The Fast Sweeping Method

Hongkai Zhao

Department of Mathematics
University of California, Irvine
Highlights of fast sweeping method

- Simple (A Gauss-Seidel type of iterative method).
- Optimal complexity.
- A truly nonlinear method.
- A local solver for convex Hamiltonian on arbitrary mesh.
Outline

Fast sweeping method (FSM):

- FSM for Eikonal equation on rectangular grid (convergence and error estimates).
- FSM on triangular mesh. (with J. Qian and Y. Zhang)
- FSM for general convex Hamilton-Jacobi equations (with J. Qian and Zhang).
- High order FSM (with Zhang and Qian).
- Parallel implementations.
- Understanding FSM in the framework of iterative methods.
The Eikonal equation: \[ |\nabla u(x)| = c(x), \quad u(x) = 0, \quad x \in S \]

\(u(x)\) is the first arrival time for a front starting at \(S\) with a propagation speed \(v_n = \frac{1}{c(x)}\).
Two crucial ingredients:

- Appropriate discretization (local solver).
  Design numerical Hamiltonian that keeps the causality, 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.
Other related work

- Danielson’s algorithm (1980).


- Boué and Dupuis, fast sweeping method for stochastic control (1999).

- Jameson and Caughey use alternating sweeping to solve linearized steady compressible Euler equation (2001).

The fast sweeping algorithm on rectangular grids

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

- Upwind difference at grid $x_{i,j}$ (**nonlinear equation**):
  
  $$
  [(u_{i,j} - u_{xmin})^+]^2 + [(u_{i,j} - u_{ymin})^+]^2 = h^2 c_{i,j}^2, \quad i, j = 1, 2, \ldots \tag{1}
  $$

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

- Initial guess: $u_{i,j} = u(x_{i,j}) = 0$, $x_{i,j} \in S$, $u_{i,j} = \infty$, $x_{i,j} \notin S$.

- G-S iterations with four alternating ordering:

  (1) $i = 1 : I, j = 1 : J$
  
  (2) $i = I : 1, j = 1 : J$
  
  (3) $i = I : 1, j = J : 1$
  
  (4) $i = 1 : I, j = J : 1$

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

$$
u_{i,j}^{new} = \min(u_{i,j}^{old}, \overline{u}),$$
Gauss-Seidel method for a linear elliptic problem

Solve \( \Delta u(x) = c(x), \quad u(x) = 0, \quad x \in S \)

- Discretization at grid \( x_{i,j} \) (linear equation), \( i, j = 1, 2, \ldots \)
  
  \[
  u_{i+1,j} - 2u_{i,j} + u_{i-1,j} + u_{i,j+1} - 2u_{i,j} + u_{i,j-1} = h^2c_{i,j}
  \]

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

- Gauss-Seidel iteration: for \( i = 1 : I, \quad j = 1 : J \)
  
  \[
  u_{i,j}^{n+1} = \frac{u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - h^2c_{i,j}}{4}
  \]

The iteration will converge. The number of iteration depends on the condition number of the matrix
The numerical Hamiltonian:

\[
H^h(p_-, p_+, q_-, q_+) = \sqrt{\max\{(p_-)^+, (p_+)^-\}^2 + \max\{(q_-)^+, (q_+)^-\}^2},
\]

where

\[
p_- = D_x u_{i,j} = u_{i,j} - u_{i-1,j}, \quad p_+ = D_x^+ u_{i,j} = u_{i+1,j} - u_{i,j},
\]
\[
q_- = D_y u_{i,j} = u_{i,j} - u_{i,j-1}, \quad q_+ = D_y^+ u_{i,j} = u_{i,j+1} - u_{i,j},
\]
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 \( \bar{x} \) to

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

which satisfies,

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

\( a_p < \bar{x} \leq a_{p+1} \) 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 \( \bar{x} = \tilde{x} \); 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 \( \bar{x} = \tilde{x} \), otherwise
3. Solve \( (x - a_1)^2 + (x - a_2)^2 + (x - a_3)^2 = c_{i,j}^2 h^2 \),

\[ \cdots, \]
untill we reach \( p \).
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 for distance function 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: how many sweeps are needed for the Gauss-Seidel iterations?

• Error estimate for the numerical solution?

• Fast sweeping method on unstructured mesh.

• High order schemes.

• More general Hamilton-Jacobi equations and hyperbolic conservation law.
The monotonicity and Lipschitz continuity of the scheme

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

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

we have

$$1 > \frac{\partial \bar{x}}{\partial a_1} \geq \frac{\partial \bar{x}}{\partial a_2} \geq \cdots \geq \frac{\partial \bar{x}}{\partial a_n} \geq 0,$$

$$\frac{\partial \bar{x}}{\partial a_1} + \frac{\partial \bar{x}}{\partial a_2} + \cdots + \frac{\partial \bar{x}}{\partial a_n} = 1,$$

and

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

**Theorem0:** (maximum change principle) The maximum change of $u^h$ at a grid point is no more than the maximum change of $u^h$ at its neighboring points in the G-S iterations.
Basic properties of the fast sweeping algorithm

**Lemma 1:** The fast sweeping algorithm is monotone in initial data.

**Lemma 2:** The numerical solution of the fast sweeping algorithm is monotonically decreasing with iterations.

**Lemma 3:** Let \( u^{(k)} \) and \( v^{(k)} \) be two numerical solutions at \( k \)-th sweep of the fast sweeping algorithm. Denote \( \| \cdot \|_\infty \) to be the maximum norm. We have

1. \( \|u^{(k)} - v^{(k)}\|_\infty \leq \|u^{(k-1)} - v^{(k-1)}\|_\infty \).
2. \( 0 \leq \max_{i,j}\{u^{(k)}_{i,j} - u^{(k+1)}_{i,j}\} \leq \max_{i,j}\{u^{(k-1)}_{i,j} - u^{(k-1)}_{i,j}\} \).
3. \( \|u^{(k)}_{i,j} - u^{(k)}_{i\pm1,j}\|_\infty \leq \|u^{(k-1)}_{i,j} - u^{(k-1)}_{i\pm1,j}\|_\infty \),
   \( \|u^{(k)}_{i,j} - u^{(k)}_{i,j\pm1}\|_\infty \leq \|u^{(k-1)}_{i,j} - u^{(k-1)}_{i,j\pm1}\|_\infty \).

Remark: (2) provides a stopping criterion. When \( \|u^{(k)} - u^{(k-1)}\|_\infty \) is \( O(h) \), the information has reached all points.
**Theorem 1:** The solution of the fast sweeping algorithm converges monotonically to the solution of the discretized system.

Proof:
(1) $0 \leq u_{i,j}^{k} \leq u_{i,j}^{k-1}$ $\Rightarrow$ $u_{i,j}^{k}$ converges $\forall i, j$.
(2) $0 \leq [(u_{i,j}^{(k)} - u_{x_{\text{min}}}^{(k)})^+]^2 + [(u_{i,j}^{(k)} - u_{y_{\text{min}}}^{(k)})^+]^2 - c_{i,j}^2 h^2$
$\leq C \max_{i,j} \{ u_{i,j}^{(k-1)} - u_{i,j}^{(k)} \}$.

Remark: If a grid point achieves the minimum value it can get at a certain iteration, it is the correct value and the value will not change in later iterations.
Main results for distance function

Let $S$ denote the data set, $u^h(x, S)$ denote the numerical solution, and $d(x, S)$ denote the exact distance function.

**Theorem 2:** For a single data point $S = \{x_0\}$, $u^h(x, x_0)$ converges after $2^n$ sweeps in $n$ dimensions and

$$d(x, x_0) \leq u^h(x, x_0) \leq d(x, x_0) + O(h|\log h|) \quad \text{(sharp!)}$$
For $S = \{x_m\}_{m=1}^M$, define $u^h(x, S) = \min[u^h(x, x_1), \ldots, u^h(x, x_M)]$

Since $d(x, S) = \min[d(x, x_1), d(x, x_2), \ldots, d(x, x_M)]$,

$$d(x, S) \leq u^h(x, S) \leq d(x, S) + O(|h \log h|).$$

**Lemma:** For a set of points $S = \{x_m\}_{m=1}^M$, $u^h(x, S) \leq u^h(x, S)$.

**Theorem 3:** For arbitrary $S$ in $n$ dimensions, after $2^n$ iterations.

$$\overline{u}(x_{i,j}, S) \leq u^h(x_{i,j}, S) \leq \overline{u}^h(x_{i,j}) \leq d(x_{i,j}, S) + O(|h \log h|),$$

where $\overline{u}(x_{i,j}, S)$ is the solution to the discretized system.
Denote $H(p, x) = |p| - c(x)$, where $p = \nabla u$.

The characteristic equations are:

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

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

\[
\begin{align*}
|\dot{x}| &= \left| \frac{\nabla u}{c(x)} \right| = 1, \\
\ddot{x} &= \frac{\nabla u}{c(x)} - \frac{\nabla u}{c} \cdot \frac{\nabla c}{c} \cdot \dot{x} = (I - P_n) \frac{\nabla c(x)}{c(x)}.
\end{align*}
\]

\(P_n\) is the projection on \(n = \frac{\nabla u}{|\nabla u|}\). \(\ddot{x}\) is the curvature.

Denote \(K = \max_{x \in \Omega} \frac{|\nabla c(x)|}{c(x)}\). The number of turns is bounded by

\[
\frac{1}{2\pi} \int_{\Gamma} |\ddot{x}| ds \leq \frac{1}{2\pi} \int_{\Gamma} \frac{|\nabla c(x)|}{c(x)} ds \leq \frac{K}{2\pi} \int_{\Gamma} ds \leq \frac{K}{2\pi} \text{length}(\Gamma)
\]
Let the characteristics $\Gamma$ joins $x_0 \in S$ and a $x \in \Omega$. 

$$c_m \int_\Gamma ds \leq \int_\Gamma c(s)ds = u(x) \leq \int_{x_0}^x c(s)ds \leq c_M \|x - x_0\|$$

Hence

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

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

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

**The solution:** Order all vertices according to the distance to a few reference points and sweep back and forth according to these orderings.
At a vertex $C$, compute $T(C)$ using its neighbors through each triangle and each edge then take the minimum one.

$$T^{new}(C) = \min\{T^{old}(C), T_{1}^{new}(C), T_{2}^{new}(C), \ldots\}$$
Local solver in one triangle

\[ T_C = \min_{s \in [0,1]} \left\{ sT_B + (1 - s)T_A + \frac{d(s)}{v_g(C; s)} \right\}. \]

\[ F = sA + (1 - s)B, \quad d(s) = AF, \]

\( v_g(C; s) \) is the group velocity in the direction of \( AF \).

Remark: For isotropic case,

- \( v_g(C; s) = v_g(C) \).

- if the triangle support a consistent and causality satisfying solution then \( T_C > \max\{T_A, T_B\} \).
Local solver in one triangle for Eikonal case

Given $T(A), T(B)$ compute $T(C)$.

$$c(T(B) - T(A)) = T(C) = c^*h + T(B)$$

$$T(C) = \text{min}\{T(A) + c^*AC, T(B) + c^*BC\}$$

$$T^{new}(C) = \text{min}\left\{ T^{old}(C), T(A) + \frac{AC}{v}, T(B) + \frac{BC}{v}, T(A, B) \right\}$$

Simple because the characteristic (ray) direction is the same as the travel time gradient (phase velocity).

$$\dot{x} = \nabla_p H = \frac{\nabla T}{c(x)}$$
Local solver for general convex Hamiltonian

**Difficulty:** the relation between ray direction and phase velocity.

**Our Solution:** separate the consistency condition from causality check.

Step 1: Find consistent $T_C$ from $T_A$ and $T_B$.

$$
\nabla T(C) \approx P^{-1} \left( \begin{pmatrix} \frac{T_C-T_A}{b} \\ \frac{T_C-T_B}{a} \end{pmatrix} \right),
$$

(2)

using linear approximation, where

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

(3)

Plug $\nabla T(C)$ into the PDE and solve for $T_C$

$$
H(C, \nabla T(C)) = 1, \quad \text{or} \quad \widehat{H}(C, T_C, T_A, T_B) = 1.
$$
Local solver (continued)

- No solution for $T_C \Rightarrow T_A, T_B$ do not support a consistent $T_C$ in this triangle.
- There are one or more than one solutions for $T_C \Rightarrow$ causality check.

Step 2: Check **Causality Condition**: the characteristic starting from $C$ against the direction $\nabla_p H(C, \nabla T(C))$ intersects the line segment $AB$.

- If no $T_C$ satisfies the causality condition $\Rightarrow T_A, T_B$ do not support a $T_C$ that satisfies both the consistency and causality in this triangle.
- If multiple $T_C$ satisfy both the consistency and causality in this triangle $\Rightarrow$ choose the smallest one using the first-arrival time principle.
Remark: If the Hamiltonian is convex $\Rightarrow$ there are at most two solutions for $T_C$.
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.
Monotonicity and contraction property of the scheme

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

$$\hat{H} \left( C, \frac{T_C - T_A}{b}, \frac{T_C - T_B}{a} \right) = H(C, p)$$

(4)

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

$$0 \leq \frac{\partial T_C}{\partial T_A}, \frac{\partial T_C}{\partial T_B} \leq 1, \quad \frac{\partial T_C}{\partial T_A} + \frac{\partial T_C}{\partial T_B} = 1$$

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

$$T_C = sT_B + (1 - s)T_A + \frac{d(s)}{v_g(C)}, \quad 0 < s < 1$$

then

$$0 < \frac{\partial T_C}{\partial T_A}, \frac{\partial T_C}{\partial T_B} < 1, \quad \frac{\partial T_C}{\partial T_A} + \frac{\partial T_C}{\partial T_B} = 1$$

If the true solution $T_C$ depends on $T_A$ and $T_B$ and the ordering is correct, each vertex update has a rate of contraction bounded away from 1.
Convergence

**Theorem** There exists a unique solution for the discretized non-linear system and the fast sweeping iteration converges.

For convex Hamiltonian, if \( \lim_{\lambda \to 0} \nabla_p H(x, \lambda p) = 0 \)

\[
\Rightarrow \frac{dT}{dt} = p \cdot \frac{dx}{dt} = p \cdot \nabla_p H \geq 0
\]

- The fast sweeping method works.
- The local solver is as difficult as solving the nonlinear Hamiltonian \( H(x, p) = 0 \).
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.
Parallel implementations of the fast sweeping method

- **Parallel sweeping**: do different sweeping orderings in parallel. Synchronization: for each point take the minimal value from each sweeping (causality of the solution).

- **Domain decomposition**: do fast sweeping in each subdomain either in additive or multiplicative version. No overlap is needed between subdomains. Communications between subdomains: Take the minimal value at the boundary between two subdomains, which allows only correct information passing among subdomains (causality of the solution).
Can an iterative method converge in finite iterations?

**Elliptic problem: no!**
Every point is coupled with all other points ⇒ the discretized system cannot 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 ⇒ 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 *a 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.
Remarks on the complexity of FSM

Let $M$ be the total number of nodes, the complexity of FSM:

- on rectangular grids: $O(M)$;
- on unstructured meshes: $O(M \log M)$.

**Remark 1:** On unstructured meshes the log $M$ factor comes from the initial sorting of all nodes. Once the sorting is done the complexity of the fast sweeping iterations is $O(M)$.

**Remark 2:** The constant in the complexity formula does not depend on the anisotropy of the Hamiltonian.
Anisotropic eikonal equations

\[
[\nabla T(x) M(x) \nabla T(x)]^{\frac{1}{2}} = 1, \quad x \in \mathbb{R}^d, \tag{5}
\]

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

\[
H = \sqrt{a(x) \ p_1^2 - 2c(x) \ p_1 \ p_2 + b(x) \ p_2^2} = 1, \tag{6}
\]

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

\[
M = \begin{pmatrix}
a, & -c \\
-c, & b
\end{pmatrix}
\]

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

\[
\eta = \sqrt{\frac{\lambda_{\text{max}}(M)}{\lambda_{\text{min}}(M)}},
\]

where \( \lambda_{\text{max}}(M) \) and \( \lambda_{\text{min}}(M) \) 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.
<table>
<thead>
<tr>
<th>Nodes</th>
<th>Elements</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iter</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1473</td>
<td>2816</td>
<td>8.78E-3</td>
<td>–</td>
<td>4</td>
<td>8.87E-3</td>
<td>–</td>
<td>4</td>
</tr>
<tr>
<td>5716</td>
<td>11264</td>
<td>4.04E-3</td>
<td>1.12</td>
<td>4</td>
<td>5.38E-3</td>
<td>0.72</td>
<td>4</td>
</tr>
<tr>
<td>22785</td>
<td>45056</td>
<td>2.10E-3</td>
<td>0.94</td>
<td>4</td>
<td>3.22E-3</td>
<td>0.74</td>
<td>5</td>
</tr>
<tr>
<td>90625</td>
<td>180224</td>
<td>1.04E-3</td>
<td>1.02</td>
<td>4</td>
<td>1.88E-3</td>
<td>0.77</td>
<td>5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Elements</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iter</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1473</td>
<td>2816</td>
<td>6.23E-3</td>
<td>–</td>
<td>4</td>
<td>5.72E-3</td>
<td>–</td>
<td>4</td>
</tr>
<tr>
<td>5716</td>
<td>11264</td>
<td>2.89E-3</td>
<td>1.11</td>
<td>4</td>
<td>3.33E-3</td>
<td>0.78</td>
<td>4</td>
</tr>
<tr>
<td>22785</td>
<td>45056</td>
<td>1.53E-3</td>
<td>0.92</td>
<td>4</td>
<td>1.93E-3</td>
<td>0.79</td>
<td>5</td>
</tr>
<tr>
<td>90625</td>
<td>180224</td>
<td>7.66E-4</td>
<td>1.00</td>
<td>4</td>
<td>1.10E-3</td>
<td>0.80</td>
<td>5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Mesh</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iter</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 x 40</td>
<td>1.17E-1</td>
<td>–</td>
<td>4</td>
<td>1.57E-2</td>
<td>–</td>
<td>4</td>
</tr>
<tr>
<td>80 x 80</td>
<td>6.35E-2</td>
<td>0.88</td>
<td>4</td>
<td>8.18E-3</td>
<td>0.94</td>
<td>4</td>
</tr>
<tr>
<td>160 x 160</td>
<td>3.39E-2</td>
<td>0.90</td>
<td>4</td>
<td>4.18E-3</td>
<td>0.97</td>
<td>4</td>
</tr>
<tr>
<td>320 x 320</td>
<td>1.78E-2</td>
<td>0.93</td>
<td>4</td>
<td>2.12E-3</td>
<td>0.98</td>
<td>4</td>
</tr>
</tbody>
</table>

Comparison between the five point stencils and the nine point stencils. $a = 1, b = 1, c = 0.9$. 
Reconstruction of functions (images) from gradient

Reconstruct $u$ from $|\nabla u|$ and local minima or maxima (boundary conditions) by solving the Eikonal equation, e.g., compression of Digital Terrain Elevation Data (DTED).

**Example:**

Given an image (the intensity function) $u_{i,j}$,

**step 1:** Record locations and values of local minima. $u_{i,j}$ is a local minimum if $u_{i,j} \leq \min\{u_{i\pm1,j}, u_{i,j\pm1}\}$.

**step 2:** Extract the gradient.

Define

$$u_x = \begin{cases} \frac{u_{i,j}-u_{x\text{min}}}{h} & \text{if } u_{i,j} > u_{x\text{min}} \\ 0 & \text{else} \end{cases}$$

$$u_y = \begin{cases} \frac{u_{i,j}-u_{y\text{min}}}{h} & \text{if } u_{i,j} > u_{y\text{min}} \\ 0 & \text{else} \end{cases}$$

$$f = \sqrt{u_x^2 + u_y^2}$$
Remarks:

- The gradient extraction is the exact inverse procedure of solving the Eikonal equation.

- Gradient space has many nice properties for geometric features. Many processing operations can be done in the gradient space.

- The reconstruction is an integral operator and is true multidimensional.

- Higher order of gradients can be used.

- Extraction of gradient and local minimum (or saddle) points needs care.
The equation for the graph of surface $u(x, y)$ is

$$\vec{v} \cdot \vec{n} = I(x, y),$$

$\vec{v}$ is the direction of incoming light, $\vec{n} = (-u_x, -u_y, 1)/\sqrt{u_x^2 + u_y^2 + 1}$ is the unit surface normal, and $I(x, y)$ is the reflected light intensity. Take $\vec{v} = (0, 0, 1)$ we get the H-J equation for $u(x, y)$:

$$\sqrt{u_x^2 + u_y^2 + 1} = \frac{1}{I(x, y)}$$
Examples

$\Omega = [0, 1] \times [0, 1]$, $u(x, y) = 0$ at $\partial \Omega$.

$I(x, y) = 1/\sqrt{1 + (2\pi \cos(2\pi x) \sin(2\pi y))^2 + (2\pi \sin(2\pi x) \cos(2\pi y))^2}$,

(a) $u(\frac{1}{4}, \frac{1}{4}) = u(\frac{3}{4}, \frac{3}{4}) = 1$, $u(\frac{1}{4}, \frac{3}{4}) = u(\frac{3}{4}, \frac{1}{4}) = -1$, $u(\frac{1}{2}, \frac{1}{2}) = 0$

(b) $u(\frac{1}{4}, \frac{1}{4}) = u(\frac{3}{4}, \frac{3}{4}) = u(\frac{1}{4}, \frac{3}{4}) = u(\frac{3}{4}, \frac{1}{4}) = 1$, $u(\frac{1}{2}, \frac{1}{2}) = 2$
**First order method on rectangular grids**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( h = )</td>
<td>0.01</td>
<td>0.005</td>
<td>0.0025</td>
</tr>
<tr>
<td></td>
<td>( | e |_\infty )</td>
<td>0.0524</td>
<td>0.0254</td>
<td>0.0125</td>
</tr>
<tr>
<td></td>
<td>( | e |_{L^2} )</td>
<td>0.0311</td>
<td>0.0158</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>( | e |_{L^1} )</td>
<td>0.0266</td>
<td>0.0137</td>
<td>0.0069</td>
</tr>
<tr>
<td># of sweeps</td>
<td></td>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

**case (b)**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( h = )</td>
<td>0.01</td>
<td>0.005</td>
<td>0.0025</td>
</tr>
<tr>
<td></td>
<td>( | e |_\infty )</td>
<td>0.0388</td>
<td>0.0202</td>
<td>0.0105</td>
</tr>
<tr>
<td></td>
<td>( | e |_{L^2} )</td>
<td>0.0184</td>
<td>0.01</td>
<td>0.0054</td>
</tr>
<tr>
<td></td>
<td>( | e |_{L^1} )</td>
<td>0.014</td>
<td>0.076</td>
<td>0.0041</td>
</tr>
<tr>
<td># of sweeps</td>
<td></td>
<td>7</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
First order method on triangular mesh

Order of convergence

<table>
<thead>
<tr>
<th>Nodes</th>
<th>dist. to 2 circles</th>
<th>SFS(a)</th>
<th>SFS(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L^1$ error</td>
<td>order</td>
<td>$L^1$ error</td>
</tr>
<tr>
<td>1473</td>
<td>7.71E-3</td>
<td>–</td>
<td>4.54E-2</td>
</tr>
<tr>
<td>5716</td>
<td>4.21E-3</td>
<td>0.87</td>
<td>2.54E-2</td>
</tr>
<tr>
<td>22785</td>
<td>2.18E-3</td>
<td>0.95</td>
<td>1.34E-2</td>
</tr>
<tr>
<td>90625</td>
<td>1.11E-3</td>
<td>0.97</td>
<td>6.90E-3</td>
</tr>
</tbody>
</table>

Treatment for abrupt angle

<table>
<thead>
<tr>
<th>Ele.</th>
<th>Obt. ele.</th>
<th>$L^1$ error</th>
<th>order</th>
<th>ite</th>
<th>$L^1$ error</th>
<th>order</th>
<th>ite</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>78</td>
<td>6.54E-2</td>
<td>–</td>
<td>4</td>
<td>4.20E-2</td>
<td>–</td>
<td>4</td>
</tr>
<tr>
<td>800</td>
<td>528</td>
<td>2.46E-2</td>
<td>1.41</td>
<td>8</td>
<td>1.68E-2</td>
<td>1.32</td>
<td>5</td>
</tr>
<tr>
<td>3200</td>
<td>958</td>
<td>2.36E-2</td>
<td>0.06</td>
<td>8</td>
<td>7.91E-3</td>
<td>1.09</td>
<td>8</td>
</tr>
<tr>
<td>12800</td>
<td>5890</td>
<td>1.52E-2</td>
<td>0.63</td>
<td>9</td>
<td>4.09E-3</td>
<td>0.95</td>
<td>12</td>
</tr>
<tr>
<td>51200</td>
<td>40558</td>
<td>4.46E-3</td>
<td>1.77</td>
<td>16</td>
<td>2.19E-3</td>
<td>0.90</td>
<td>11</td>
</tr>
</tbody>
</table>
Number of iterations for different ordering

Spherical ordering using $l_2$ distance $\sqrt{(x - x_{ref})^2 + (y - y_{ref})^2}$

<table>
<thead>
<tr>
<th>Nodes</th>
<th>dist. to 2 circles</th>
<th>SFS (a)</th>
<th>SFS (b)</th>
<th>five-rings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1473</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>19</td>
</tr>
<tr>
<td>5716</td>
<td>6</td>
<td>13</td>
<td>13</td>
<td>20</td>
</tr>
<tr>
<td>22785</td>
<td>8</td>
<td>11</td>
<td>13</td>
<td>21</td>
</tr>
<tr>
<td>90625</td>
<td>8</td>
<td>11</td>
<td>13</td>
<td>21</td>
</tr>
</tbody>
</table>

Planar ordering using $l_1$ distance $|x - x_{ref}| + |y - y_{ref}|$

<table>
<thead>
<tr>
<th>Nodes</th>
<th>two-circles</th>
<th>SFS (a)</th>
<th>SFS (b)</th>
<th>five-rings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1473</td>
<td>9</td>
<td>12</td>
<td>9</td>
<td>26</td>
</tr>
<tr>
<td>5716</td>
<td>9</td>
<td>12</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>22785</td>
<td>9</td>
<td>16</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>90625</td>
<td>9</td>
<td>15</td>
<td>9</td>
<td>27</td>
</tr>
</tbody>
</table>
3D computation on unstructured mesh

80 x 80 x 80 mesh
3rd-order Godunov
fast-sweeping

80 x 80 x 80 mesh
3rd-order Godunov
fast-sweeping

Interior of left picture

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Elements</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>9261</td>
<td>48000</td>
<td>1.25E-2</td>
<td>–</td>
<td>12 × 16</td>
</tr>
<tr>
<td>68921</td>
<td>384000</td>
<td>7.17E-3</td>
<td>0.81</td>
<td>12 × 16</td>
</tr>
<tr>
<td>531441</td>
<td>3072000</td>
<td>3.79E-3</td>
<td>0.92</td>
<td>12 × 16</td>
</tr>
</tbody>
</table>

Tetrahedron mesh. Spherical sweeping front with 8 reference points
High order methods: using Godunov flux

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

$$
\left[\left(\frac{\phi_{i,j}^{\text{new}} - \phi_{i,j}^{(x\text{min})}}{h}\right)^+\right]^2 + \left[\left(\frac{\phi_{i,j}^{\text{new}} - \phi_{i,j}^{(y\text{min})}}{h}\right)^+\right]^2 = f_{i,j}^2
$$

where

$$
\begin{align*}
\phi_{i,j}^{(x\text{min})} &= \min(\phi_{i,j}^{\text{old}} - h \cdot (\phi_x)_{i,j}^-, \phi_{i,j}^{\text{old}} + h \cdot (\phi_x)_{i,j}^+), \\
\phi_{i,j}^{(y\text{min})} &= \min(\phi_{i,j}^{\text{old}} - h \cdot (\phi_y)_{i,j}^-, \phi_{i,j}^{\text{old}} + h \cdot (\phi_y)_{i,j}^+).
\end{align*}
$$

$\phi_{i,j} \pm h \cdot (\phi_x)_{i,j}^\pm$ 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

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

where \( \alpha x = \max_{A \leq u \leq B, C \leq v \leq D} |H_1(u, v)| \) and \( \alpha y = \max_{A \leq u \leq B, C \leq v \leq D} |H_2(u, v)| \).

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

\[ \phi_{i,j}^{\text{new}} = \left( \frac{1}{\alpha x h_x + \alpha y h_y} \right) \left[ f - H\left( \frac{\phi_{i+1,j}^- - \phi_{i-1,j}^+}{2h_x}, \frac{\phi_{i,j+1}^- - \phi_{i,j-1}^+}{2h_y} \right) \right. \\
+ \alpha x \frac{\phi_{i+1,j}^+ + \phi_{i-1,j}^-}{2h_x} + \alpha y \frac{\phi_{i,j+1}^+ + \phi_{i,j-1}^-}{2h_y} \]

replace \( \phi_{i\pm 1,j} \) and \( \phi_{i,j\pm 1} \) 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:

\[ \phi_{i,j}^{\text{new}} = \left( \frac{1}{\alpha x h_x + \alpha y h_y} \right) \left[ f - H\left( \frac{(\phi_x)_{i,j}^- + (\phi_x)_{i,j}^+}{2}, \frac{(\phi_y)_{i,j}^- + (\phi_y)_{i,j}^+}{2} \right) \right. \\
+ \alpha x \frac{(\phi_x)_{i,j}^+ - (\phi_x)_{i,j}^-}{2} + \alpha y \frac{(\phi_y)_{i,j}^+ - (\phi_y)_{i,j}^-}{2} \] + \phi_{i,j}^{\text{old}}. \]
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.

<table>
<thead>
<tr>
<th>mesh</th>
<th>smooth region</th>
<th>whole region</th>
<th>rarefaction</th>
<th>iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>80 × 80</td>
<td>1.03E-5</td>
<td>-</td>
<td>1.70E-4</td>
<td>-</td>
</tr>
<tr>
<td>160×160</td>
<td>1.32E-6</td>
<td>2.96</td>
<td>6.09E-5</td>
<td>1.48</td>
</tr>
<tr>
<td>320 × 320</td>
<td>2.89E-7</td>
<td>2.19</td>
<td>1.72E-5</td>
<td>1.82</td>
</tr>
<tr>
<td>640 × 640</td>
<td>4.03E-8</td>
<td>2.84</td>
<td>4.13E-6</td>
<td>2.06</td>
</tr>
</tbody>
</table>

exact solution

numerical solution, mesh: 80×80
High order methods

3rd order Godunov method

<table>
<thead>
<tr>
<th>mesh</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iteration #</th>
<th>mesh</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iteration #</th>
</tr>
</thead>
<tbody>
<tr>
<td>80 × 80</td>
<td>1.24E-4</td>
<td>–</td>
<td>16x4</td>
<td>1.14E-3</td>
<td>–</td>
<td>18x4</td>
<td></td>
</tr>
<tr>
<td>160 × 160</td>
<td>5.78E-6</td>
<td>4.43</td>
<td>23x4</td>
<td>1.98E-4</td>
<td>2.53</td>
<td>23x4</td>
<td></td>
</tr>
<tr>
<td>320 × 320</td>
<td>3.27E-7</td>
<td>4.14</td>
<td>29x4</td>
<td>2.95E-5</td>
<td>2.74</td>
<td>31x4</td>
<td></td>
</tr>
<tr>
<td>640 × 640</td>
<td>3.56E-8</td>
<td>3.20</td>
<td>52x4</td>
<td>7.28E-6</td>
<td>2.02</td>
<td>57x4</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>mesh</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iteration #</th>
<th>mesh</th>
<th>$L^1$ error</th>
<th>order</th>
<th>iteration #</th>
</tr>
</thead>
<tbody>
<tr>
<td>80 × 80</td>
<td>1.44E-4</td>
<td>–</td>
<td>22x4</td>
<td>2.94E-3</td>
<td>–</td>
<td>28x4</td>
<td></td>
</tr>
<tr>
<td>160 × 160</td>
<td>7.68E-6</td>
<td>4.23</td>
<td>33x4</td>
<td>8.35E-4</td>
<td>1.81</td>
<td>38x4</td>
<td></td>
</tr>
<tr>
<td>320 × 320</td>
<td>4.93E-7</td>
<td>3.96</td>
<td>58x4</td>
<td>2.20E-4</td>
<td>1.92</td>
<td>75x4</td>
<td></td>
</tr>
<tr>
<td>640 × 640</td>
<td>5.58E-8</td>
<td>3.14</td>
<td>111x4</td>
<td>4.67E-5</td>
<td>2.24</td>
<td>136x4</td>
<td></td>
</tr>
</tbody>
</table>

Comparision between time-marching and fast sweeping with 3rd order WENO.
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
<table>
<thead>
<tr>
<th>mesh</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
<th>iteration #</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 × 40</td>
<td>7.23E-4</td>
<td></td>
<td>3.07E-3</td>
<td></td>
<td>27x4</td>
</tr>
<tr>
<td>80 × 80</td>
<td>9.33E-5</td>
<td>2.95</td>
<td>5.81E-4</td>
<td>2.40</td>
<td>36x4</td>
</tr>
<tr>
<td>160 × 160</td>
<td>9.58E-6</td>
<td>3.28</td>
<td>6.72E-5</td>
<td>3.11</td>
<td>58x4</td>
</tr>
<tr>
<td>320 × 320</td>
<td>1.27E-6</td>
<td>2.92</td>
<td>8.94E-6</td>
<td>2.91</td>
<td>98x4</td>
</tr>
<tr>
<td>640 × 640</td>
<td>1.62E-7</td>
<td>2.97</td>
<td>1.16E-6</td>
<td>2.95</td>
<td>178x4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>mesh</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
<th>iteration #</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 × 40</td>
<td>9.20E-4</td>
<td></td>
<td>3.65E-3</td>
<td></td>
<td>42x4</td>
</tr>
<tr>
<td>80 × 80</td>
<td>6.21E-5</td>
<td>3.89</td>
<td>6.32E-4</td>
<td>2.53</td>
<td>34x4</td>
</tr>
<tr>
<td>160 × 160</td>
<td>2.39E-6</td>
<td>4.70</td>
<td>2.28E-5</td>
<td>4.79</td>
<td>57x4</td>
</tr>
<tr>
<td>320 × 320</td>
<td>4.61E-7</td>
<td>2.37</td>
<td>1.47E-6</td>
<td>3.96</td>
<td>99x4</td>
</tr>
<tr>
<td>640 × 640</td>
<td>5.97E-8</td>
<td>2.95</td>
<td>2.97E-7</td>
<td>2.30</td>
<td>181x4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>mesh</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
<th>iteration #</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 × 40</td>
<td>2.19E-3</td>
<td></td>
<td>1.53E-2</td>
<td></td>
<td>42x4</td>
</tr>
<tr>
<td>80 × 80</td>
<td>6.09E-4</td>
<td>1.84</td>
<td>8.02E-3</td>
<td>0.93</td>
<td>34x4</td>
</tr>
<tr>
<td>160 × 160</td>
<td>1.62E-4</td>
<td>1.91</td>
<td>4.19E-3</td>
<td>0.94</td>
<td>57x4</td>
</tr>
<tr>
<td>320 × 320</td>
<td>3.70E-5</td>
<td>2.13</td>
<td>2.00E-3</td>
<td>1.06</td>
<td>99x4</td>
</tr>
<tr>
<td>640 × 640</td>
<td>1.02E-5</td>
<td>1.86</td>
<td>8.21E-4</td>
<td>1.29</td>
<td>181x4</td>
</tr>
</tbody>
</table>
travel time for quasi-P wave (left) and quasi-SV wave (right)
Fast sweeping method for steady state scalar hyperbolic conservation laws (joint with Chen, Shu, Zhang)

- Motivation: Information propagates along characteristics starting from inflow boundary until shocks form.

- Main difficulties: causality rule and upwind difference schemes are more complicated.
Numerical procedure

For a particular ordering march from the boundary inward as far as possible, e.g., at a grid point update its value using an upwind scheme corresponding to this sweeping direction unless

- either upwind causality is violated: the grid value updated from the upwind scheme using the sweeping ordering is against the causality.
- or shock is detected: current values of the point and its neighbors satisfies both the conservation law and the entropy condition for a shock.
Discretization

1D case: \( f(u)_x = h(u, x) \)

The discretization is

\[
\frac{\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}}{\Delta x} = h(u_j, x_j)
\]

using Roe flux:

\[
\hat{f}_{j+\frac{1}{2}} = \begin{cases} 
  f(u_j), & \frac{f(u_{j+1})-f(u_j)}{u_{j+1}-u_j} \geq 0; \\
  f(u_{j+1}), & \frac{f(u_{j+1})-f(u_j)}{u_{j+1}-u_j} < 0.
\end{cases}
\]

2D case: \( f(u)_x + g(u)_y = h(u, x, y) \) is treated similarly.

High resolution method such as the discontinuous Galerkin (DG) method can also be used in this formulation.
Example 1: \[ \begin{cases} \frac{1}{2} u^2_x = a(x) u, & 0 \leq x \leq 1; \\ u(0) = 1, u(1) = -0.1. \end{cases} \]

where \( a(x) = 6x - 3. \) Exact solution for this problem is

\[ u(x) = \begin{cases} 3x^2 - 3x + 1, & 0 \leq x < x^*; \\ 3x^2 - 3x - 0.1, & x^* < x \leq 1. \end{cases} \]

\( x^* = \frac{1 - \sqrt{0.4}}{2} \approx 0.18 \) is the shock location.

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
N & 1^{st} \text{ order} & L_1 \text{ error} & \text{order} & \text{DG} P^1 & L_1 \text{ error} & \text{order} \\
\hline
20 & & 1.02E-01 & - & & 4.57E-4 & - \\
80 & & 2.66E-02 & 0.971 & & 3.05E-5 & 1.954 \\
320 & & 6.62E-03 & 1.003 & & 1.94E-6 & 1.986 \\
\hline
\end{array}
\]

two sweeps are needed
Example 2. \( yu_x - xu_y = 0 \), \( (x, y) \in [-1, 1] \times [-1, 1] \).

The exact solution is \( u(x, y) = -\sqrt{x^2 + y^2} \).

<table>
<thead>
<tr>
<th>( N )</th>
<th>( 1^{st} ) order</th>
<th>( L_1 ) error</th>
<th>order</th>
<th>DG ( P^1 )</th>
<th>( L_1 ) error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>8.21E-02</td>
<td>-</td>
<td></td>
<td>6.46E-3</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>4.56E-02</td>
<td>0.848</td>
<td></td>
<td>1.64E-3</td>
<td>1.981</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>2.43E-02</td>
<td>0.909</td>
<td></td>
<td>4.12E-4</td>
<td>1.990</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>1.26E-02</td>
<td>0.945</td>
<td></td>
<td>1.03E-4</td>
<td>1.994</td>
<td></td>
</tr>
</tbody>
</table>

four sweeps are needed

left: 1st order upwind; right: DG \( P^1 \)
Example 3. $u_y + \left(\frac{u^2}{2}\right)_x = 0$, $(x, y) \in [0, 1] \times [0, 1]$. $u(x, y) = \begin{cases} 1.5, & x = 0, \\ -2.5x + 1.5, & y = 0, \\ -1.0, & x = 1. \end{cases}$

Exact solution is

$$u(x, y) = \begin{cases} u_L, & (x, y) \text{ between lines } g_1 \text{ and } g_3, \\ u_R, & (x, y) \text{ between lines } g_2 \text{ and } g_3, \\ (u_R - u_L)(x_p + \frac{y_p(x-x_p)}{y_p-y}) + u_L, & (x, y) \text{ between lines } g_1 \text{ and } g_2, \end{cases}$$

where $u_L = 1.5, u_R = -1.0, x_p = u_L/(u_L - u_R), y_p = 1/(u_L - u_R)$; the leftmost char. $g_1$ is $y = x/u_L$, the rightmost char. $g_2$ is $y = (x-1)/u_R$, the shock $g_3$ is $y = (2x-1)/(u_L + u_R)$.

Two sweeps are needed.

Contour plot of the solution. Solid line: exact; Dashed line: numerical
Example 4. \( u_y + \left( \frac{u^2}{2} \right)_x = 0 \), \((x, y) \in [0, 1] \times [0, 1]\). \( u(x, y) = \begin{cases} -0.5, & x \leq 0.6, y = 0, \\ 0.4, & x > 0.6, y = 0. \end{cases} \)

Exact solution is

\[
u(x, y) = \begin{cases} -0.5, & \frac{x-0.6}{y-0.4} \leq -0.5, \\ \frac{x-0.6}{y-0.4}, & -0.5 < \frac{x-0.6}{y-0.4} < 0.4, \\ 0.4, & \frac{x-0.6}{y-0.4} \geq 0.4. \end{cases}
\]

Fixed Roe and special treatment is needed for rarefaction.

Two sweeps are needed.

Contour plot of the solution. Solid line: exact; Dashed line: numerical
Example 5. $(\frac{u^2}{2\sqrt{2}})_x + (\frac{u^2}{2\sqrt{2}})_y = -\pi \cos(\pi \frac{x+y}{\sqrt{2}})u$, $(x, y) \in [0, \frac{1}{\sqrt{2}}] \times [0, \frac{1}{\sqrt{2}}]$. Where $x_s = 0.14876$, and the exact solution is

$$u(x, y) = \begin{cases} 
1 - \sin(\pi \frac{x+y}{\sqrt{2}}), & \text{if } \frac{x+y}{\sqrt{2}} \leq x_s, \\
-0.1 - \sin(\pi \frac{x+y}{\sqrt{2}}), & \text{if } \frac{x+y}{\sqrt{2}} > x_s.
\end{cases}$$

Four sweeps are needed.

Contour plot of the solution. Solid line: exact; Dashed line: numerical
Conclusion and current work

- Conclusion: FS method is a GS type of iterative scheme. Properly designed, it can work efficiently for hyperbolic problems.

- Current work:
  - Hyperbolic conservation laws, in particular for systems.
  - Convection dominated diffusion problem.
  - Righthand (the velocity) is discontinuous across an interface. The solution satisfies a nonlinear jump condition (Snell’s law) across the interface.

\[
[u] = 0, \quad [u_n^2] = [c^2]
\]

How to enforce the jump conditions to get high order accuracy across the interface?