T_C)$.
  - else if $\tau_{C,2}$ satisfies the causality condition, then $T_C = \text{min}(\tau_{C,2} T_0(C), T_C)$.
  - else if none of the two roots satisfies the causality condition, then $T_C = \text{min}\{T_C, T_W + |WC| S(C), T_S + |SC| S(C)\}$.
- else, $T_C = min\{T_C, T_W + |WC| S(C), T_S + |SC| S(C)\}$.

Example 1: Constant gradient of slowness squared: a point source at $\mathbf{x}_0$ and

$$S^2(\mathbf{x}) = S_0^2 + 2\,\mathbf{g}_0 \cdot (\mathbf{x} - \mathbf{x}_0)$$

with constant S_0 and $\mathbf{g}_0$, the analytical solution is

$$T(\mathbf{x}) = \bar{S}^2\,\sigma - |\mathbf{g}_0|^2\,\frac{\sigma^3}{6}\;,$$

where

$$\bar{S}(\mathbf{x}) = \sqrt{\frac{S^2(\mathbf{x}) + S_0^2}{2}} = \sqrt{S_0^2 + \mathbf{g}_0 \cdot (\mathbf{x} - \mathbf{x}_0)}\;.$$

$$\sigma^2 = \frac{2\left(\bar{S}^2 - \sqrt{\bar{S}^4 - |\mathbf{g}_0|^2\;|\mathbf{x} - \mathbf{x}_0|^2}\right)}{|\mathbf{g}_0|^2} = \frac{2\;|\mathbf{x} - \mathbf{x}_0|^2}{\bar{S}^2 + \sqrt{\bar{S}^4 - |\mathbf{g}_0|^2\;|\mathbf{x} - \mathbf{x}_0|^2}}\;,$$

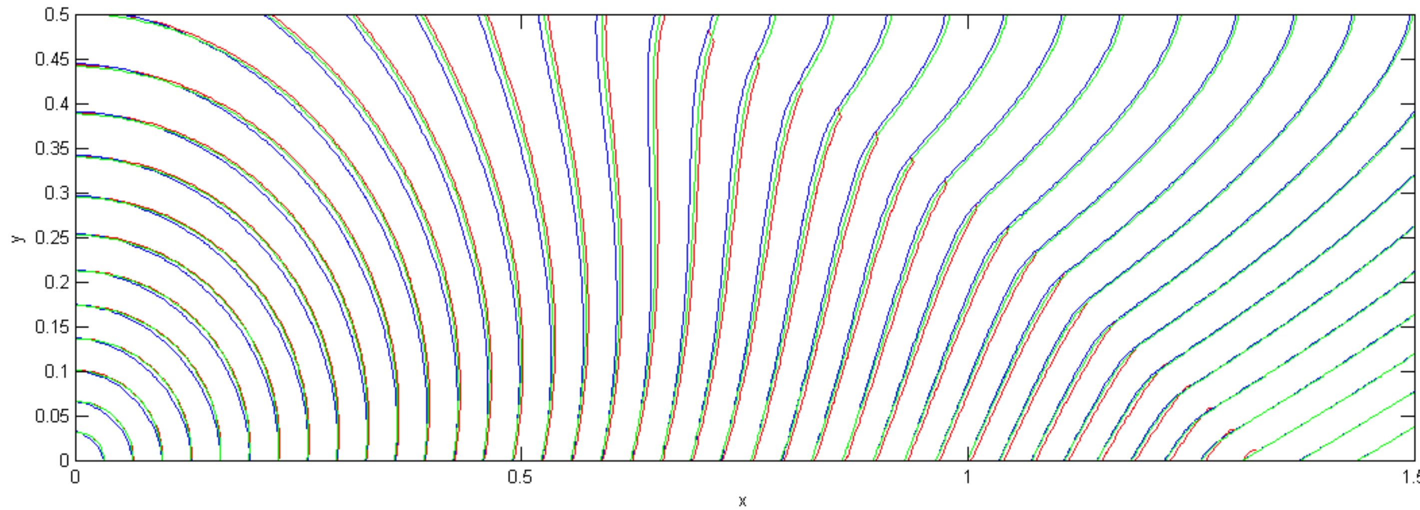
Our numerical setup: domain=$[0, 1.5] \times [0, 0.5]$,

$$\mathbf{x}_0 = (0, 0), \qquad S_0 = 2\,\text{s/km}, \qquad \mathbf{g}_0 = \{0, -3\}\,\text{s}^2/\text{km}^3.$$

Constant gradient slowness squared solution.

Factored eikonal equation		
Mesh	Max_error in domain$[0, 0.5]^2$	#iteration
150x50	0.0050124	3
300x100	0.0025031	3
600x200	0.0012507	3
1200x400	0.0006251	3

Original eikonal equation		
Mesh	Max_error in domain$[0, 0.5]^2$	#iteration
150x50	0.0213824	3
300x100	0.0129489	3
600x200	0.0076360	3
1200x400	0.0044102	3



Red: analytical solution. Blue:original eikonal solution. Green: factored eikonal solution.

Example 2: Constant gradient of velocity:
a point source at $\mathbf{x}_0$ and

$$\frac{1}{S(\mathbf{x})} = \frac{1}{S_0} + \mathbf{G}_0 \cdot (\mathbf{x} - \mathbf{x}_0)$$

with constant S_0 and $\mathbf{G}_0$, the analytical solution is

$$T(\mathbf{x}) = \frac{1}{|\mathbf{G}_0|} \operatorname{arccosh}\left(1 + \frac{1}{2} S(\mathbf{x})\, S_0\, |\mathbf{G}_0|^2\, |\mathbf{x} - \mathbf{x}_0|^2\right) ,$$

where arccosh is the inverse hyperbolic cosine function

$$\operatorname{arccosh}(z) = \ln\left(z + \sqrt{z^2 - 1}\right) .$$

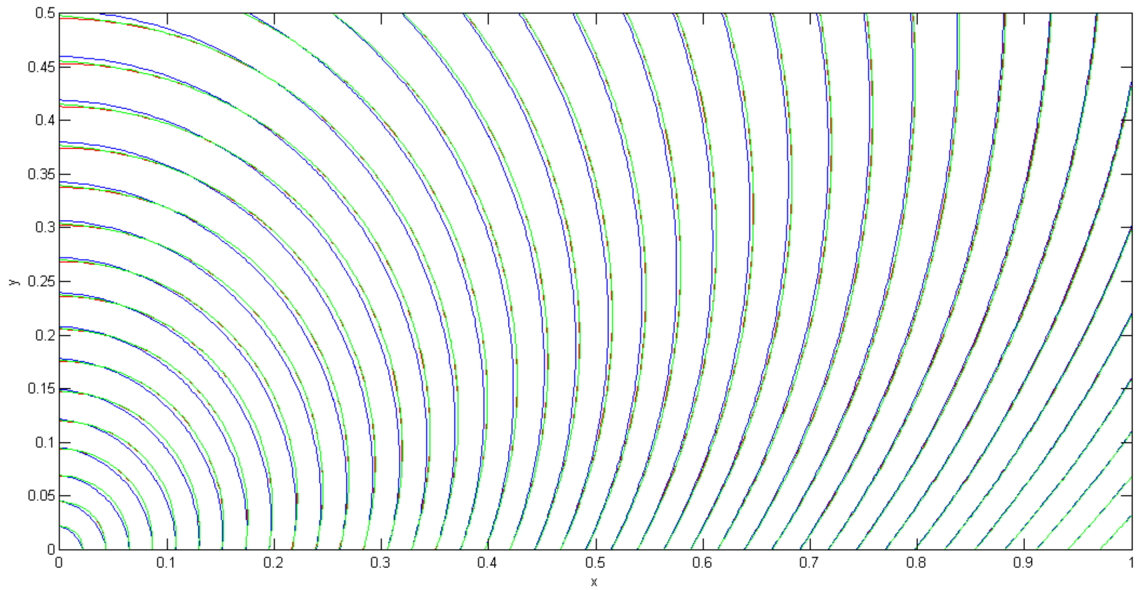
Our numerical setup: domain=$[0, 1] \times [0, 0.5]$,

$$\mathbf{x}_0 = (0, 0), \qquad S_0 = 2\,\text{s/km}, \qquad \mathbf{G}_0 = \{0, -1\}\,1/\text{s}.$$

Constant gradient of velocity solution.

Factored eikonal equation		
Mesh	Max_error	#iteration
80x40	0.0062109	3
160x80	0.0031152	3
320x160	0.0015600	3
1200x400	0.0007806	3

Original eikonal equation		
Mesh	Max_error	#iteration
80x40	0.0253779	3
160x80	0.0153785	3
320x160	0.0090909	3
640x320	0.0052675	3



Red: analytical solution. Blue:original eikonal solution. Green: factored eikonal solution.

Example 3: Use the analytical solution of example 2 as the T_0 for example 1:

Factored eikonal equation		
Mesh	Max_error in $[0, 0.5]^2$	#iteration
150x50	0.0050124	3
300x100	0.0025031	3
600x200	0.0012507	3
1200x400	0.0006251	3

Example 4: Use the analytical solution of example 1 as the T_0 for example 2:

maximum error		
Mesh	Max_error in $[0, 0.5]^2$	#iteration
81x81	0.0031152	3
161x161	0.0015600	3
321x321	0.0007806	3
641x641	0.0003904	3

Any T_0 works as long as it is a good approximation.

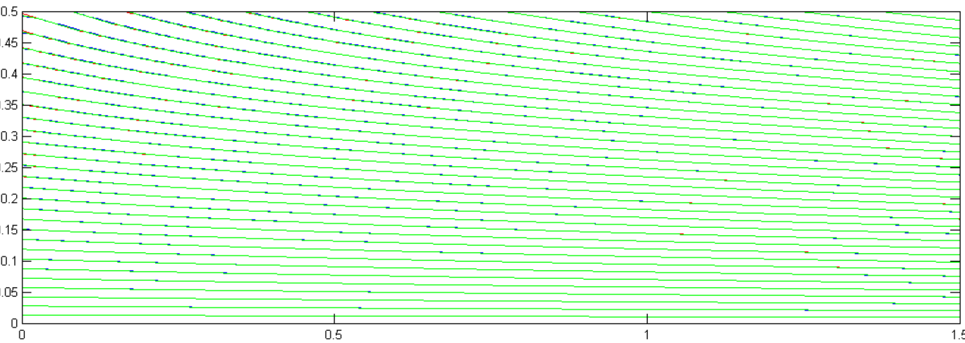
Example 5: Plane wave source: the analytical solution is:

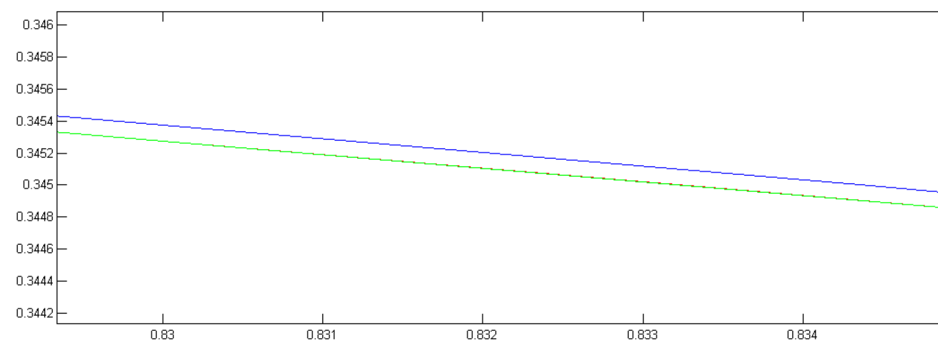
$$T(\mathbf{x}) = \frac{\sqrt{\omega}}{3} \frac{(3S_1^4 + g_1^2 z^2) z}{\sqrt{S_1^2 (S_1^4 + 3g_1^2 z^2) + (S_1^4 - g_1^2 z^2)^{3/2}}}$$

where $\mathbf{x} = (x, z), g_0 = (g_x, g_z), g_1 = \sqrt{4g_x^4 + g_z^2}$,
and $S_1 = \sqrt{S^2(x, z) - g_z z} = \sqrt{S_0^2 + 2g_x(x - x_0) + g_z z}$.

Our numerical setup:

- computational domain: $[0, 1.5] \times [0, 0.5]$
- plane-wave source on $z = 0$
- $g_0 = (1, -3)\mathrm{s}^2/\mathrm{km}^3$
- $S_0 = 2\mathrm{s/km}$, $T_0(x, z) = S_0 z$.





Red: analytical solution. Blue: original eikonal solution. Green: factored eikonal solution.

Plane-wave source with $T_0(x, z) = S_0 z$

maximum error of Factored eikonal equation		
Mesh	Max_error in domain$[0.75, 1.5] \times [0, 0.5]$	#iteration
150x50	0.0003085	2
300x100	0.0001540	2
600x200	0.0000770	2
1200x400	0.0000385	2

maximum error of original eikonal equation		
Mesh	Max_error in domain$[0.75, 1.5] \times [0, 0.5]$	#iteration
150x50	0.0039623	2
300x100	0.0019791	2
600x200	0.0009891	2
1200x400	0.0004944	2

Even with no singularity the factored eikonal equation can achieve better accuracy

Remarks

- Our fast sweeping algorithm for factored eikonal equation can achieve much better accuracy while maintaining the same efficiency of the original fast sweeping method.

- Recent work by Luo and Qian extend the method to anisotropic eikonal equation as well as coupled with amplitude computation.

Deferred correction

Goal: post-processing the numerical solution from 1st order scheme to achieve high order accuracy.
Key idea: utilize the PDE to approximate the Taylor expansion and information from 1st order solution.

$$\phi(X+\delta X) = \phi(X)+\delta X\cdot\nabla\phi(X)+\frac{1}{2} < \delta X, D^2\phi(X)\cdot\delta X > +O(\delta X^3)$$

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

$$\begin{aligned}&\tfrac{1}{2} < \delta X, D^2\phi(X)\cdot\delta X > = \tfrac{1}{2\,n^4}(\delta X\cdot\nabla\phi)^2 < \nabla\phi, D^2\phi\cdot\nabla\phi > \\ &+\tfrac{1}{n^4}(\delta X\cdot\nabla\phi)(\delta X\cdot\nabla\phi^{\perp}) < \nabla\phi^{\perp}, D^2\phi\cdot\nabla\phi > \\ &+\tfrac{1}{2\,n^4}(\delta X\cdot\nabla\phi^{\perp})^2 < \nabla\phi^{\perp}, D^2\phi\cdot\nabla\phi^{\perp} >\end{aligned}$$

From the eikonal equation: $D^2\phi(X)\cdot\nabla\phi(X) = n(X)\nabla n(X)$.
The only unknown among the second order terms is the curvature of the wave front: $C = < \nabla\phi^{\perp}, D^2\phi\cdot\nabla\phi^{\perp} >$.

Deferred correction (continued)

Denote $\nabla\phi = (a, b)$, for $n(x) = 1$,

$$\phi(X + \delta X) = \phi(X) + \delta X \cdot (a, b) + \frac{C}{2\,n^4}(\delta X \cdot (-b, a))^2 + O(\delta X^3)$$

If the characteristic direction is up-right, we use the four grid points $(\{0,0\},\{-h,0\},\{0,-h\},\{-h,-h\})$ to determine (a, b, C) from

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

which is quite hard.

Deferred correction (continued)

Key simplification: use 1st order information.

Let a_1, b_1 be the numerical gradient from 1st order solution.

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

which is a linear equation for (a, b, C).

The basic algorithm:
Do the second order correction according to the ordering from the first order solution.

Remarks:
1. Chose an appropriate three point stencils depending on (a_1, b_1).
2. First order solution, its gradient, ordering are all used.
3. One pass algorithm.

Issues for high order methods for stationary HJ equations

- No explicit time dependence or one special direction.

- Appropriate discretization.

1.consistent, stable, and convergent to the viscosity solution.
2. achieves high order accuracy away from singularities and suppresses oscillations at/across singularities.

- A large system of nonlinear equations has to be solved.

For iterative methods: convergence and how fast???
Some approaches:
1. Robust low order scheme can provide good initial guess and other important information.
2. upwind schemes combined with right orderings can help convergence for hyperbolic problems.

- Boundary condition

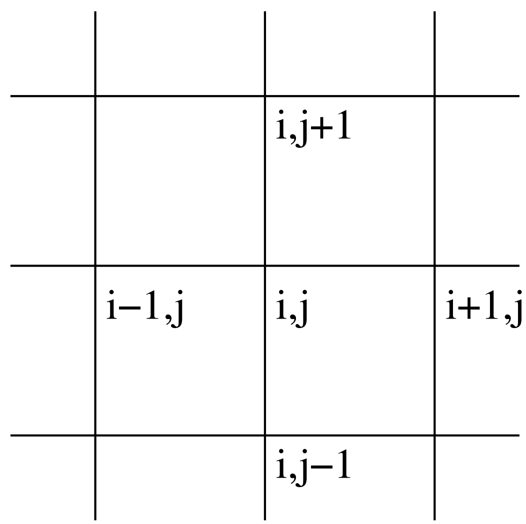
Upwind scheme for eikonal equation on rectangular grids.

Isotropic eikonal equation: $\|\nabla\phi(\boldsymbol{x})\| = c(\boldsymbol{x})$ with B.C.

5 point stencil upwind scheme:

$$[(\phi_{i,j}-\phi_{xmin})^{+}]^2+[(\phi_{i,j}-\phi_{ymin})^{+}]^2=h^2c_{i,j}^2, \qquad i,j=1,2,\ldots$$

where $\phi_{xmin}=\min(\phi_{i-1,j},\ \phi_{i+1,j}), \phi_{ymin}=\min(\phi_{i,j-1},\phi_{i,j+1})$.



High order methods: using Godunov flux

High order method: using the same Godunov flux with high order WENO approximation of $D^{\pm}$.

$$[(\frac{\phi_{i,j}^{new}-\phi_{i,j}^{(xmin)}}{h})^{+}]^2+[(\frac{\phi_{i,j}^{new}-\phi_{i,j}^{(ymin)}}{h})^{+}]^2=f_{i,j}^2$$

where
$$\begin{cases}\phi_{i,j}^{(xmin)}=\min(\phi_{i,j}^{old}-h\cdot(\phi_x)_{i,j}^{-},\phi_{i,j}^{old}+h\cdot(\phi_x)_{i,j}^{+}),\\ \phi_{i,j}^{(ymin)}=\min(\phi_{i,j}^{old}-h\cdot(\phi_y)_{i,j}^{-},\phi_{i,j}^{old}+h\cdot(\phi_y)_{i,j}^{+}).\end{cases}$$

$\phi_{i,j}\pm h\cdot(\phi_x)_{i,j}^{\pm}$ and $\phi_{i,j}\pm h\cdot(\phi_y)_{i,j}^{\pm}$ can be considered as approximations to $\phi_{i\pm1,j}$ and $\phi_{i,j\pm1}$ respectively. Here $(\phi_x)_{i,j}^{\pm}$ and $(\phi_y)_{i,j}^{\pm}$ are computed using WENO schemes. When the iterations converge, we have solved the system

$$\sqrt{\max\{[(\phi_x)_{i,j}^{-}]^{+},[-(\phi_x)_{i,j}^{+}]^{+}\}^2+\max\{[(\phi_y)_{i,j}^{-}]^{+},[-(\phi_y)_{i,j}^{+}]^{+}\}^2}=f_{i,j}$$

High order methods: using Lax-Friedrichs

$$\widehat{H}(u^-,u^+;v^-,v^+) = H\left(\frac{u^-+u^+}{2},\frac{v^-+v^+}{2}\right) - \alpha^x \frac{u^+-u^-}{2} - \alpha^y \frac{v^+-v^-}{2}$$

where $\alpha^x = \max_{\substack{A\le u\le B\\ C\le v\le D}} |H_1(u,v)|$, $\quad \alpha^y = \max_{\substack{A\le u\le B\\ C\le v\le D}} |H_2(u,v)|$.

The 1st order L-F scheme by Kao, Osher and Qian:

$$\begin{aligned}\phi_{i,j}^{new} &= \left(\frac{1}{\frac{\alpha_x}{h_x}+\frac{\alpha_y}{h_y}}\right)\Bigg[f - H\left(\frac{\phi_{i+1,j}-\phi_{i-1,j}}{2h_x},\frac{\phi_{i,j+1}-\phi_{i,j-1}}{2h_y}\right) \\ &\quad + \alpha_x \frac{\phi_{i+1,j}+\phi_{i-1,j}}{2h_x} + \alpha_y \frac{\phi_{i,j+1}+\phi_{i,j-1}}{2h_y}\Bigg]\end{aligned}$$

replace $\phi_{i\pm1,j}$ and $\phi_{i,j\pm1}$ by $\phi_{i,j} \pm h\cdot(\phi_x)_{i,j}^{\pm}$ and $\phi_{i,j} \pm h\cdot(\phi_y)_{i,j}^{\pm}$ respectively, where $(\phi_x)_{i,j}^{\pm}$ and $(\phi_y)_{i,j}^{\pm}$ are computed using WENO:

$$\begin{aligned}\phi_{i,j}^{new} &= \left(\frac{1}{\frac{\alpha_x}{h_x}+\frac{\alpha_y}{h_y}}\right)\Bigg[f - H\left(\frac{(\phi_x)_{i,j}^- + (\phi_x)_{i,j}^+}{2},\frac{(\phi_y)_{i,j}^- + (\phi_y)_{i,j}^+}{2}\right) \\ &\quad + \alpha_x \frac{(\phi_x)_{i,j}^+ - (\phi_x)_{i,j}^-}{2} + \alpha_y \frac{(\phi_y)_{i,j}^+ - (\phi_y)_{i,j}^-}{2}\Bigg] + \phi_{i,j}^{old}.\end{aligned}$$

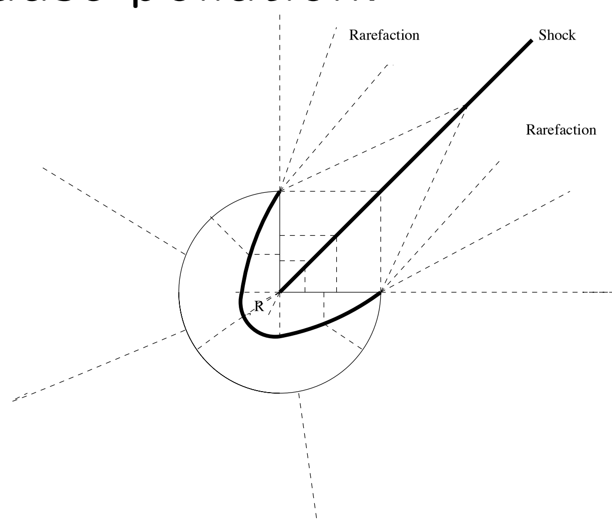
Remarks

- Godunov: needs local flux solver; more upwind; fewer iterations; need a first order sweep for good initial guess.

- L-F: works for general Hamiltonian; less upwind; more iterations; less sensitive to initial guess.

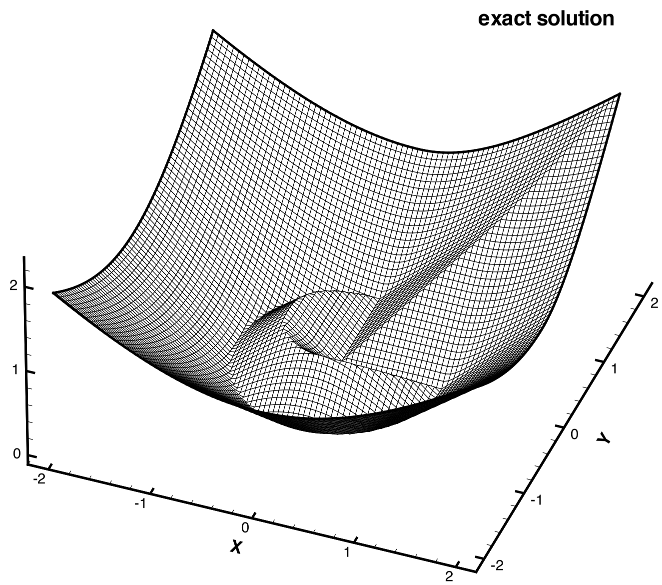
- The number of iterations depends on grid size but is much fewer than time marching.

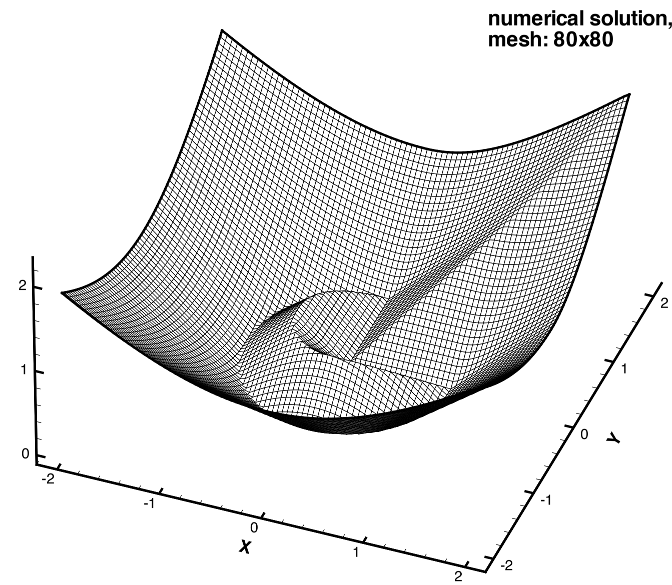
- Retain high order in certain regions. Shocks give no pollution. Rarefactions cause pollution.



Accuracy in different regions. Godunov Hamiltonian.

	smooth region		whole region		rarefaction		
mesh	L^1 error	order	L^1 error	order	L^1 error	order	iter
80×80	1.03E-5	–	1.70E-4	–	5.71E-4	–	38x4
160×160	1.32E-6	2.96	6.09E-5	1.48	2.26E-4	1.33	26x4
320×320	2.89E-7	2.19	1.72E-5	1.82	5.83E-5	1.96	34x4
640×640	4.03E-8	2.84	4.13E-6	2.06	1.03E-5	2.50	53x4





High order methods

3rd order Godunov method

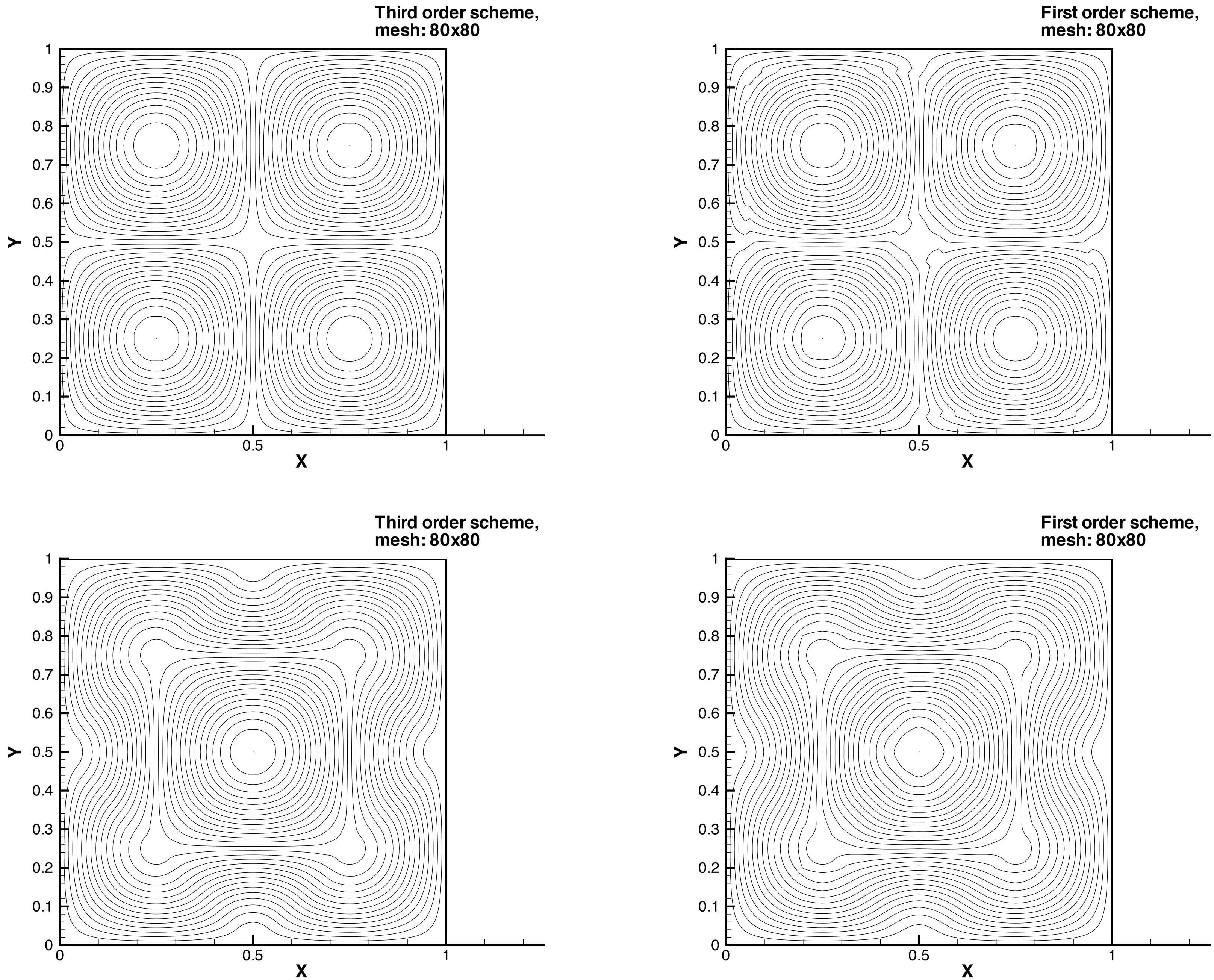
	SFS case a (smooth)			SFS case b (non-smooth)		
mesh	L^1 error	order	iteration #	L^1 error	order	iteration #
80×80	1.24E-4	–	16x4	1.14E-3	–	18x4
160×160	5.78E-6	4.43	23x4	1.98E-4	2.53	23x4
320×320	3.27E-7	4.14	29x4	2.95E-5	2.74	31x4
640×640	3.56E-8	3.20	52x4	7.28E-6	2.02	57x4

3rd order Lax-Friedrichs method $\alpha_x = \alpha_y = 1$.

	SFS case a (smooth)			SFS case b (non-smooth)		
mesh	L^1 error	order	iteration #	L^1 error	order	iteration #
80×80	1.44E-4	–	22x4	2.94E-3	–	28x4
160×160	7.68E-6	4.23	33x4	8.35E-4	1.81	38x4
320×320	4.93E-7	3.96	58x4	2.20E-4	1.92	75x4
640×640	5.58E-8	3.14	111x4	4.67E-5	2.24	136x4

Comparision between time-marching and fast sweeping with 3rd order WENO

	3rd-order Time-marching			3rd-order fast sweeping		
mesh	L^1 error	order	iteration #	L^1 error	order	iteration #
80×80	1.14E-3	–	70×3	1.14E-3	–	18×4
160×160	1.98E-4	2.53	133×3	1.98E-4	2.53	23×4
320×320	2.96E-5	2.74	281×3	2.95E-5	2.74	31×4
640×640	7.31E-6	2.02	578×3	7.28E-6	2.02	57×4


Third order scheme,
mesh: 80x80
First order scheme,
mesh: 80x80
Third order scheme,
mesh: 80x80
First order scheme,
mesh: 80x80
Y
X
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Travel-time for elastic wave

The quasi-P and the quasi-SV slowness surfaces are:

$$c_1\phi_x^4 + c_2\phi_x^2\phi_y^2 + c_3\phi_y^4 + c_4\phi_x^2 + c_5\phi_y^2 + 1 = 0,$$

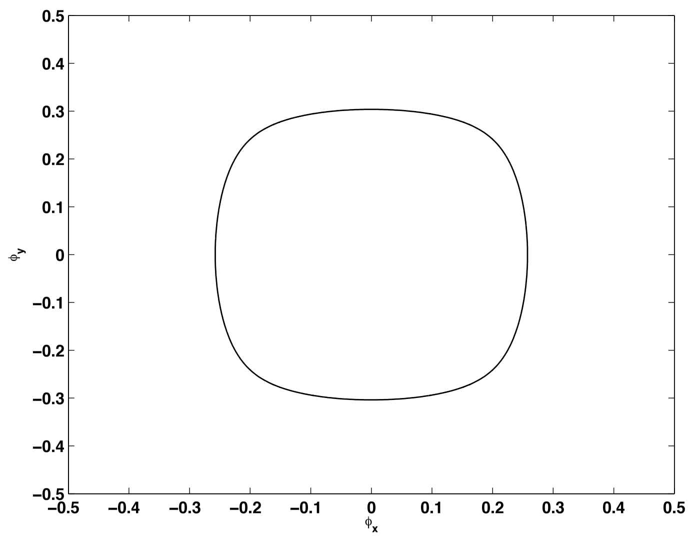
where c_i are defined by elastic tensor a_{ij}.
The quasi-P wave eikonal equation (convex) is:

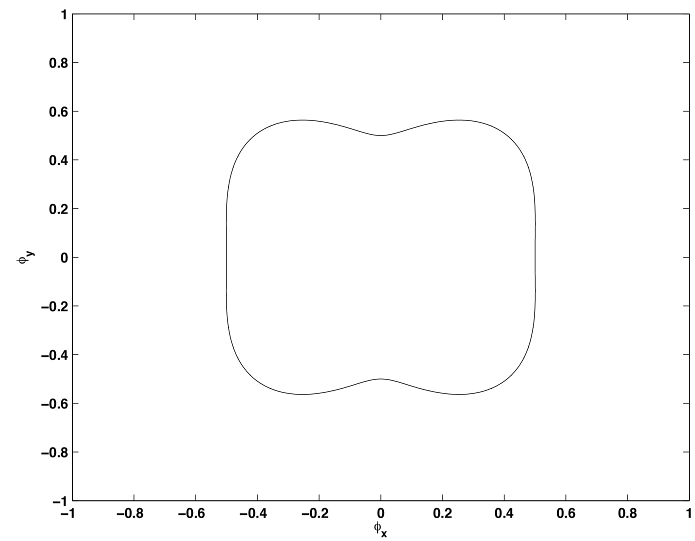
$$\sqrt{-\frac{1}{2}(c_4\phi_x^2 + c_5\phi_y^2) + \sqrt{\frac{1}{4}(c_4\phi_x^2 + c_5\phi_y^2)^2 - (c_1\phi_x^4 + c_2\phi_x^2\phi_y^2 + c_3\phi_y^4)}} = 1$$

The quasi-SV wave eikonal equation (non-convex) is:

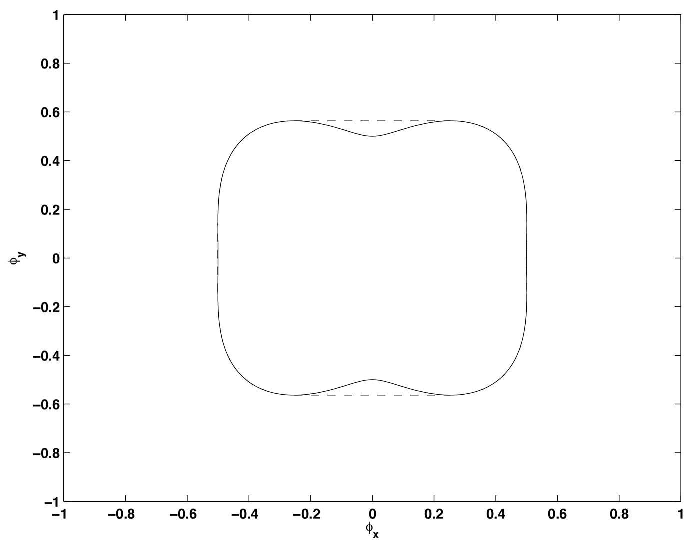
$$\sqrt{-\frac{1}{2}(c_4\phi_x^2 + c_5\phi_y^2) - \sqrt{\frac{1}{4}(c_4\phi_x^2 + c_5\phi_y^2)^2 - (c_1\phi_x^4 + c_2\phi_x^2\phi_y^2 + c_3\phi_y^4)}} = 1$$

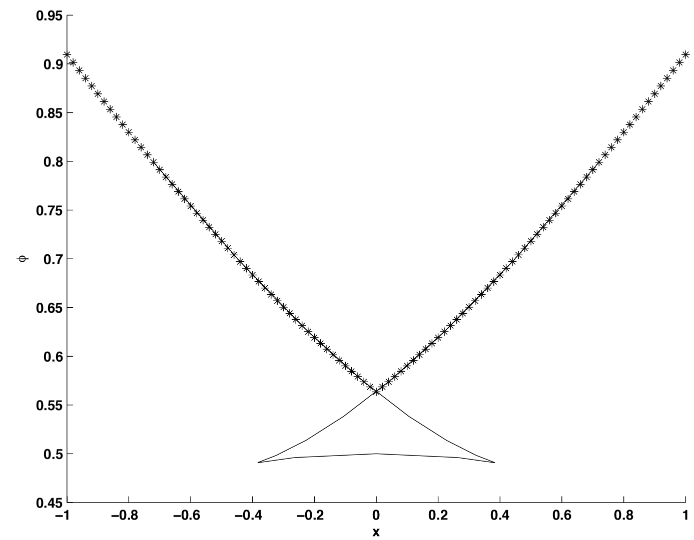
Slowness surfaces for quasi-P wave (left) and quasi-SV wave (right)





Convexification of the Hamiltonian for the quasi-SV wave



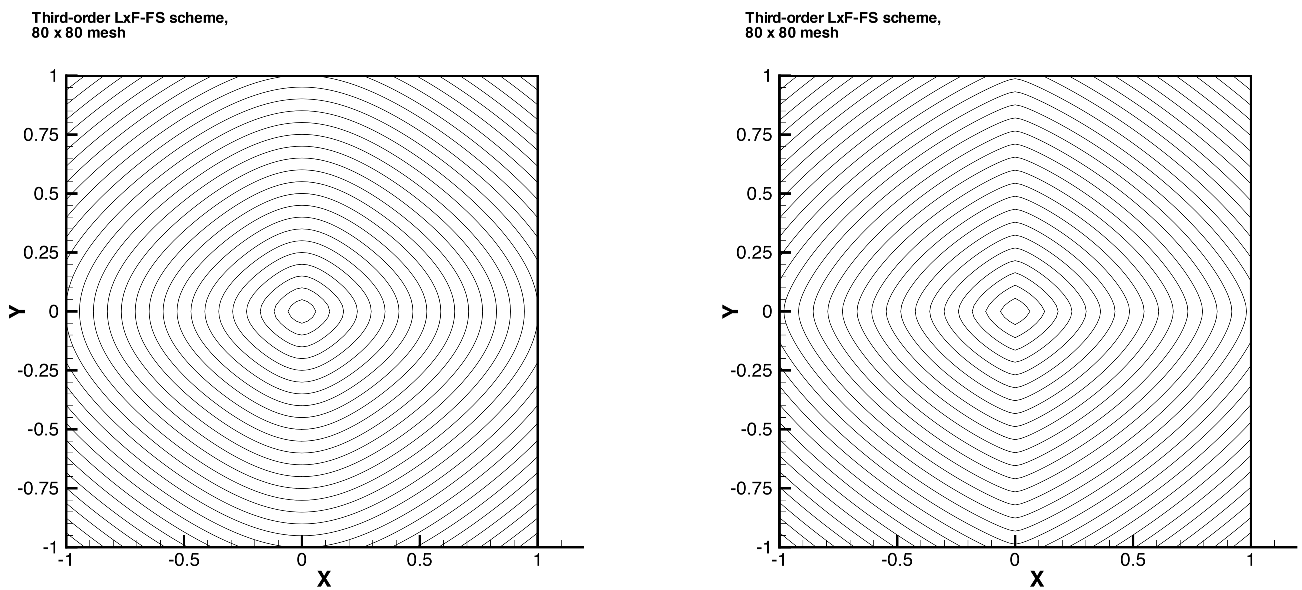


mesh	L^1 error	order	L^∞ error	order	iteration #
40×40	7.23E-4	–	3.07E-3	–	27x4
80×80	9.33E-5	2.95	5.81E-4	2.40	36x4
160×160	9.58E-6	3.28	6.72E-5	3.11	58x4
320×320	1.27E-6	2.92	8.94E-6	2.91	98x4
640×640	1.62E-7	2.97	1.16E-6	2.95	178x4

Quasi-P wave, Lax-Friedrichs method, $\alpha_x = \alpha_y = 4$.

mesh	L^1 error	order	L^∞ error	order	iteration #
	smooth region 0.15 away from $x = 0$ and $y = 0$				
40×40	9.20E-4	–	3.65E-3	–	42x4
80×80	6.21E-5	3.89	6.32E-4	2.53	34x4
160×160	2.39E-6	4.70	2.28E-5	4.79	57x4
320×320	4.61E-7	2.37	1.47E-6	3.96	99x4
640×640	5.97E-8	2.95	2.97E-7	2.30	181x4
	Whole region				
40×40	2.19E-3	–	1.53E-2	–	42x4
80×80	6.09E-4	1.84	8.02E-3	0.93	34x4
160×160	1.62E-4	1.91	4.19E-3	0.94	57x4
320×320	3.70E-5	2.13	2.00E-3	1.06	99x4
640×640	1.02E-5	1.86	8.21E-4	1.29	181x4

Quasi-SV wave, Lax-Friedrichs method, $\alpha_x = \alpha_y = 2$.



travel time for quasi-P wave (left) and quasi-SV wave (right)

Summary 1

The finite difference (WENO) based high order method is

- Pros:
  simple

- Cons:
  1. Uses wide stencils and is difficult to extend to unstructured mesh.
  2. Number of iterations depends on grid size.

Discontinuous Galerkin (DG) based high order method

Motivation: apply DG method to develop a **compact, upwind and versatile** high order discretization.

Notation: partition $\Omega = \cup_{1\le i\le I, 1\le j\le J} I_{ij}$ where $I_{ij} = I_i \times J_j$ and $I_i = [x_{i-1/2}, x_{i+1/2}]$, $J_j = [y_{j-1/2}, y_{j+1/2}]$. The centers of I_i, J_j are denoted by $x_i = \frac{1}{2}(x_{i-1/2} + x_{i+1/2})$ and $y_j = \frac{1}{2}(y_{j-1/2} + y_{j+1/2})$, and the lengths of I_i, J_j are denoted by $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and $\Delta y_j = y_{j+1/2} - y_{j-1/2}$.
Define the piecewise linear approximation space as

$$V_h^1 = \{v : v|_{I_{ij}} \in P^1(I_{ij}), \forall i, j\}$$

where $P^1(I_{ij})$ is the set of linear polynomials on I_{ij}.

Local solver

Find $\phi_h \in V_h^1$, such that (following Cheng & Shu (JCP, 07) for time dependent HJ)

$$
\begin{aligned}
&\int_{I_{ij}} |\nabla \phi_h| w_h(x,y)dxdy \\
&+\alpha_{r,ij} \int_{J_j} [\phi_h](x_{i+\frac{1}{2}}, y) w_h(x^-_{i+\frac{1}{2}}, y)dy + \alpha_{l,ij} \int_{J_j} [\phi_h](x_{i-\frac{1}{2}}, y) w_h(x^+_{i-\frac{1}{2}}, y)dy \\
&+\alpha_{t,ij} \int_{I_i} [\phi_h](x, y_{j+\frac{1}{2}}) w_h(x, y^-_{j+\frac{1}{2}})dx + \alpha_{b,ij} \int_{I_i} [\phi_h](x, y_{j-\frac{1}{2}}) w_h(x, y^+_{j-\frac{1}{2}})dx \\
&= \int_{I_{ij}} f(x,y) w_h(x,y)dxdy, \qquad \forall\, i,j, \ \forall\, w_h \in V_h^1
\end{aligned}
$$

where $[\phi_h]$ denotes the jump of ϕ_h across the cell interface.

Note: $\alpha_{r,ij}, \alpha_{l,ij}, \alpha_{t,ij}, \alpha_{b,ij}$ are constants which only depend on the numerical solutions in the neighboring cells of I_{ij}. They are crucial to provide: stability, viscosity and causality for the discretization.

Local solver (continued)

The piecewise linear approximation $\phi_h|_{I_{ij}} = \bar{\phi}_{ij} + u_{ij}X_i + v_{ij}Y_j$, where $\bar{\phi}_{ij}$, u_{ij}, v_{ij} are the unknowns, and $X_i = \frac{x-x_i}{h}$, $Y_j = \frac{y-y_j}{h}$.
According to the causality we choose

$$\alpha_{l,ij} = \begin{cases} \max(0, u_{i-1,j}/(hf_{i-1,j})) & \text{when } \bar{\phi}_{i-1,j} \le \bar{\phi}_{i+1,j}; \\ 0 & \text{when } \bar{\phi}_{i-1,j} > \bar{\phi}_{i+1,j}, \end{cases}$$

$$\alpha_{r,ij} = \begin{cases} 0 & \text{when } \bar{\phi}_{i-1,j} \le \bar{\phi}_{i+1,j}; \\ \min(0, u_{i+1,j}/(hf_{i+1,j})) & \text{when } \bar{\phi}_{i-1,j} > \bar{\phi}_{i+1,j}, \end{cases}$$

$$\alpha_{b,ij} = \begin{cases} \max(0, v_{i,j-1}/(hf_{i,j-1})) & \text{when } \bar{\phi}_{i,j-1} \le \bar{\phi}_{i,j+1}; \\ 0 & \text{when } \bar{\phi}_{i,j-1} > \bar{\phi}_{i,j+1}, \end{cases}$$

$$\alpha_{t,ij} = \begin{cases} 0 & \text{when } \bar{\phi}_{i,j-1} \le \bar{\phi}_{i,j+1}; \\ \min(0, v_{i,j+1}/(hf_{i,j+1})) & \text{when } \bar{\phi}_{i,j-1} > \bar{\phi}_{i,j+1}, \end{cases}$$

where $f_{i,j} = f(x_i, y_j)$.
Denote $H_1 = \frac{\partial H}{\partial \phi_x}$, $H_2 = \frac{\partial H}{\partial \phi_y}$. $\alpha_{r,ij}, \alpha_{l,ij}, \alpha_{t,ij}, \alpha_{b,ij}$ approximate $H_1(\nabla\phi_h)$ and $H_2(\nabla\phi_h)$ in the four **neighboring** cells of I_{ij}. They are different from Cheng & Shu 07.

Local solver (continued)

Local system of nonlinear equations for $\bar{\phi}_{ij}$, u_{ij}, v_{ij}:

$$\begin{aligned} \sqrt{u_{ij}^2 + v_{ij}^2} + \gamma_{ij}\bar{\phi}_{ij} + \beta_{ij}u_{ij} + \lambda_{ij}v_{ij} &= R_{1,ij} \\ 12\beta_{ij}\bar{\phi}_{ij} + \zeta_{ij}u_{ij} &= R_{2,ij} \\ 12\lambda_{ij}\bar{\phi}_{ij} + \eta_{ij}v_{ij} &= R_{3,ij} \end{aligned}$$

where $\beta_{ij}, \lambda_{ij}, \gamma_{ij}, \zeta_{ij}, \eta_{ij}, R_{1,ij}, R_{2,ij}, R_{3,ij}$ are constants that depend on $\alpha_{r,ij}, \alpha_{l,ij}, \alpha_{t,ij}, \alpha_{b,ij}, f$, **which do not depend on** $\phi_h|_{I_{ij}}$**.** $\gamma_{ij}, \zeta_{ij}, \eta_{ij}$ are either all zero or all positive.

Local solver (continued)

1. If $\gamma_{ij}, \zeta_{ij}, \eta_{ij}$ are all zero, no update.

2. If $\gamma_{ij}, \zeta_{ij} > 0, \eta_{ij} > 0$, and

   - two sets of *real* solutions $(\bar{\phi}_{ij}^{new}, u_{ij}^{new}, v_{ij}^{new})$: update $\phi_h|_{I_{ij}}$ if the following conditions are satisfied:

$$\begin{cases} \bar{\phi}_{ij}^{new} \geq \bar{\phi}_{i-1,j} & \text{and} \quad u_{ij}^{new} \geq 0, \quad \text{if} \quad \alpha_{l,ij} > 0 \\ \bar{\phi}_{ij}^{new} \geq \bar{\phi}_{i+1,j} & \text{and} \quad u_{ij}^{new} \leq 0, \quad \text{if} \quad \alpha_{r,ij} < 0 \\ \bar{\phi}_{ij}^{new} \geq \bar{\phi}_{i,j-1} & \text{and} \quad v_{ij}^{new} \geq 0, \quad \text{if} \quad \alpha_{b,ij} > 0 \\ \bar{\phi}_{ij}^{new} \geq \bar{\phi}_{i,j+1} & \text{and} \quad v_{ij}^{new} \leq 0, \quad \text{if} \quad \alpha_{t,ij} < 0 \end{cases}$$

     If both sets of the solutions satisfy the above conditions, choose the one with the smaller value of $\bar{\phi}_{ij}^{new}$;

   - two sets of *complex* solutions: no update.

Summary of the DG based method

- pros:
  - Compact.
  - Flexible.
  - Fast convergence.
- cons:
  - Enforcement of causality.
  - Boundary condition.

Boundary condition

Use the PDE and tangential derivatives at the boundary to get other derivatives of the solution at the boundary (Huang & Shu)

Suppose the boundary $\Gamma = \{(x, y) : x = 0\}$ and $\phi(0, y) = g(y)$. $\phi_y, \phi_{yy}, ...$ are given by $g', g'', ...$ Now find $\phi_x, \phi_{xx}, ...$ using the PDE.

1. Find $\phi_x(0, y)$ from the PDE $H(\phi_x(0, y), g'(y)) = f(0, y)$ and the influx condition $H_u(\phi_x(0, y), g'(y)) > 0$.

2. Find $\phi_{xx}(0, y)$ by

$$H_u(\phi_x(0, y), g'(y))\phi_{xy}(0, y) + H_v(\phi_x(0, y), g'(0, y))g''(y) = f_y(0, y)$$

$\Rightarrow \phi_{xy}(0, y)$.

$$H_u(\phi_x(0, y), g'(y))\phi_{xx}(0, y) + H_v(\phi_x(0, y), g'(0, y))\phi_{x,y}(0, y) = f_x(0, y)$$

$\Rightarrow \phi_{xx}(0, y)$.

Some interesting problems

- Error analysis for numerical gradient and the optimal path.
- Non-convex case.
- Hyperbolic conservation law.
- Convection dominated diffusion problem.
- Interesting applications ...