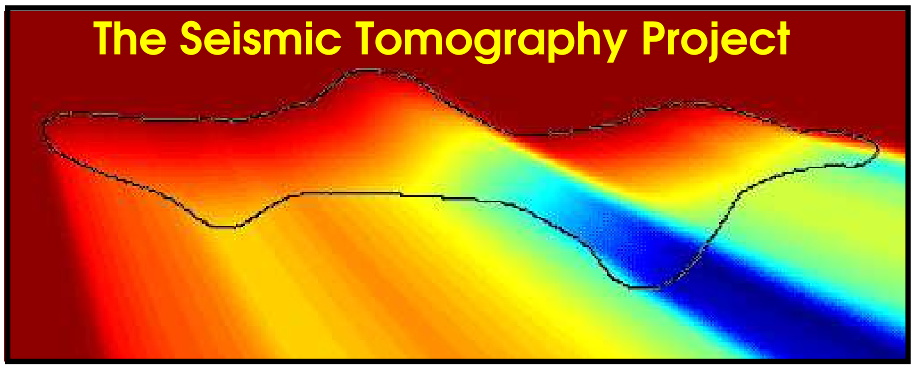
A Fast Marching Scheme For The Linearized Eikonal Equation

Jonathan Franklin And Jerry Harris



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

Interest in the linearized eikonal equation and a general class of advection−like PDE's

a. used for cheap updates of traveltimes in iterative tomography.

b. original plan involved using iterative applications of a linearized solver to compute multi−valued traveltime maps.

c. amplitude and pulse−width calculation for attenuation tomography

1. Development of an efficient FD linearized eikonal solver without aperture limitations.

2. Experimentation with higher order FD stencils within the fast−marching framework.

3. Extensions to solve the ray−>cartesian coordinate mapping problem
   a. Take−Off angle
   b. Arc length integration
   c. Geometric spreading estimates

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

$$S(x,z) = S_o(x,z) + \delta S(x,z)$$

$$t = t_o + \tau$$

Substituting the perturbed and original fields ...

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

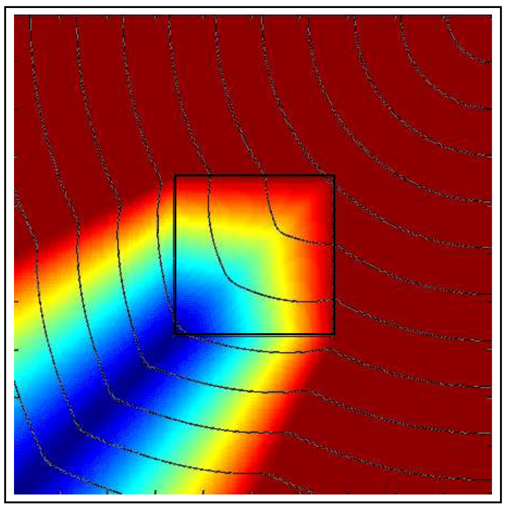
Expansion

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

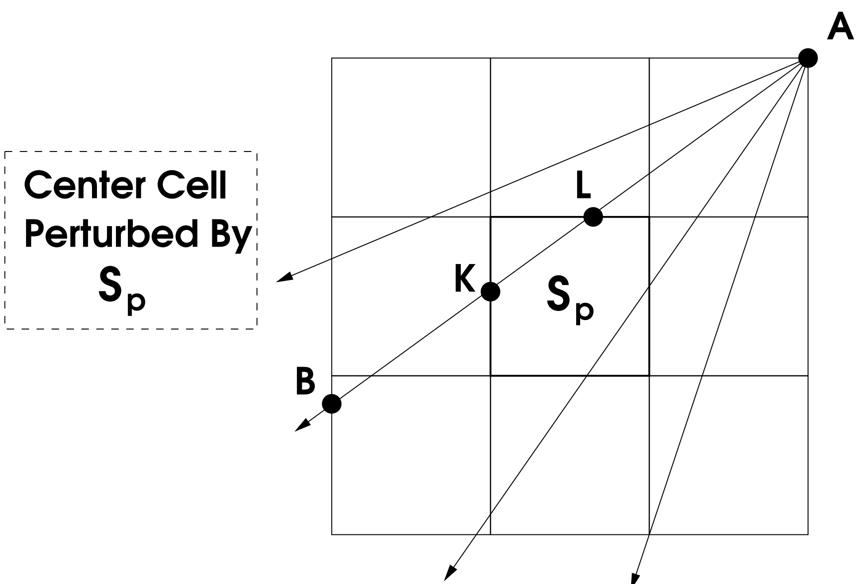
And after dropping higher order terms and dividing by 2 ....

The Linearized Eikonal Equation

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



Colors: Traveltime Perturbation
Contours: $t_o+\tau$



Characteristics (Rays) For A Constant Slowness Background

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

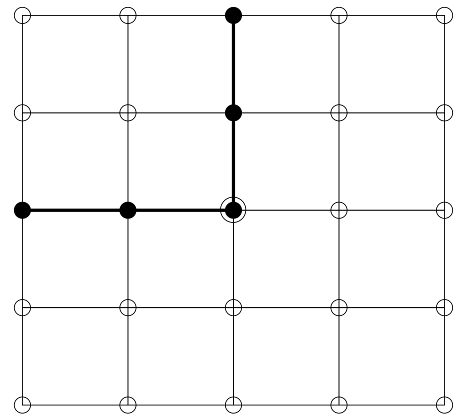
Linearized Eikonal Equation

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

The Two Sides Of FD Traveltime Methods

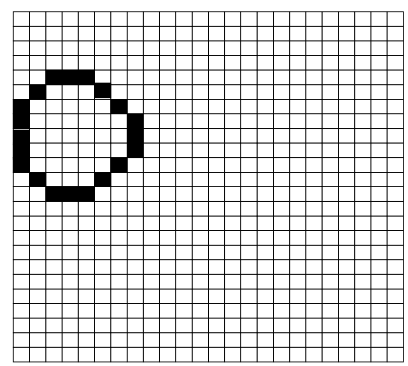
Micro Scheme

A local method for calculating derivatives and updating traveltimes: Shoulde produce accurate, smooth local extrapolations.



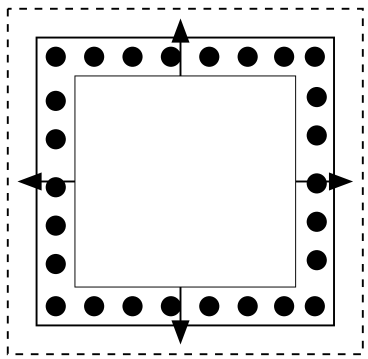
Macro Scheme

A global method for ordering the evaluation of the finite−difference operators − ideally the macro scheme insures causality.



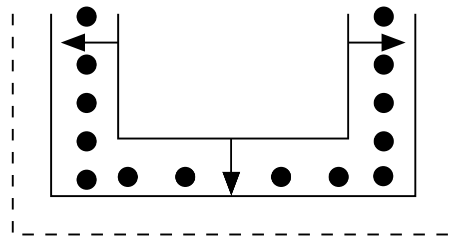
Marching Schemes: Choices

Static Or Quasi–Static Marching Schemes



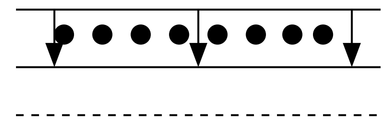
Expanding Box

(Vidale ’88)



Down’n’Out

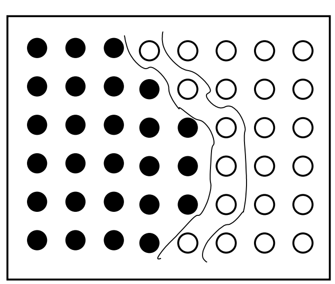
(Dellinger + Symes 97’)
(Kim + Cook 98’)



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.

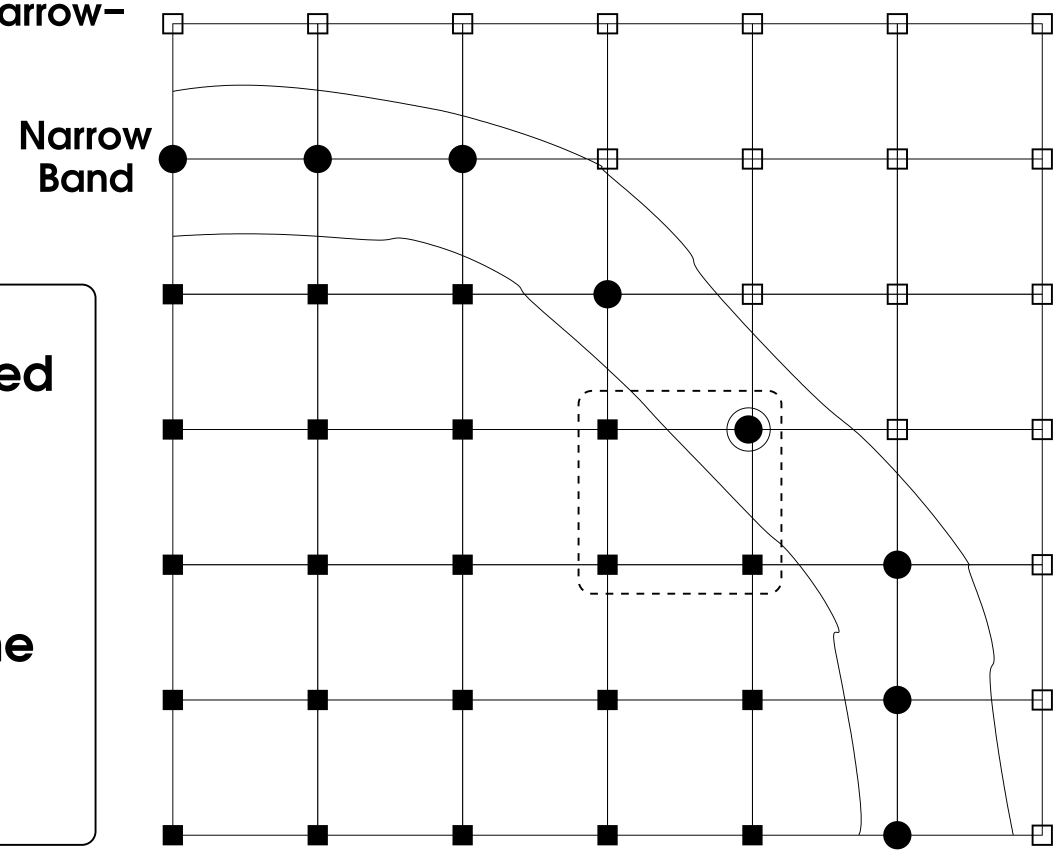
(Qin et.al. 92’)
(Cao + Greenhalgh 94’)
(Sethian 96’)
(Popovici + Sethian 97’)

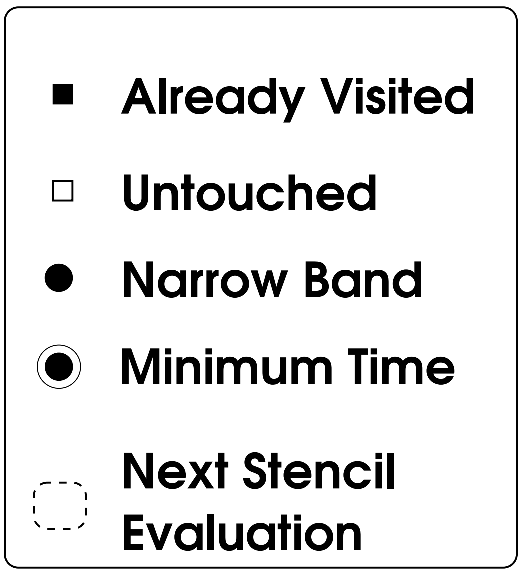
Fast Marching : Macro Algorithm

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)

(computational grid)

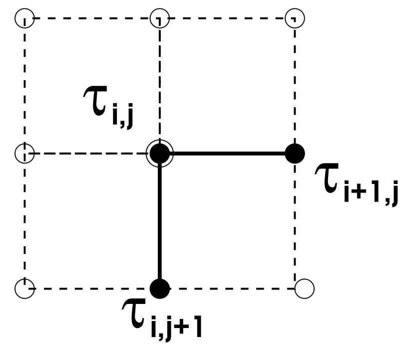




A Simple Expression

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

$$\frac{\partial \tau}{\partial x}=\frac{\tau_{i+1,j}-\tau_{i,j}}{\Delta x} \qquad h=\Delta x=\Delta z \qquad \frac{\partial \tau}{\partial z}=\frac{\tau_{i,j+1}-\tau_{i,j}}{\Delta z}$$



$$\boxed{\tau_{i,j}=\frac{\left(\frac{\partial t_o}{\partial x}\right)\tau_{i+1,j}+\left(\frac{\partial t_o}{\partial x}\right)\tau_{i,j+1}-h\,S_o\,\delta S}{\left(\frac{\partial t_o}{\partial x}\right)+\left(\frac{\partial t_o}{\partial x}\right)}}$$

Upwind Difference Approximations

Order

1st
$$\frac{\partial \tau}{\partial x} = \frac{\tau_{i+1} - \tau_i}{\Delta x}$$

2nd
$$\frac{\partial \tau}{\partial x} = \frac{-\tau_{i+2} + 4\tau_{i+1} - 3\tau_i}{2\Delta x}$$

3rd
$$\frac{\partial \tau}{\partial x} = \frac{2\tau_{i+3} - 9\tau_{i+2} + 18\tau_{i+1} - 11\tau_i}{6\Delta x}$$

4th
$$\frac{\partial \tau}{\partial x} = \frac{-\tau_{i+4} + 6\tau_{i+3} + 18\tau_{i+2} + 10\tau_{i+1} - 33\tau_i}{60\Delta x}$$

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

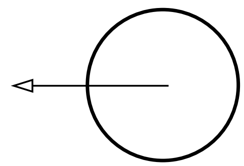
$$\tau_{i,j} = \frac{[h\, g_x\, g_z\, S_0\, S_1] - (g_z)(\frac{\partial t_o}{\partial x})(P_x) - (g_x)(\frac{\partial t_o}{\partial z})(P_z)}{(g_z)(c_{x0})(\frac{\partial t_o}{\partial x}) + (g_x)(c_{z0})(\frac{\partial t_o}{\partial z})}$$

A PDE For Take−Off Angle Calculation

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

Along any given characteristic, take−off angle is constant (Zhang 93')

$$\nabla t \cdot \nabla \theta = 0$$



Dot product form

$$\theta_{i,j} = \frac{(g_x)(P_z)(\frac{\partial t}{\partial z}) - (g_z)(P_x)(\frac{\partial t}{\partial x})}{(g_z)(c_{x0})(\frac{\partial t}{\partial x}) + (g_x)(c_{z0})(\frac{\partial t}{\partial z})}$$

A general FD formula for arbitrary order upwind stencils

$$s_{i,j} = \frac{[h(g_x\, g_z)S] - (g_z)(\frac{\partial t}{\partial x})(P_{xr}) - (g_x)(\frac{\partial t}{\partial z})(P_{zr})}{(g_z)(c_{x0})(\frac{\partial t}{\partial x}) + (g_x)(c_{z0})(\frac{\partial t}{\partial z})}$$

And Amplitudes ?

$$\nabla t \cdot \nabla A + \frac{1}{2}(\nabla^2 t)A = 0$$

Amplitude Transport Equation

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

Amplitudes in terms of source radiation pattern, velocity, and geometric spreading

Unfortunately, difficulties with J due to inaccuracies in take–off angle derivatives

Method In A Nutshell

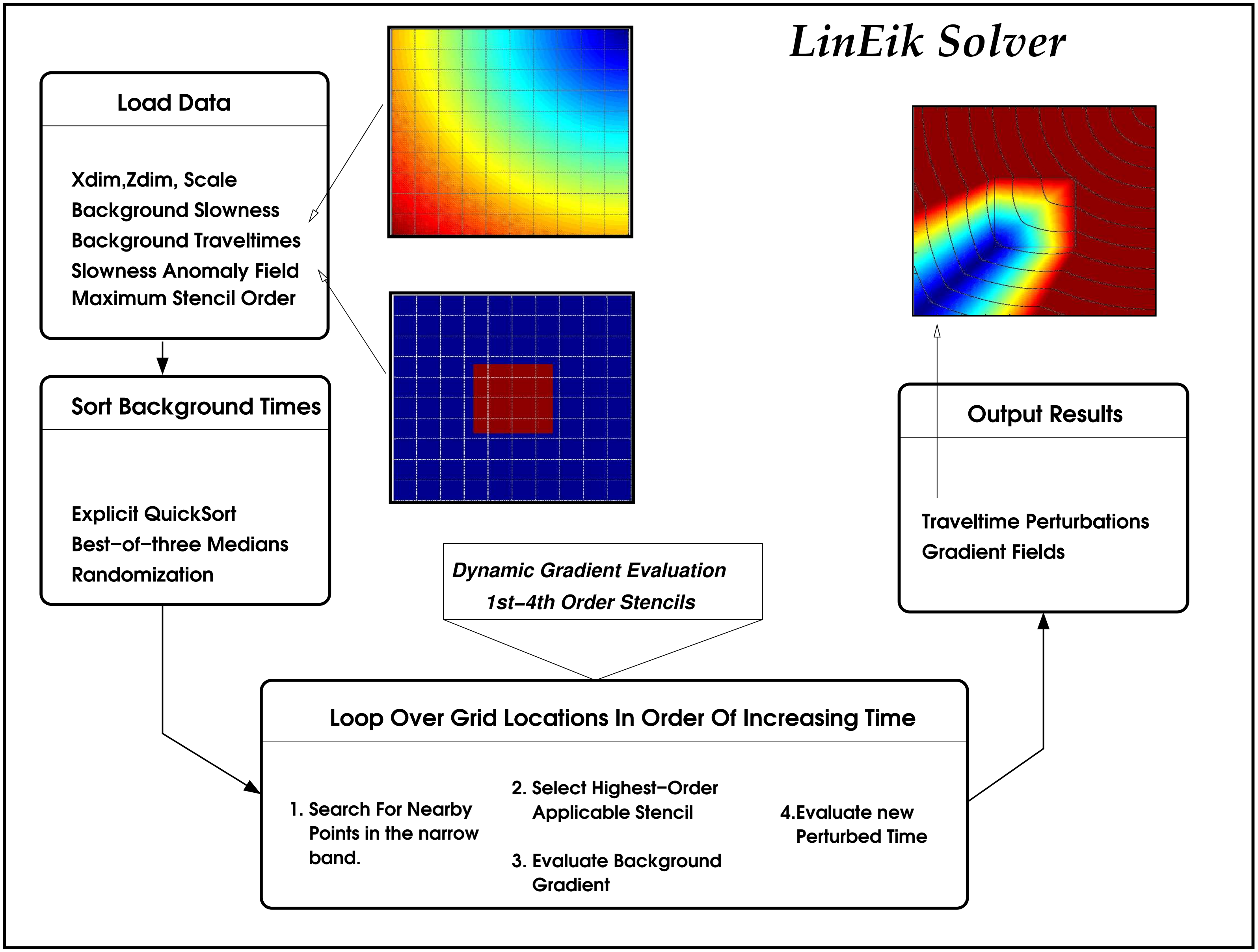
Initial Data : Traveltime Table, T_1

1. Sort traveltime data in increasing order
   (or use previously determined ordering)

2. Initialize values near source

3. Evaluate Finite−Difference Operators in the order
   determined by the sorted traveltimes

   3a. At each evaluation, choose an appropriate
   upwind stencil


LinEik Solver
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
Output Results
Traveltime Perturbations
Gradient Fields
Loop Over Grid Locations In Order Of Increasing Time
1. Search For Nearby Points in the narrow band.
2. Select Highest–Order Applicable Stencil
3. Evaluate Background Gradient
4.Evaluate new Perturbed Time

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 ....

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

Analytic Solution: A Single Perturbed Layer

$$||K_1|| = \frac{d_1}{cos(tan^{-1}\frac{x}{z})}$$

$$||K_2|| = \frac{d_1 + d_2}{cos(tan^{-1}\frac{x}{z})}$$

$$||K_r|| = \frac{z_r}{cos(tan^{-1}\frac{x}{z})}$$

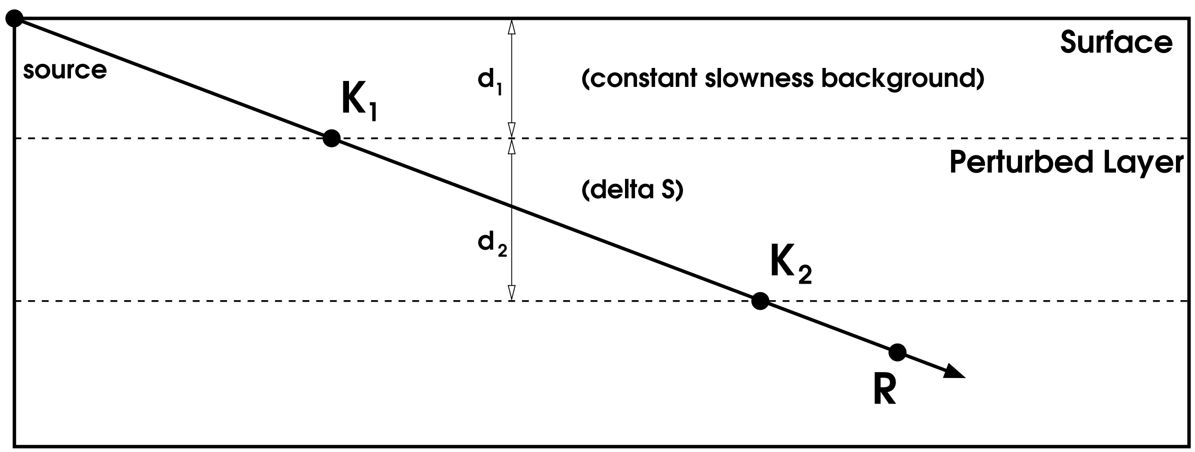
if $z_r < d_1$, $\int \delta S \, dr = 0$

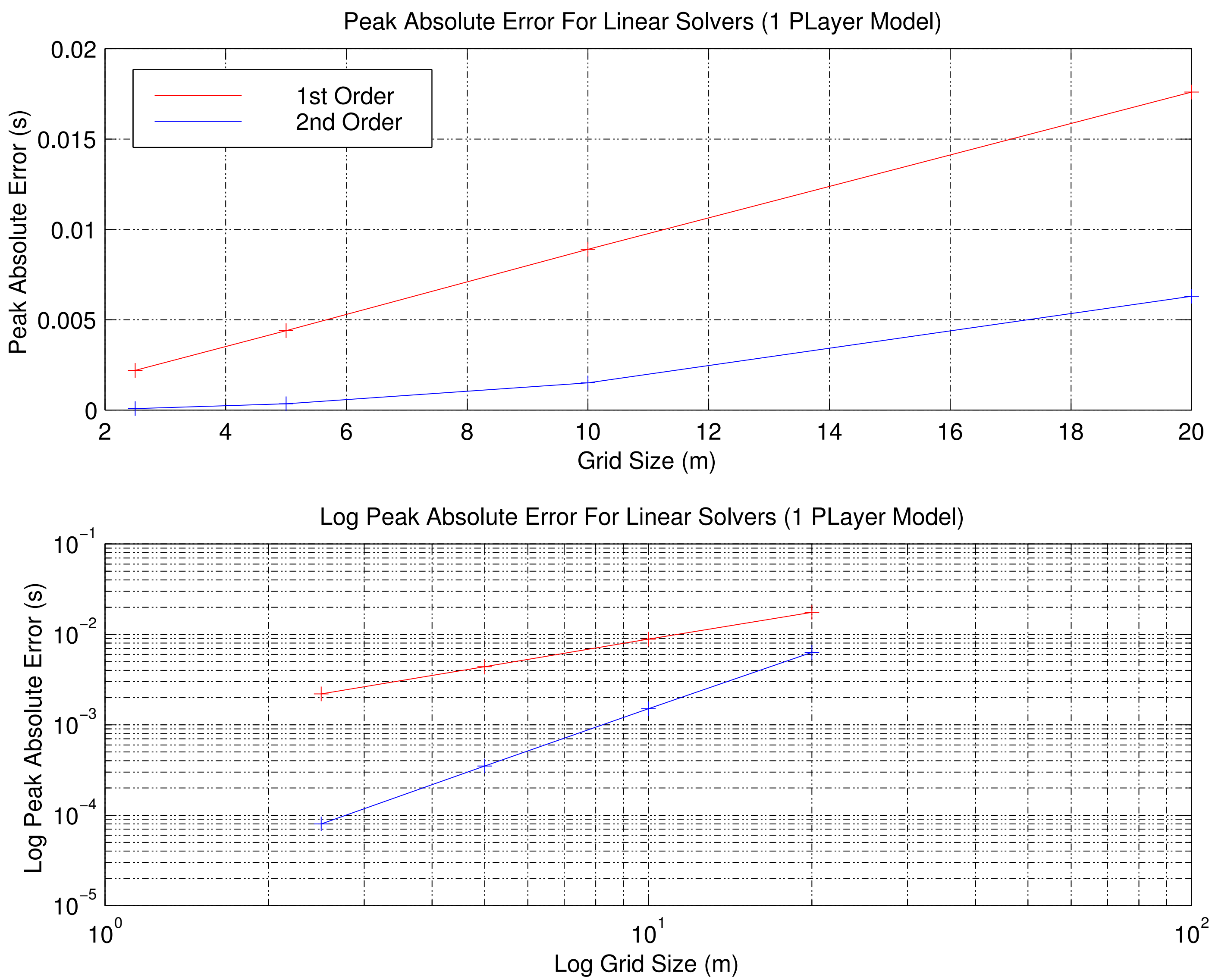
if $z_r \geq d_1$ ...

$$\int \delta S \, dr = \delta S[||K_r|| - ||K_1||]$$

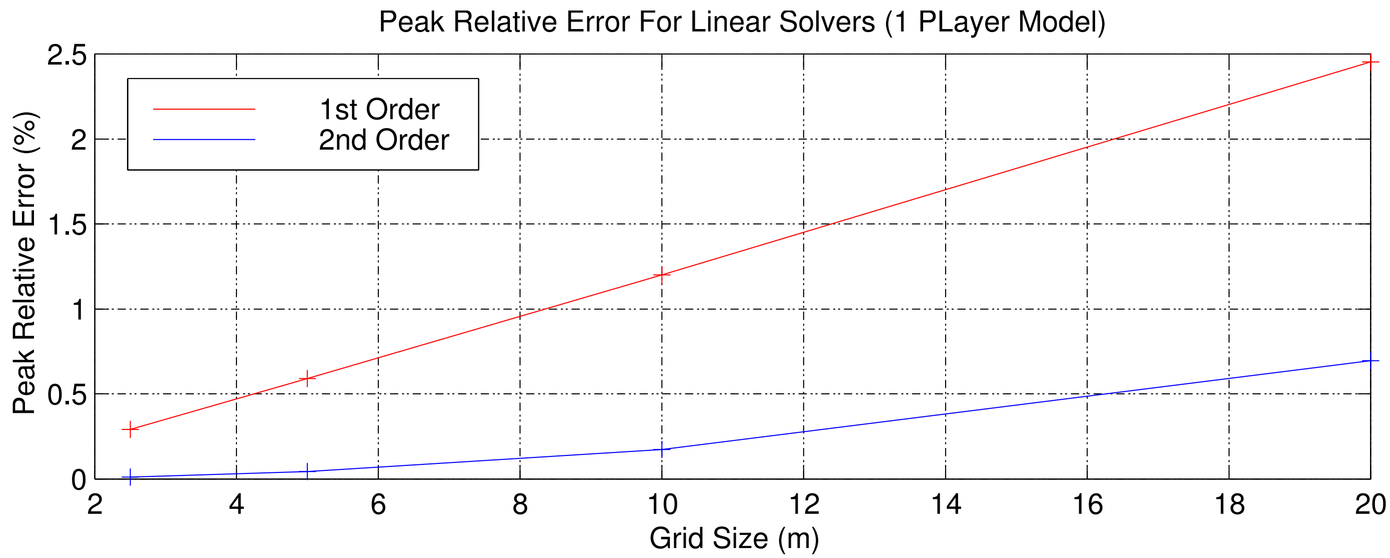
if $z_r > (d_1 + d_2)$ ...

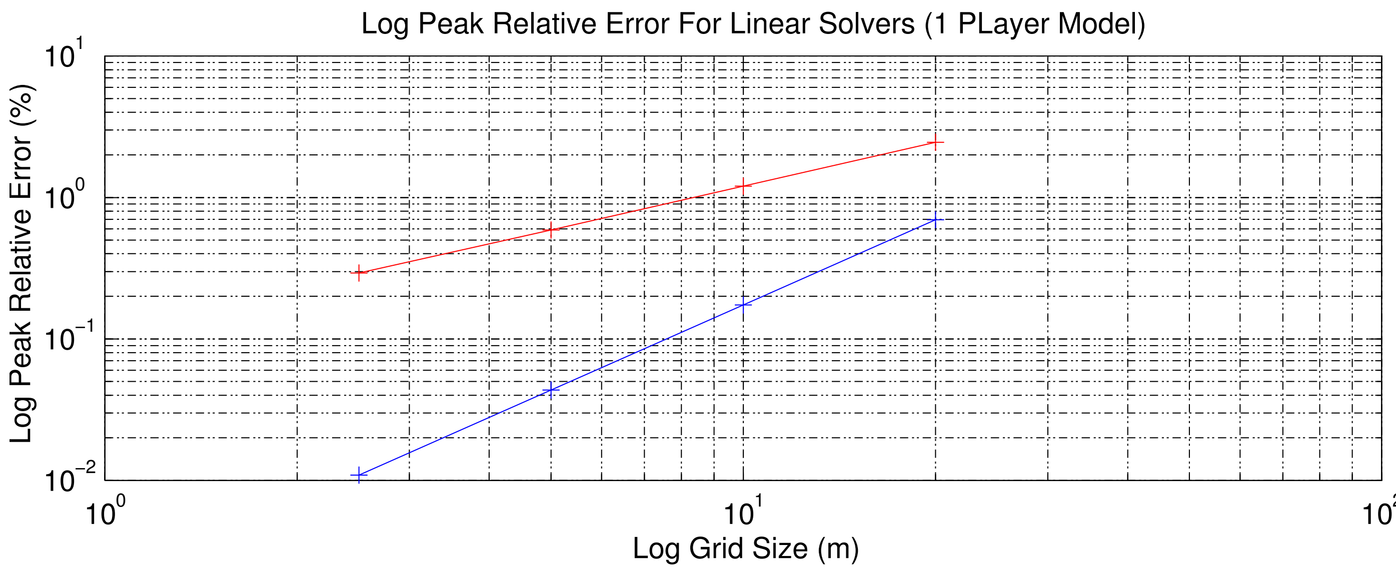
$$\int \delta S \, dr = \delta S[||K_2|| - ||K_1||]$$

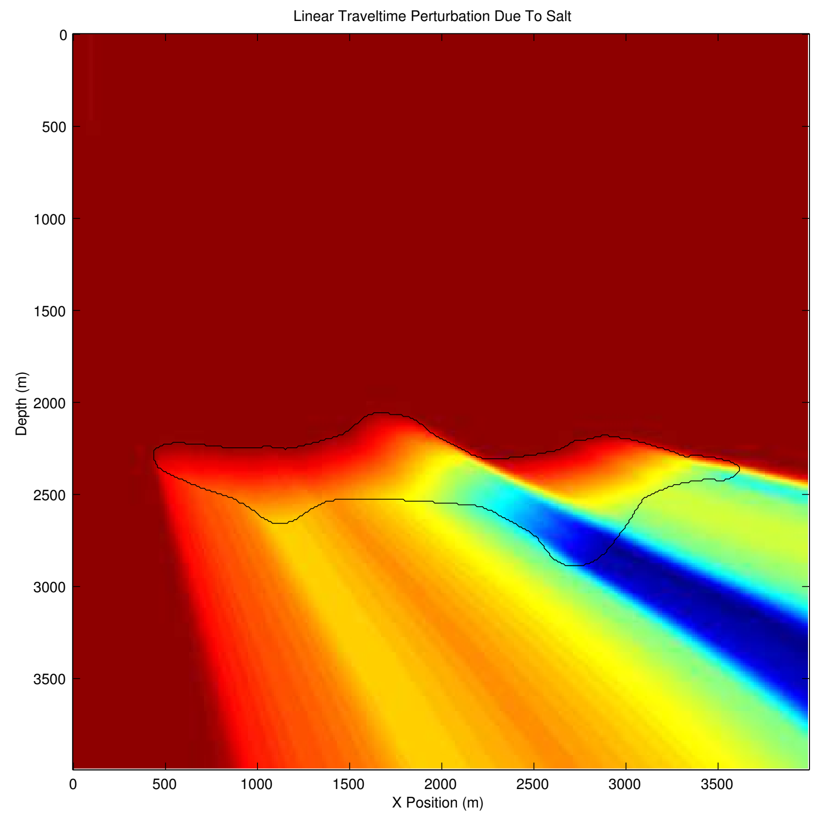




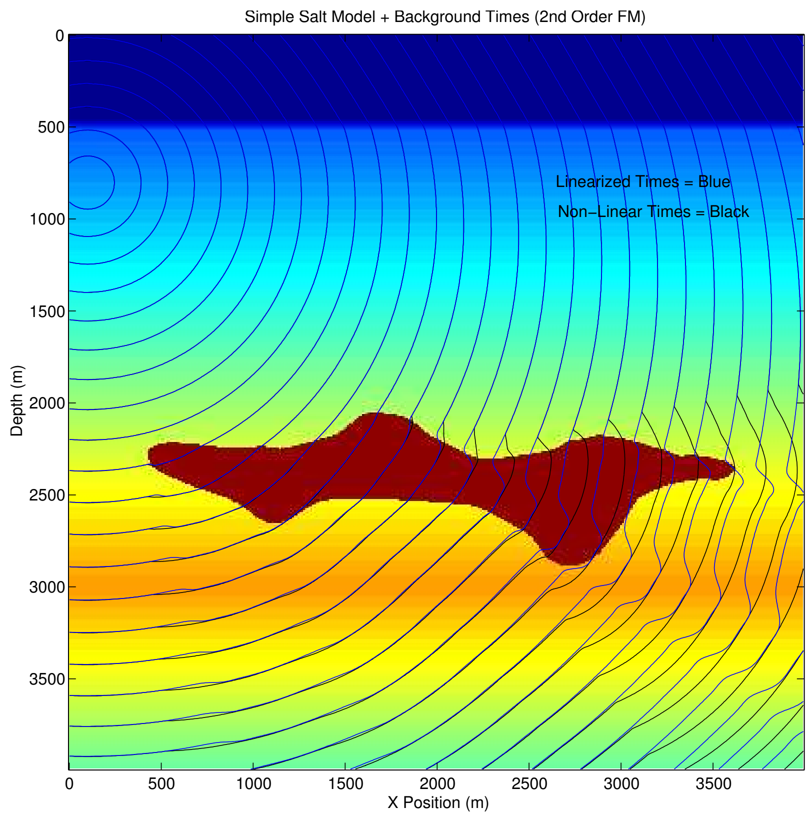
Peak Absolute Error For Linear Solvers (1 PLayer Model)
1st Order
2nd Order
Peak Absolute Error (s)
0.02
0.015
0.01
0.005
0
2
4
6
8
10
12
14
16
18
20
Grid Size (m)
Log Peak Absolute Error For Linear Solvers (1 PLayer Model)
Log Peak Absolute Error (s)
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{0}
10^{1}
10^{2}
Log Grid Size (m)


Peak Relative Error For Linear Solvers (1 PLayer Model)
1st Order
2nd Order
Peak Relative Error (%)
Grid Size (m)
2.5
2
1.5
1
0.5
0
2
4
6
8
10
12
14
16
18
20

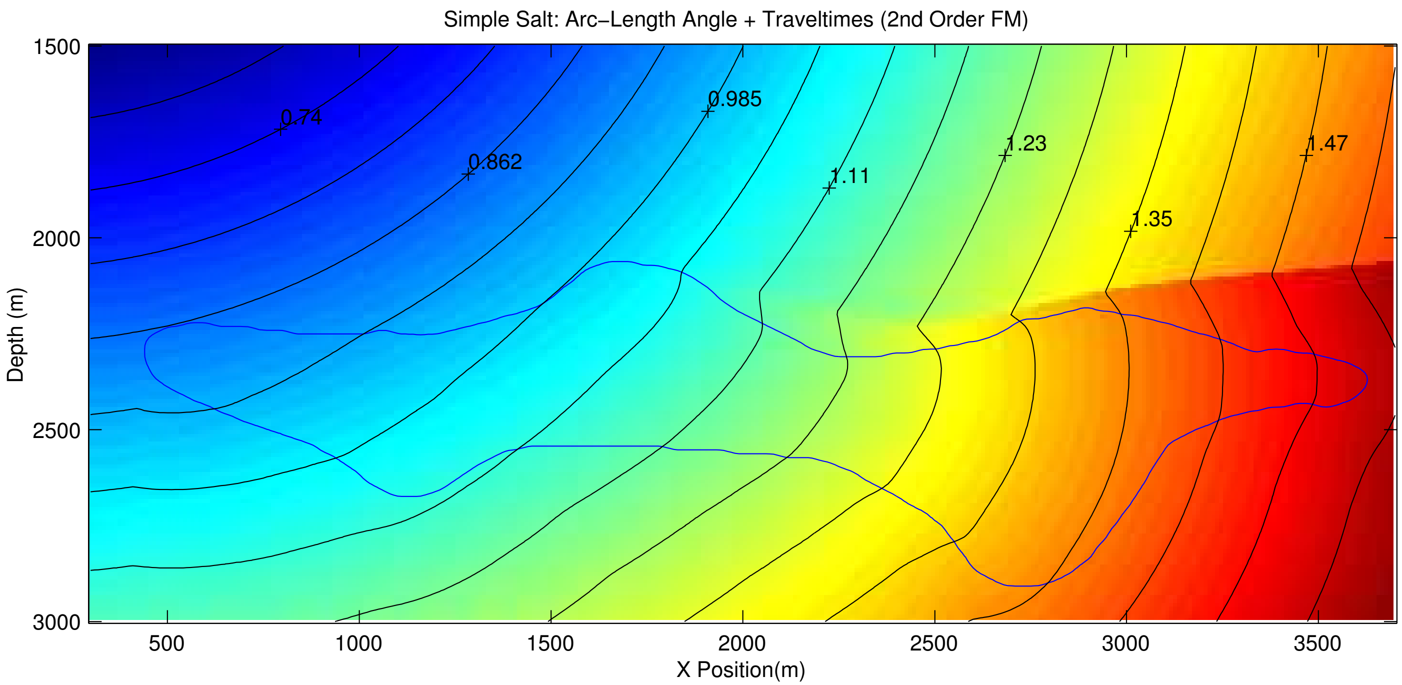

Log Peak Relative Error For Linear Solvers (1 PLayer Model)
Log Peak Relative Error (%)
Log Grid Size (m)
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{0}
10^{1}
10^{2}

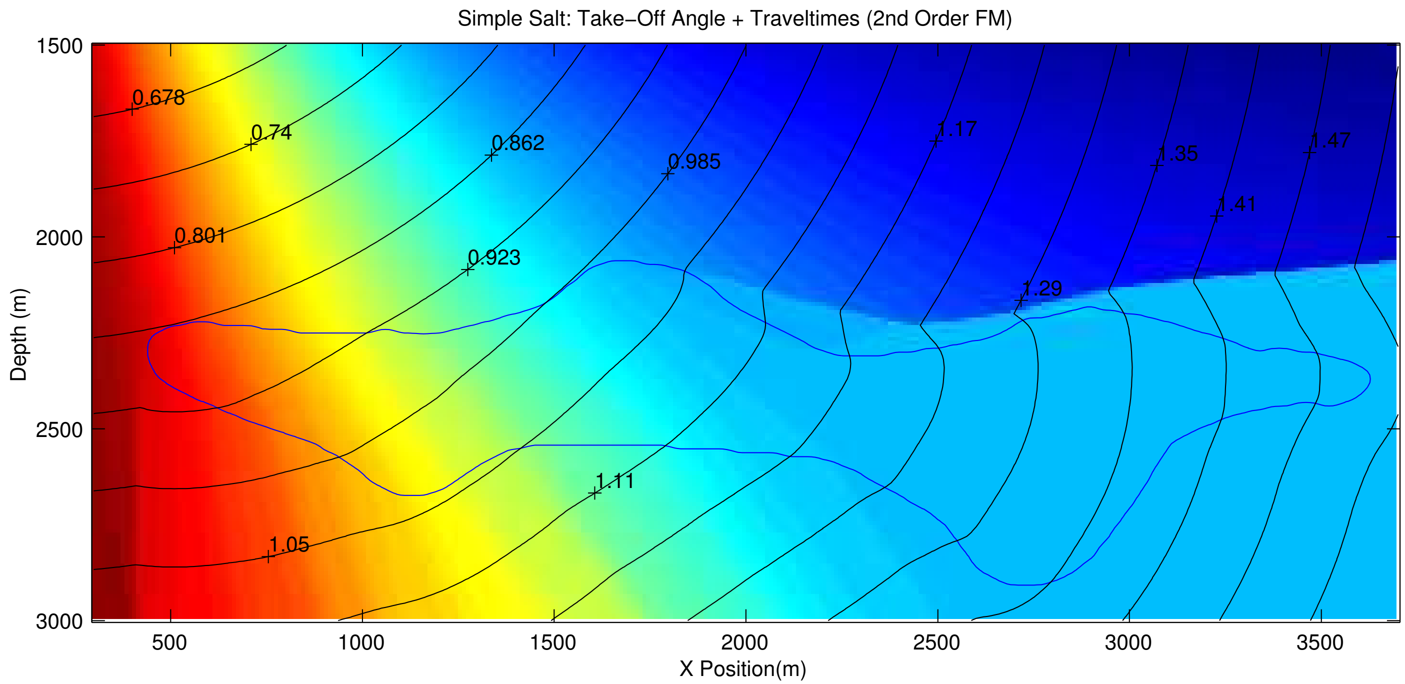


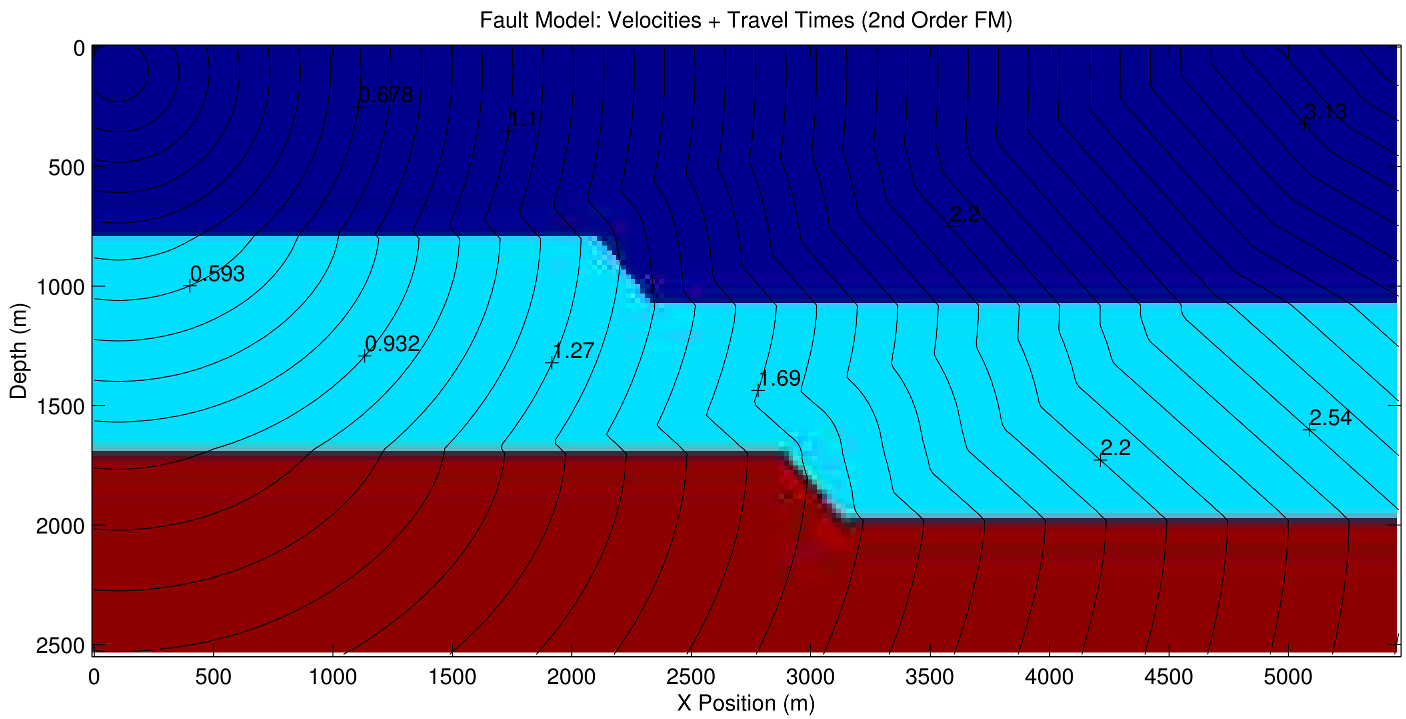
Linear Traveltime Perturbation Due To Salt
Depth (m)
X Position (m)
0
500
1000
1500
2000
2500
3000
3500



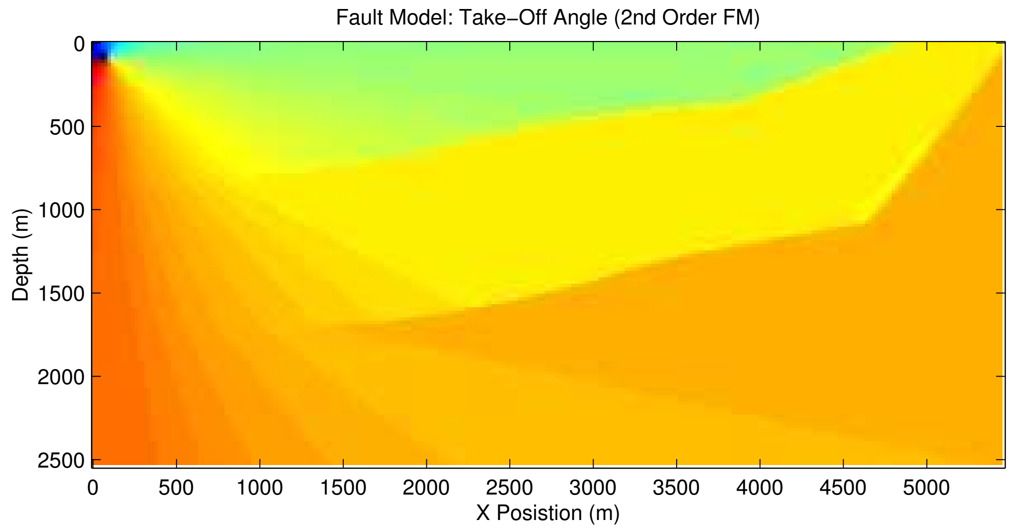
Simple Salt Model + Background Times (2nd Order FM)
Linearized Times = Blue
Non−Linear Times = Black
Depth (m)
X Position (m)
0
500
1000
1500
2000
2500
3000
3500

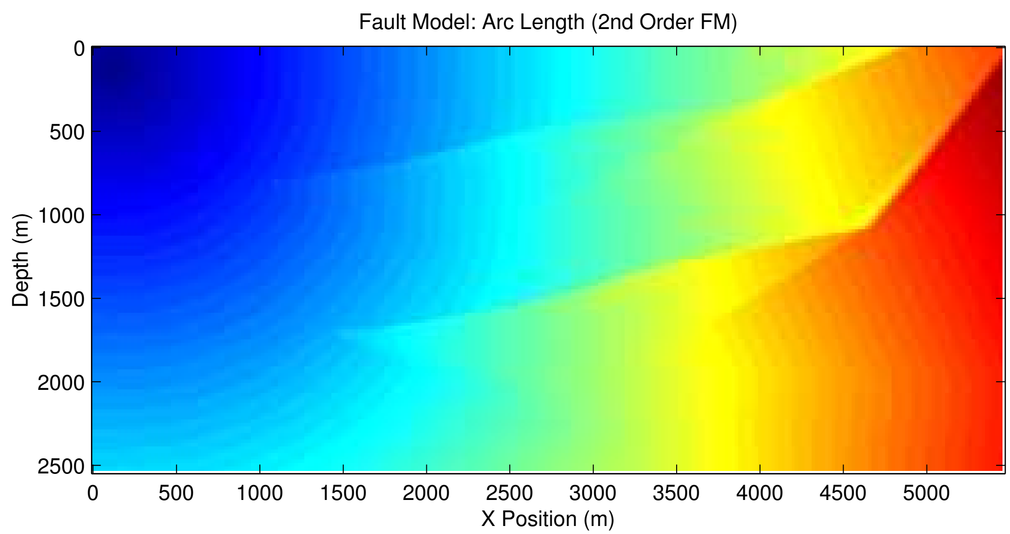

Simple Salt: Arc−Length Angle + Traveltimes (2nd Order FM)
0.74
0.862
0.985
1.11
1.23
1.35
1.47
Depth (m)
1500
2000
2500
3000
X Position(m)
500
1000
1500
2000
2500
3000
3500

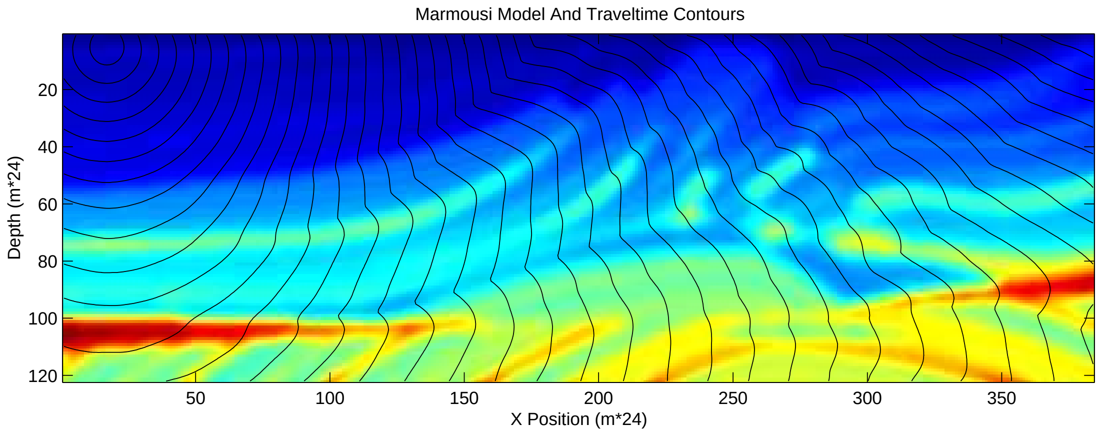

Simple Salt: Take−Off Angle + Traveltimes (2nd Order FM)
1500
2000
2500
3000
Depth (m)
500
1000
1500
2000
2500
3000
3500
X Position(m)
0.678
0.74
0.801
0.862
0.923
0.985
1.05
1.11
1.17
1.29
1.35
1.41
1.47

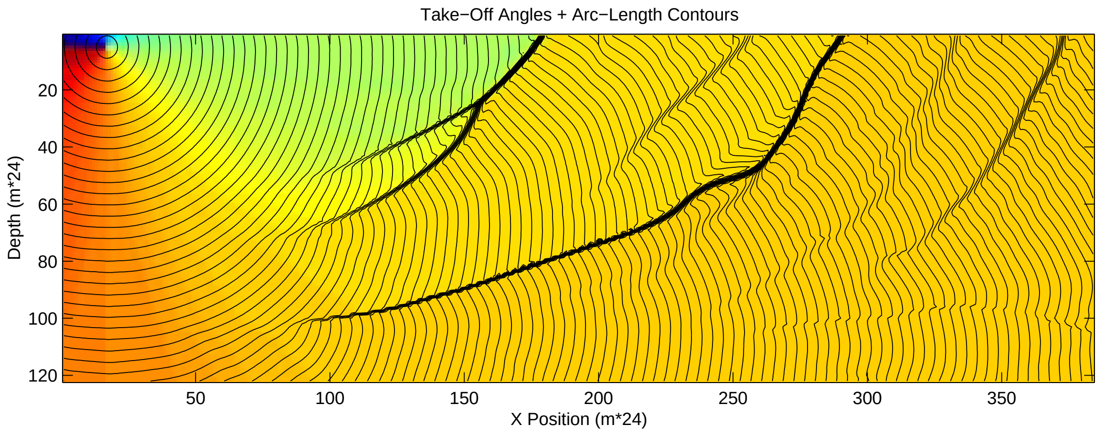


Fault Model: Velocities + Travel Times (2nd Order FM)
Depth (m)
X Position (m)
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
0.678
1.1
3.13
2.2
0.593
0.932
1.27
1.69
2.54
2.2


Fault Model: Take−Off Angle (2nd Order FM)
Depth (m)
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
X Posistion (m)


Fault Model: Arc Length (2nd Order FM)
Depth (m)
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
X Position (m)


Marmousi Model And Traveltime Contours
Depth (m*24)
20
40
60
80
100
120
50
100
150
200
250
300
350
X Position (m*24)


Take−Off Angles + Arc−Length Contours
Depth (m*24)
20
40
60
80
100
120
50
100
150
200
250
300
350
X Position (m*24)

Performance Aspects

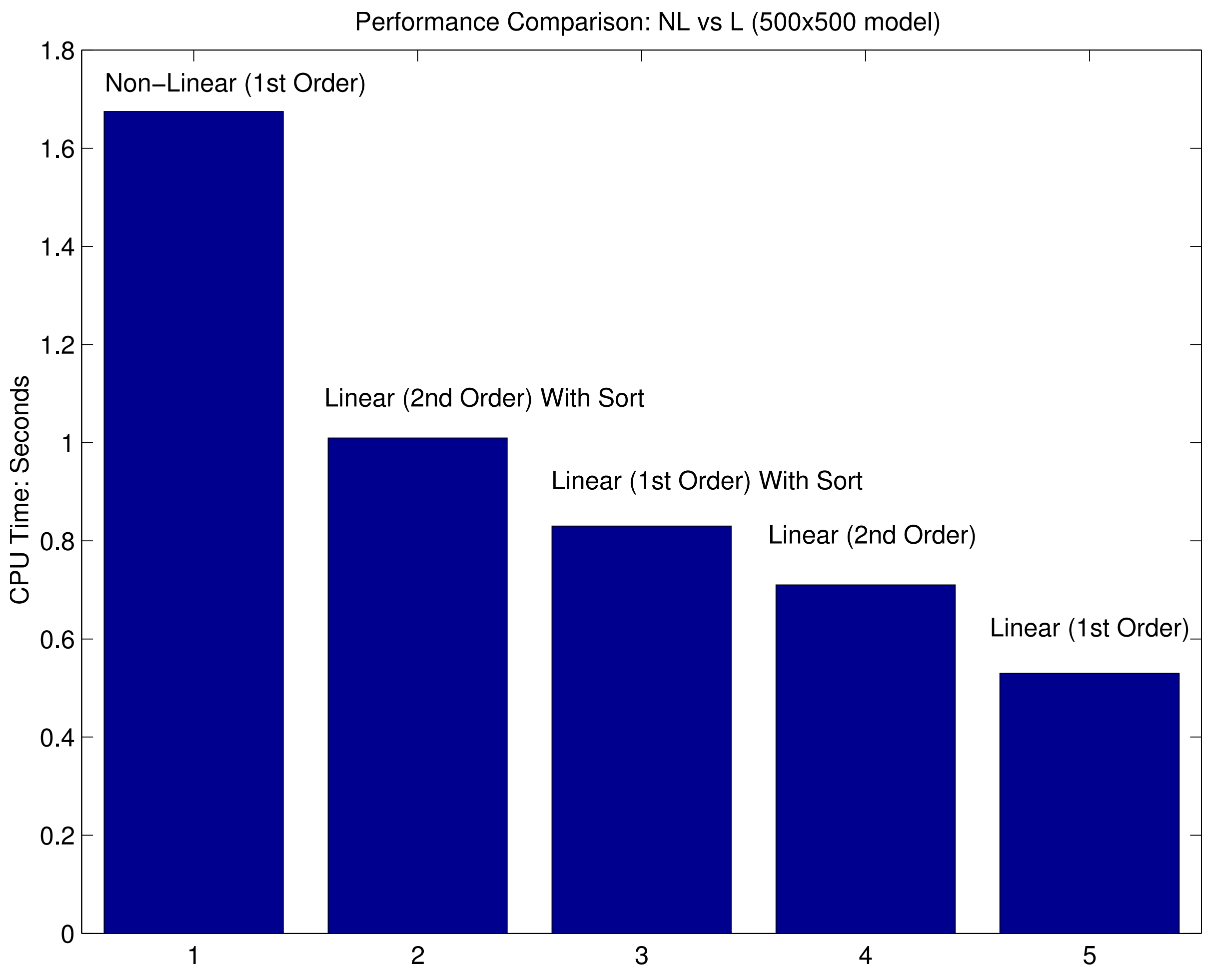
If the sorting operation is included, both linear and non−linear fast−marching algorithms are O(n lg n).

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)
CPU Time: Seconds
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Non−Linear (1st Order)
Linear (2nd Order) With Sort
Linear (1st Order) With Sort
Linear (2nd Order)
Linear (1st Order)
1
2
3
4
5

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.

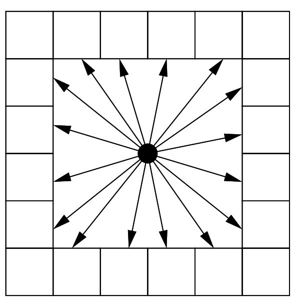
Dealing With Near–Source Numerics

The Problem:

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

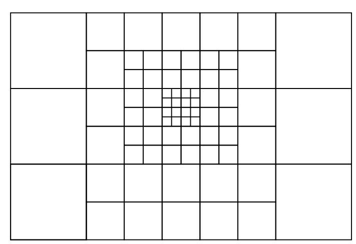
Solutions:

Ray Trace Near–Source Region



(Folklore)

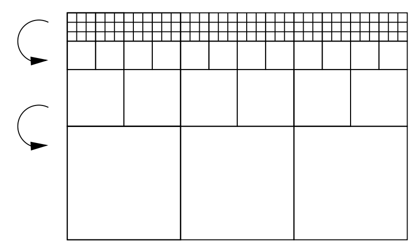
L.U.M.R. (Locally Uniform Mesh Refinement)



static

(Kim + Cook, 98')

Adaptive Grid Strategies



dynamic

(Belfi + Symes, 98')

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

1. Perfection of amplitude calculation

2. Stabilize higher–order FD schemes

3. Develop a scheme for calculating pulse broadening in attenuating media

4. Continue the quest for multivalued FD Traveltimes

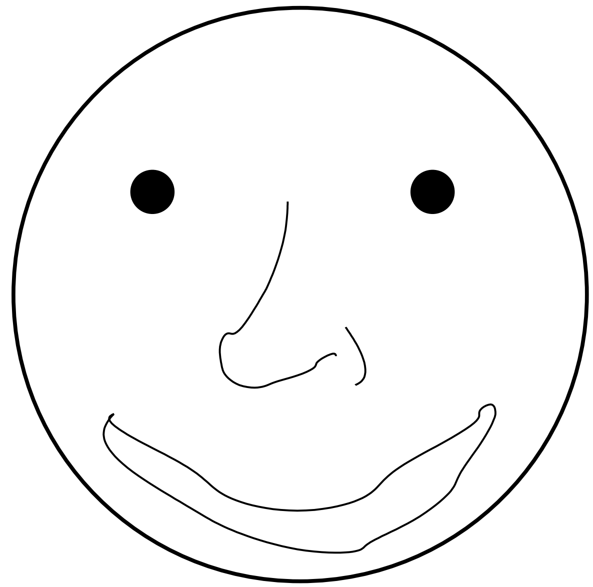
Many Thanks To ...

Guan Wang (STP)

Bill Symes (TRIP)

Sergei Fomel (SEP)

Jianliang Qian (TRIP)



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