A Fast Marching Scheme
For The
Linearized Eikonal Equation

Jonathan Franklin And Jerry Harris

The Seismic Tomography Project
Stanford University
Introduction

1. Motivation And Overview

2. Example A: 2nd Order Fast-Marching Solver For The Linearized Eikonal Equation.

3. Example B: 2nd Order Schemes For Take-Off Angle/Arc Length Calculation

4. Numerical Difficulties: Solutions And Questions

5. Conclusions And Future Work
## Motivations ... And ... Contributions

<table>
<thead>
<tr>
<th>Interest in the linearized eikonal equation and a general class of advection-like PDE’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. used for cheap updates of traveltimes in iterative tomography.</td>
</tr>
<tr>
<td>b. original plan involved using iterative applications of a linearized solver to compute multi-valued traveltime maps.</td>
</tr>
<tr>
<td>c. amplitude and pulse-width calculation for attenuation tomography</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1. Development of an efficient FD linearized eikonal solver without aperture limitations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Experimentation with higher order FD stencils within the fast-marching framework.</td>
</tr>
<tr>
<td>3. Extensions to solve the ray-&gt;cartesian coordinate mapping problem</td>
</tr>
<tr>
<td>a. Take-Off angle</td>
</tr>
<tr>
<td>b. Arc length integration</td>
</tr>
<tr>
<td>c. Geometric spreading estimates</td>
</tr>
</tbody>
</table>
Deriving The Linearized Eikonal Equation

\[ [S(x, z)]^2 = \left( \frac{\partial t}{\partial x} \right)^2 + \left( \frac{\partial t}{\partial z} \right)^2 \]

We begin with the Eikonal Equation

We then perturb both the slowness and traveltime fields

\[ S(x, z) = S_o(x, z) + \delta S(x, z) \]

Substituting the perturbed and original fields ...

\[ [S_o + \delta S]^2 = \left( \frac{\partial t_o}{\partial x} + \frac{\partial \tau}{\partial x} \right)^2 + \left( \frac{\partial t_o}{\partial z} + \frac{\partial \tau}{\partial z} \right)^2 \]

Expansion

\[ S_o^2 + 2S_o\delta S + \delta S^2 = \left( \frac{\partial t_o}{\partial x} \right)^2 + \left( \frac{\partial t_o}{\partial z} \right)^2 + \left( \frac{\partial \tau}{\partial x} \right)^2 + \left( \frac{\partial \tau}{\partial z} \right)^2 + 2 \left( \frac{\partial t_o}{\partial x} \frac{\partial \tau}{\partial x} \right) + 2 \left( \frac{\partial t_o}{\partial z} \frac{\partial \tau}{\partial z} \right) \]

And after dropping higher order terms and dividing by 2 ....
The Linearized Eikonal Equation

\[
\left( \frac{\partial t_0}{\partial x} \right) \left( \frac{\partial \tau}{\partial x} \right) + \left( \frac{\partial t_0}{\partial z} \right) \left( \frac{\partial \tau}{\partial z} \right) = S_o(x, z) \delta S(x, z)
\]
## A Useful Class Of PDE’s

\[
\left( \frac{\partial t}{\partial x} \right) \left( \frac{\partial u}{\partial x} \right) + \left( \frac{\partial t}{\partial z} \right) \left( \frac{\partial u}{\partial z} \right) = R(x, z)
\]

Corresponds to conserving or integrating some quantity over the characteristic curves defined by the time field

\[
\left( \frac{\partial t}{\partial x} \right) \left( \frac{\partial \theta}{\partial x} \right) + \left( \frac{\partial t}{\partial z} \right) \left( \frac{\partial \theta}{\partial z} \right) = 0
\]

Conservation of Take-Off angle

\[
\left( \frac{\partial t}{\partial x} \right) \left( \frac{\partial s}{\partial x} \right) + \left( \frac{\partial t}{\partial z} \right) \left( \frac{\partial s}{\partial z} \right) = \frac{1}{v}
\]

Arc-Length Integration

**Linearized Eikonal Equation**

\[
\left( \frac{\partial t_o}{\partial x} \right) \left( \frac{\partial \tau}{\partial x} \right) + \left( \frac{\partial t_o}{\partial z} \right) \left( \frac{\partial \tau}{\partial z} \right) = S_o(x, z)\delta S(x, z)
\]
The Two Sides Of FD Traveltime Methods

<table>
<thead>
<tr>
<th>Micro Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>A local method for calculating derivatives and updating traveltimes: Should produce accurate, smooth local extrapolations.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Macro Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>A global method for ordering the evaluation of the finite-difference operators – ideally the macro scheme insure causality.</td>
</tr>
</tbody>
</table>
### Marching Schemes: Choices

#### Static Or Quasi–Static Marching Schemes

- **Expanding Box**
  - (Vidale ’88)
- **Down’n’Out**
  - (Dellinger + Symes 97’)
  - (Kim + Cook 98’)
- **Depth Stepping**
  - (Reshef + Kosloff 86’)
  - (El Mageed 96’)

#### Dynamic Marching  Expanding Wavefronts

- Expanding in order of minimum time: mimics wavefront and guarantees causal application of FD operators.
  - (Qin et.al. 92’)
  - (Cao + Greenhalgh 94’)
  - (Sethian 96’)
  - (Popovici + Sethian 97’)

---

**Qin et.al. 92’**

**Cao + Greenhalgh 94’**

**Popovici + Sethian 97’**
Fast Marching : Macro Algorithm

The causality (upwind ordering) of the finite-difference operators is preserved by updating minimum time nodes within the narrow-band.

(Qin et.al. 92, Sethian ’96)

- Already Visited
- Untouched
- Narrow Band
- Minimum Time
- Next Stencil Evaluation

(computational grid)
A Simple Expression

\[
\left( \frac{\partial t_o}{\partial x} \right) \left( \frac{\partial \tau}{\partial x} \right) + \left( \frac{\partial t_o}{\partial z} \right) \left( \frac{\partial \tau}{\partial z} \right) = S_o(x, z) \delta S(x, z)
\]

\[
\frac{\partial \tau}{\partial x} = \frac{\tau_{i+1,j} - \tau_{i,j}}{\Delta x} \quad h = \Delta x = \Delta z \quad \frac{\partial \tau}{\partial z} = \frac{\tau_{i,j+1} - \tau_{i,j}}{\Delta z}
\]

\[
\tau_{i,j} = \frac{\left( \frac{\partial t_o}{\partial x} \right) \tau_{i+1,j} + \left( \frac{\partial t_o}{\partial x} \right) \tau_{i,j+1} - h S_o \delta S}{\left( \frac{\partial t_o}{\partial x} \right) + \left( \frac{\partial t_o}{\partial x} \right)}
\]
## Upwind Difference Approximations

<table>
<thead>
<tr>
<th>Order</th>
<th>( \frac{\partial \tau}{\partial x} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>( \frac{\tau_{i+1} - \tau_i}{\Delta x} )</td>
</tr>
<tr>
<td>2nd</td>
<td>( \frac{-\tau_{i+2} + 4\tau_{i+1} - 3\tau_i}{2\Delta x} )</td>
</tr>
<tr>
<td>3rd</td>
<td>( \frac{2\tau_{i+3} - 9\tau_{i+2} + 18\tau_{i+1} - 11\tau_i}{6\Delta x} )</td>
</tr>
<tr>
<td>4th</td>
<td>( \frac{-\tau_{i+4} + 6\tau_{i+3} + 18\tau_{i+2} + 10\tau_{i+1} - 33\tau_i}{60\Delta x} )</td>
</tr>
</tbody>
</table>

Substitution of appropriate stencils into the linearized eikonal equation and solution for \( \tau_{i,j} \) yields an explicit update formula.
A Useful Explicit Form

Upwind difference operators of arbitrary order can be expressed as...

\[
\frac{\partial \tau}{\partial x} = \frac{1}{g_x \Delta x} \sum_{i=0}^{n} c_i \tau_{i,j}
\]

\[
\frac{\partial \tau}{\partial z} = \frac{1}{g_z \Delta z} \sum_{j=0}^{n} c_j \tau_{i,j}
\]

Compressing the summations as P’s and equalizing cell dimensions ...

\[
P_x = \sum_{i=0}^{n} c_i \tau_{i,j}
\]

\[
P_z = \sum_{j=0}^{n} c_j \tau_{i,j}
\]

\[
h = \Delta x = \Delta z
\]

Yielding an explicit extrapolation formula for the linearized eikonal equation for arbitrary order upwind difference systems

\[
\tau_{i,j} = \frac{\left[ h g_x g_z S_0 S_1 \right] - \left( g_z \right) \left( \frac{\partial t_\phi}{\partial x} \right) \left( P_x \right) - \left( g_x \right) \left( \frac{\partial t_\phi}{\partial z} \right) \left( P_z \right) \left( g_z \right) \left( c_{x0} \right) \left( \frac{\partial t_\phi}{\partial x} \right) + \left( g_x \right) \left( c_{z0} \right) \left( \frac{\partial t_\phi}{\partial z} \right)}{\left( g_z \right) \left( c_{x0} \right) \left( \frac{\partial t_\phi}{\partial x} \right) + \left( g_x \right) \left( c_{z0} \right) \left( \frac{\partial t_\phi}{\partial z} \right)}
\]
A general FD formula for arbitrary order upwind stencils

A PDE For Take-Off Angle Calculation

\[ \left( \frac{\partial t}{\partial x} \right) \left( \frac{\partial \theta}{\partial x} \right) + \left( \frac{\partial t}{\partial z} \right) \left( \frac{\partial \theta}{\partial z} \right) = 0 \]

\[ \nabla t \cdot \nabla \theta = 0 \]

Along any given characteristic, take-off angle is constant (Zhang 93’)

Dot product form

A general FD formula for arbitrary order upwind stencils

\[ \theta_{i,j} = \frac{(g_x)(P_z)(\frac{\partial t}{\partial z}) - (g_z)(P_x)(\frac{\partial t}{\partial x})}{(g_z)(c_x0)(\frac{\partial t}{\partial x}) + (g_x)(c_z0)(\frac{\partial t}{\partial z})} \]

\[ s_{i,j} = \frac{[h(g_x g_z)S] - (g_z)(\frac{\partial t}{\partial z})(P_{xr}) - (g_x)(\frac{\partial t}{\partial x})(P_{zr})}{(g_z)(c_x0)(\frac{\partial t}{\partial x}) + (g_x)(c_z0)(\frac{\partial t}{\partial z})} \]
And Amplitudes?

<table>
<thead>
<tr>
<th>Amplitude Transport Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nabla t \cdot \nabla A + \frac{1}{2} (\nabla^2 t) A = 0$</td>
</tr>
</tbody>
</table>

| $A(x, z) = R(\theta) \sqrt{\frac{v_o}{|J(x, z)| v(x, z)}}$ |

Unfortunately, difficulties with J due to inaccuracies in take-off angle derivatives.
### Method In A Nutshell

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Sort traveltime data in increasing order (or use previously determined ordering)</td>
</tr>
<tr>
<td>2.</td>
<td>Initialize values near source</td>
</tr>
<tr>
<td>3.</td>
<td>Evaluate Finite–Difference Operators in the order determined by the sorted traveltimes</td>
</tr>
<tr>
<td>3a.</td>
<td>At each evaluation, choose an appropriate upwind stencil</td>
</tr>
</tbody>
</table>

**Initial Data:** Traveltime Table, $T_1$
1. Search For Nearby Points in the narrow band.
2. Select Highest-Order Applicable Stencil
3. Evaluate Background Gradient
4. Evaluate new Perturbed Time

Load Data
Xdim, Zdim, Scale
Background Slowness
Background Traveltimes
Slowness Anomaly Field
Maximum Stencil Order

Sort Background Times
Explicit QuickSort
Best-of-three Medians
Randomization

Dynamic Gradient Evaluation
1st-4th Order Stencils

Loop Over Grid Locations In Order Of Increasing Time

Output Results
Traveltime Perturbations
Gradient Fields
## Implementation Details

1. Coded in modular C++ with polymorphic storage constructs (template based).

2. Optimized Quicksort exploits
   - a. Best-Of-3 Median Picks
   - b. Explicit swaps
   - c. Stack Formulated (no recursion)
   - d. Low-level randomization phase

3. Written in general form to allow quick adaptation to any PDE expressable as ....

\[
\frac{\partial t}{\partial x} \left( \frac{\partial u}{\partial x} \right) + \left( \frac{\partial t}{\partial z} \right) \left( \frac{\partial u}{\partial z} \right) = R(x, z)
\]
Analytic Solution: A Single Perturbed Layer

\[
\| K_1 \| = \frac{d_1}{\cos(\tan^{-1} \frac{\pi}{2})}
\]
\[
\| K_2 \| = \frac{d_1 + d_2}{\cos(\tan^{-1} \frac{\pi}{2})}
\]
\[
\| K_r \| = \frac{z_r}{\cos(\tan^{-1} \frac{\pi}{2})}
\]

if \( z_r < d_1 \), \( \int \delta S \, dr = 0 \)
if \( z_r \geq d_1 \) ...
\[
\int \delta S \, dr = \delta S [\| K_r \| - \| K_1 \|]
\]
if \( z_r > (d_1 + d_2) \) ...
\[
\int \delta S \, dr = \delta S [\| K_2 \| - \| K_1 \|]
\]
Simple Salt: Arc-Length Angle + Traveltimes (2nd Order FM)

0.74
0.862
0.985
1.11
1.23
1.35
1.47

500 1000 1500 2000 2500 3000 3500

1500
2000
2500
3000
Performance Aspects

If the sorting operation is included, both linear and non-linear fast-marching algorithms are $O(n \lg n)$. However, a linearized solver has several small wins from a performance standpoint.

1. Might not require a traveltime sort.

2. Quick Sort has a low constant.

3. The FD stencil does not have a sqrt operation.
Performance Comparison: NL vs L (500x500 model)

- Non-Linear (1st Order)
- Linear (2nd Order) With Sort
- Linear (1st Order) With Sort
- Linear (1st Order)
- Linear (2nd Order)
## Numerical Difficulties

1. All schemes initially exhibited only 1st order convergence.

   Solution: Careful treatment of near–source problem via local tracing and LUMR.

2. Both 3rd and 4th order schemes proved to be unstable. Cause?

3. Instabilities in calculating geometric spreading and amplitudes.

   Possible culprits: initial traveltimes only 2nd order accurate i.e. 0th order spreading estimates.
Dealing With Near-Source Numerics

The Problem:

Time field is non-differentiable at source: introduces first-order error into higher-order schemes unless explicitly dealt with.

Solutions:

<table>
<thead>
<tr>
<th>Ray Trace Near-Source Region (Folklore)</th>
<th>L.U.M.R. (Locally Uniform Mesh Refinement)</th>
<th>Adaptive Grid Strategies</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Kim + Cook, 98’)</td>
<td>static</td>
<td>dynamic</td>
</tr>
<tr>
<td></td>
<td>(Belfi + Symes, 98’)</td>
<td>(Belfi + Symes, 98’)</td>
</tr>
</tbody>
</table>
Conclusions

1. We have developed an efficient finite-difference scheme for solving a useful class of PDE’s including
   a. Linearized Eikonal Equation
   b. Take-Off Angle PDE
   c. Arc Length Calculation

2. As part of the above methods, we have extended traditional Fast-Marching techniques to 2nd and higher orders of accuracy.
## Future Work

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Perfection of amplitude calculation</td>
</tr>
<tr>
<td>2</td>
<td>Stabilize higher-order FD schemes</td>
</tr>
<tr>
<td>3</td>
<td>Develop a scheme for calculating pulse broadening in attenuating media</td>
</tr>
<tr>
<td>4</td>
<td>Continue the quest for multivalued FD Traveltimes</td>
</tr>
</tbody>
</table>
Many Thanks To ...

<table>
<thead>
<tr>
<th>Guan Wang (STP)</th>
<th></th>
<th>Sergei Fomel (SEP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill Symes (TRIP)</td>
<td></td>
<td>Jianliang Qian (TRIP)</td>
</tr>
</tbody>
</table>

STP = Seismic Tomography Project
SEP = Stanford Exploration Project
TRIP = The Rice Inversion Project