Journal of Computational Acoustics,
© IMACS

A High-Order Fast Marching Scheme For The Linearized Eikonal Equation *

Jonathan B. Franklin
Stanford University
Mitchell Building Rm. 453B, 397 Panama Mall, Stanford, CA, 94305

Jerry M. Harris
Stanford University
Mitchell Building Rm. 321, 397 Panama Mall, Stanford, CA, 94305

Received (to be inserted
Revised by Publisher)

We present a high-order upwind finite-difference scheme for solving a useful family of first-order partial differential equations, of which the linearized eikonal equation is a member. Fast solutions of the linearized eikonal equation have applications in traveltime tomography and residual migration algorithms. The technique, besides being both accurate and stable, escapes aperture limitations inherent in static marching schemes. We use a time-sequential evaluation method similar to Sethian's Fast Marching strategy to insure causal operator evaluation. We apply our technique to several complex slowness distributions, including the Marmousi model. We also use an adaptation of our technique to compute Cartesian-to-Ray coordinate transforms for the same slowness models.

1. Introduction

Traveltime calculation exists as a major component of modern seismic imaging algorithms including reflection tomography and migration. Many of these algorithms are iterative in nature and require traveltimes for models only locally perturbed from an initial slowness distribution. In cases where this perturbation is sufficiently small, solving the linearized eikonal equation can be an efficient approach to estimating the accompanying traveltime variation.

We have developed a fast explicit finite-difference solution to the linearized eikonal equation which uses 2nd and higher order FD operators as well as a stable marching scheme. Our method improves upon previous FD schemes which have suffered from either aperture limitations [11] or poor computational performance due to implicit formulations [5]. The linearized eikonal equation is a member of an interesting class of linear PDEs, which includes the equations governing the transformation between Cartesian and ray coordinate systems.

[*] Presented at ICTCA '99 : the 4th International Conference on Theoretical and Computational Acoustics, May 1999, Trieste, Italy

We begin by deriving the linearized eikonal equation and describing some related PDEs that are potentially useful in the context of seismic imaging. We then describe the two components of our algorithm, a stable marching scheme and an accurate class of higher order traveltime extrapolation stencils. The algorithm is then extended to solve the Cartesian-to-ray mapping problem and tested on a simple model for which analytic solutions are available. After applying both algorithms to realistic geologic models including the Marmousi model, we conclude with a discussion of some problems we encountered in the development process.

2. Deriving The Linearized Eikonal Equation

The familiar non-linear eikonal equation relates a slowness distribution to a traveltime field and is valid only at the high frequency limit. The non-linear eikonal equation can be derived through WKBJ theory [6] or via variational methods and Fermat's principle. Expressed in two dimensions, the non-linear eikonal equation is

$$\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial z}\right)^2 = S(x,z)^2, \tag{2.1}$$

where S and t are the two dimensional slowness and traveltime fields respectively. The linearized eikonal equation arises in several contexts but is most frequently appealed to in traveltime tomography [1], where it is used to justify the linear relation between perturbations in slowness (δS) and perturbations in traveltime (δt), or

$$\int_L \delta S \, d\ell \approx \delta t, \tag{2.2}$$

where L is a given ray-path and $d\ell$ represents a differential arc-length element. To derive the linearized eikonal equation, we perturb both the slowness and traveltime fields around background fields S_0 and t_0.

$$S = S_0 + \delta S \qquad\qquad t = t_0 + \delta t, \tag{2.3}$$

For convenience in notation we replace δt with τ when referring to the perturbed time field. Substituting these expressions back into the eikonal equation and expanding yields

$$\left(\frac{\partial t_0}{\partial x}\right)^2 + \left(\frac{\partial t_0}{\partial z}\right)^2 + \left(\frac{\partial \tau}{\partial x}\right)^2 + \left(\frac{\partial \tau}{\partial z}\right)^2 + 2\left[\frac{\partial t_0}{\partial x}\frac{\partial \tau}{\partial x} + \frac{\partial t_0}{\partial z}\frac{\partial \tau}{\partial z}\right] = S_0^2 + 2S_0\,\delta S + \delta S^2. \tag{2.4}$$

Subtracting the non-linear eikonal equation in t_0, dropping all higher order terms in τ and δS, and division produces the linearized eikonal equation

$$\frac{\partial t_0}{\partial x}\frac{\partial \tau}{\partial x} + \frac{\partial t_0}{\partial z}\frac{\partial \tau}{\partial z} = S_0\,\delta S. \tag{2.5}$$

Intuitively, solving the linearized eikonal equation can be thought of as integrating slowness perturbations over a path defined by the gradient of t_0. In the case of a constant slowness background, this amounts to the integration of δS over straight rays. Figure 1 shows

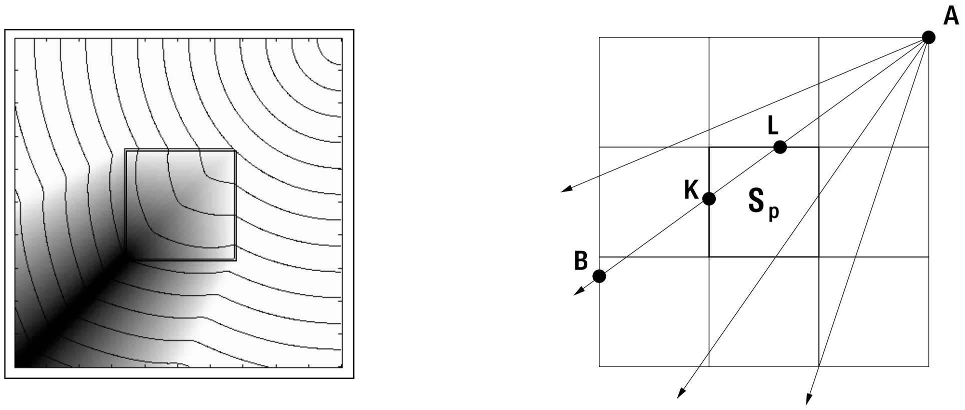

Fig. 1. A pictorial description of the solution to the linearized eikonal equation (5) : in the left panel (colors and contours map to $t_0 + \tau$) we see the effect that a slowness perturbation (S_P) within the central block has on travel times for a constant slowness background. The right panel depicts the point source, located at A, and a series of characteristic curves. The ray connecting A to B is slowed while traversing the K-L segment.

the effects of a rectangular slowness perturbation on travel times, for a constant slowness background.

3. A Useful Family Of Equations

The linearized eikonal equation derived above is just one example of a rather interesting class of first-order partial differential equations which can be written as

$$\frac{\partial t}{\partial x}\frac{\partial u}{\partial x} + \frac{\partial t}{\partial z}\frac{\partial u}{\partial z} = R(x,z), \tag{3.6}$$

where u is typically a variable that is integrated or fixed along the ray-paths defined by time field t and R is a spatially varying material property. This form can be used, for example, to develop equations describing the mapping between Cartesian coordinates (x, z) and ray coordinates (θ, ℓ) [19], where θ is ray take-off angle and ℓ: is arc-length.

$$\frac{\partial t}{\partial x}\frac{\partial \theta}{\partial x} + \frac{\partial t}{\partial z}\frac{\partial \theta}{\partial z} = 0, \tag{3.7}$$

$$\frac{\partial t}{\partial x}\frac{\partial \ell}{\partial x} + \frac{\partial t}{\partial z}\frac{\partial \ell}{\partial z} = S. \tag{3.8}$$

Equation (3.7) simply states that take-off angle is constant along any given ray-path, hence only take-off angle and arc length (3.8) are required to describe a point in ray coordinates. In equation (3.8), S is an unperturbed slowness field.

Further extensions can be developed to describe geometric spreading and hence to estimate amplitudes [15] [18]. The finite-difference scheme which we will describe is useful for solving equations in the general form of (3.6).

4. A Finite-Difference Algorithm

In most situations involving the linearized eikonal equation (2.5), t_0, S_0, and δS are known and the goal is to determine the related traveltime perturbation field, τ. We wish to accurately solve for τ on a regular rectangular grid with spacings Δx and Δz. Explicit finite-difference (FD) techniques are a highly efficient approach to solving this type of problem. When developing such a scheme, one must choose a sufficiently accurate set of *local* FD operators and a stable *global* evaluation pattern or marching order. The accuracy of the FD operator is particularly crucial in cases where higher derivatives of the calculated field are desired. A stable marching scheme is critical for development of a robust method that can accommodate complex slowness fields. Another important component for such numerical schemes is an accurate treatment of initial conditions and the near-source problem [14] [2].

4.1. *Previous Work*

The literature abounds with finite-difference treatments of the non-linear eikonal equation: seminal papers by Reshef and Kosloff [9], Vidale [17], Podvin and Lecomte [7], and van Trier and Symes [16] have been followed by more recent methods that exploit flexible wave-like marching schemes to preserve causality [8] [10] [12] [13].

Previous finite-difference solutions to the linearized eikonal equation have either been formulated implicitly [5] or explicitly with an aperture limitation. The implicit techniques, while robust, are too computationally intensive to justify use as a cheap alternative to explicit solution of the non-linear eikonal equation. The depth-stepping finite-difference schemes used by Symes and others [11] [15], while fast, cannot accommodate models where rays experience significant bending without some type of iterative correction scheme [14]. Both of these problems can be avoided as we do here, by using an adaptation of the Fast Marching algorithm developed by Sethian [12] [13].

4.2. *A Stable Global Marching Scheme*

Static marching schemes, such as depth or box expansion, cannot guarantee first arrivals in all cases [8]. More stable marching schemes for the non-linear eikonal equation have been obtained by evaluating upwind finite-difference operators in order of increasing time, starting from an initial zero-time source array. This marching order mimics the physical wavefront and insures that the true first arrival time is obtained. Among these techniques, Sethian's Fast Marching algorithm [12] [13], has proved the most popular. In practice, the most difficult part of implementing such wavefront-expansion solvers is an efficient method for choosing the minimum traveltime node for updating. Sethian uses binary heaps to produce an algorithm that is $O(n \lg n)$ where n is the total number of grid points within the model. The "narrow-band" philosophy used in Fast-Marching techniques is shown in figure 2.

We use a similar technique for solving the linearized eikonal equation. However, since the characteristics of the PDE are already captured in the background traveltime field t_0, we can simply sort these times and solve for τ in this order. Since only a pre-sorting operation is required, we do not have to maintain a binary heap during the FD evaluation stage. This allows use of Hoare's quicksort algorithm which, although asymptotically equivalent to heapsort in average-case runtime, is faster in practice [3]. A simple flow-chart of our

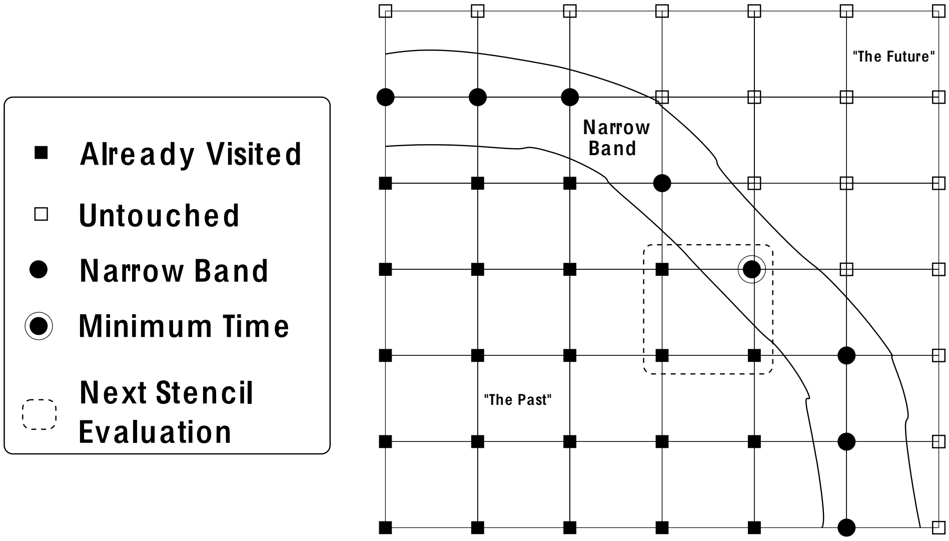

Fig. 2. A schematic representation of the Fast Marching algorithm. The FD operator is evaluated in order of increasing times to insure that causality is preserved. When solving the linearized eikonal equation, this ordering is obtained by sorting the background traveltimes, t_0, and hence we do not require an active binary heap within the marching phase.

algorithm is shown in figure 3.

Since the pre-sorting step is $O(n \lg n)$ in comparison to the $O(n)$ complexity of the actual stencil evaluations, considerable attention is paid to efficiently implementing the quicksort. Recursion and associated call overhead is replaced with a managed stack. A low-level key randomization phase is used to insure that quicksort is not laid low by its $O(n^2)$ worst-case bound. We have also explored some cache-use optimizations which have potential to further decrease runtime.

4.3. *Accurate Upwind Finite Difference Operators*

Upwind FD operators sample the field used for derivative calculation in a single direction. This property is crucial in cases where we wish information to propagate across space in a causal manner: the traveltime at a given point should be calculated using only the information from previous times. The sorted marching scheme, described previously, insures that the point being updated is on the "front" and is adjoining a region where times have already been calculated. Use of an appropriate upwind FD operator guarantees that only previous times are exploited in the traveltime extrapolation.

Upwind approximations to the x and z derivatives can be derived via Taylor series or through differentiation of an interpolating polynomial (see [4] or the Appendix). Higher-order approximations require more points for derivative calculation. If τ_i is the point where we wish to calculate $\partial\tau/\partial x$, upwind FD operators of order 1 through 4 can be expressed as

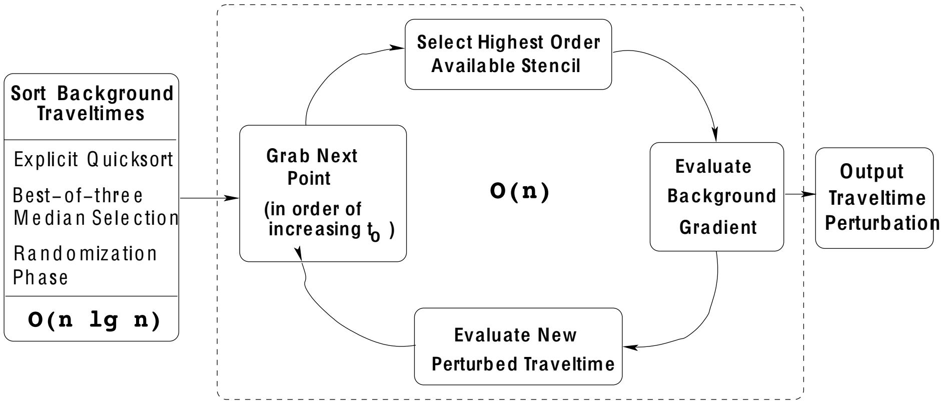
Sort Background Traveltimes
Explicit Quicksort
Best-of-three Median Selection
Randomization Phase
O(n lg n)
Grab Next Point
(in order of increasing t0)
Select Highest Order Available Stencil
O(n)
Evaluate Background Gradient
Evaluate New Perturbed Traveltime
Output Traveltime Perturbation

Fig. 3. A basic flow chart for our linearized eikonal solver. The pre-sorting phase is used to calculate a stable marching scheme. After sorting, appropriate high-order FD stencils are evaluated in order of increasing time to calculate τ.

$$\text{1st Order} \quad \frac{\partial\tau}{\partial x} = \frac{\tau_{i+1} - \tau_i}{\Delta x} \tag{4.9}$$

$$\text{2nd Order} \quad \frac{\partial\tau}{\partial x} = \frac{-\tau_{i+2} + 4\tau_{i+1} - 3\tau_i}{2\,\Delta x} \tag{4.10}$$

$$\text{3rd Order} \quad \frac{\partial\tau}{\partial x} = \frac{2\tau_{i+3} - 9\tau_{i+2} + 18\tau_{i+1} - 11\tau_i}{6\Delta x} \tag{4.11}$$

$$\text{4th Order} \quad \frac{\partial\tau}{\partial x} = \frac{-\tau_{i+4} + 6\tau_{i+3} + 18\tau_{i+2} + 10\tau_{i+1} - 33\tau_i}{60\Delta x}. \tag{4.12}$$

Once a FD operator is chosen, we can replace the derivatives in the linearized eikonal equation. The spatial derivatives of t_0 can be calculated in the same upwind fashion as the derivatives of τ or they can be evaluated statically before solution of the linearized problem begins. We assume that the derivatives of t_0 are available at all points and use first-order upwind operators as an initial example:

$$\left(\frac{\partial t_0}{\partial x}\right)_{i,j} \left(\frac{\tau_{i+1,j} - \tau_{i,j}}{\Delta x}\right) + \left(\frac{\partial t_0}{\partial z}\right)_{i,j} \left(\frac{\tau_{i,j+1} - \tau_{i,j}}{\Delta z}\right) = S_{o_{i,j}}\, \delta S_{i,j}. \tag{4.13}$$

We solve this equation for $\tau_{i,j}$, thereby obtaining an expression for extrapolating the value at (i, j) given knowledge of τ at $(i, j+1)$ and $(i+1, j)$ (see figure 4).

$$\tau_{i,j} = \frac{(\frac{\partial t_0}{\partial x})\tau_{i+1,j} + (\frac{\partial t_0}{\partial z})\tau_{i,j+1} - h\, S_{o_{i,j}}\, \delta S_{i,j}}{(\frac{\partial t_0}{\partial x}) + (\frac{\partial t_0}{\partial z})}. \tag{4.14}$$

We assume $\Delta x = \Delta z = h$ for simplicity, although inclusion of these factors is not difficult. A more general form for extrapolation formulas like equation (4.14) can easily be derived for an arbitrary order upwind finite-difference operator. A compact form for writing such a FD operator is

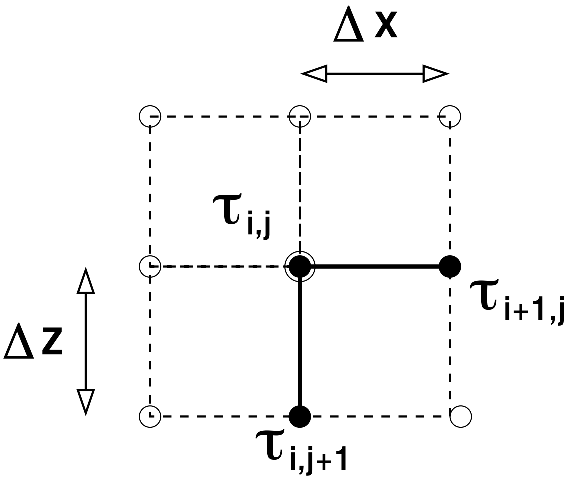

Fig. 4. A schematic showing a first-order upwind extrapolation that uses the values of τ at $(i, j+1)$ and $(i+1, j)$ to determine the value at (i, j).

$$\frac{\partial \tau}{\partial x} = \frac{1}{g_x\,\Delta x}\sum_{i=0}^{n} c_i\,\tau_{i,j} \tag{4.15}$$

$$\frac{\partial \tau}{\partial z} = \frac{1}{g_z\,\Delta z}\sum_{j=0}^{m} c_j\,\tau_{i,j}, \tag{4.16}$$

where c_i and c_j indicate the appropriate coefficient for the difference formula. The variables g_x and g_z are the multiplicative coefficients in the denominator of the FD stencil.

Assume again that the spacing in each dimension is equal, $h = \Delta x = \Delta z$, and let the summation portion of these operators be P_x and P_z respectively. We can now substitute these difference expressions into the linearized eikonal equation and solve for $\tau_{i,j}$ to produce an explicit linearized eikonal extrapolation formula for an arbitrary order operator,

$$\tau_{i,j} = \frac{(h\,g_x\,g_z\,S_{0_{i,j}}\,\delta S_{i,j}) - (g_z\,\frac{\partial t_0}{\partial x}\,P_x) - (g_x\,\frac{\partial t_0}{\partial z}\,P_z)}{(g_z\,c_{x0}\,\frac{\partial t_0}{\partial x}) + (g_x\,c_{z0}\,\frac{\partial t_0}{\partial z})}. \tag{4.17}$$

Similar expressions are easily derived for the ray-Cartesian mapping equations, (3.7) and (3.8), or for any equation of form (3.6). The extrapolation formulae for the take-off and arc-length equations become

$$\theta_{i,j} = \frac{(g_x\,P_z\,\frac{\partial t}{\partial z}) - (g_z\,P_x\,\frac{\partial t}{\partial x})}{(g_z\,c_{x0}\,\frac{\partial t}{\partial x}) + (g_x\,c_{z0}\,\frac{\partial t}{\partial z})} \tag{4.18}$$

$$\ell_{i,j} = \frac{(h\,g_x\,g_z\,S_{i,j}) - (g_z\,\frac{\partial t}{\partial x}\,P_x) - (g_x\,\frac{\partial t}{\partial z}\,P_z)}{(g_z\,c_{x0}\,\frac{\partial t}{\partial x}) + (g_x\,c_{z0}\,\frac{\partial t}{\partial z})}, \tag{4.19}$$

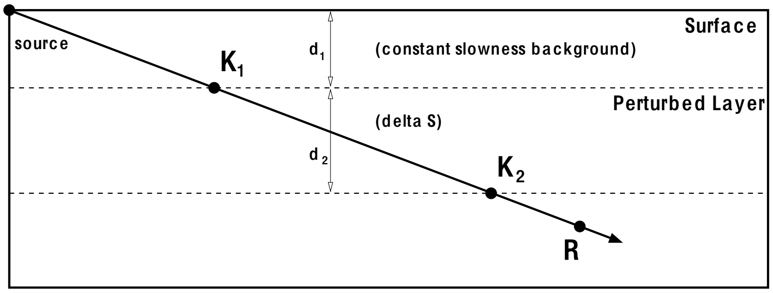

Fig. 5. A schematic depicting a simple single-layered model for which an analytic solution is available.

where the difference operators are defined over the appropriate variable. Our implementations allow the user to specify the order of operator used in solving the linearized eikonal or mapping equations, although operators of order higher than 3 have produced stability problems within our scheme.

5. Analytic Solutions For Models With A Single Perturbed Layer

To test the accuracy of our linearized eikonal solver, we compared computed solutions to analytic results for a simple model with a homogeneous background slowness field and a single perturbed layer, as shown in figure 5. The model assumes a point source at $(0,0)$, the upper left corner, and a single perturbed layer of thickness d_2 starting at depth d_1. Since the S_0 is constant, the solution to the linearized eikonal equation is simply a path integral over a straight ray. Evaluation of τ at location (x,z) reduces to,

$$\begin{aligned} \text{if } z < d_1, \quad & \textstyle\int_L \delta S d\ell &&= 0 \\ \text{if } d_1 \le z \le (d_1+d_2), \quad & \textstyle\int_L \delta S d\ell &&= \delta S[||K_R|| - ||K_1||] \\ \text{if } z > (d_1+d_2), \quad & \textstyle\int_L \delta S d\ell &&= \delta S[||K_2|| - ||K_1||], \end{aligned} \tag{5.20}$$

where the K components (segment lengths) can be expressed in closed form as

$$\begin{aligned} ||K_1|| &= \frac{d_1}{cos(tan^{-1}\frac{x}{z})} \\ ||K_2|| &= \frac{d_1+d_2}{cos(tan^{-1}\frac{x}{z})} \\ ||K_R|| &= \frac{z}{cos(tan^{-1}\frac{x}{z})}. \end{aligned} \tag{5.21}$$

We performed our error test on a 1000 by 1000 m domain with a single 200 m thick perturbed layer at a depth of 400 m. The velocity of the layer was perturbed 3% from the background wavespeed. Figure 6 shows peak relative error statistics for our 1st and 2nd order linearized eikonal solvers over a variety of cell dimensions. The lower log-log plot clearly shows the 2nd order convergence of the 2nd order method.

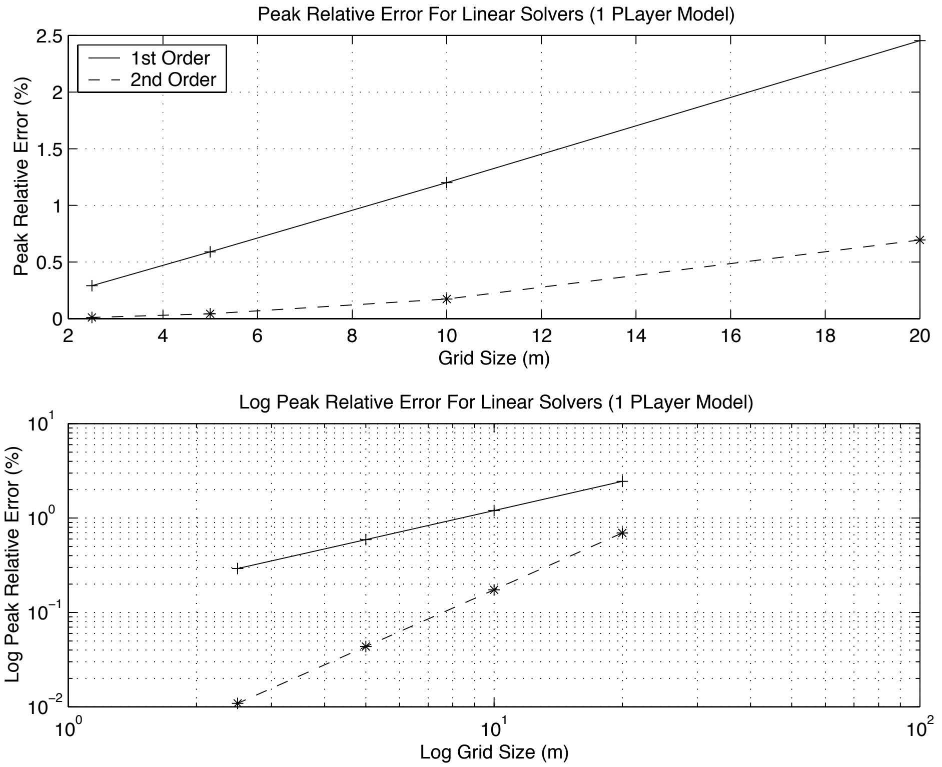

Fig. 6. Relative error and log error plots for a variety of grid sizes. The calculation was performed on a 1000 by 1000 m model with a single perturbed layer.

As is shown in figure 6, the 2nd order solver is extremely accurate: peak relative errors of less than 1 percent are maintained even for very coarse (20m) samplings. Similar tests for the arc-length and take-off angle calculation schemes on constant slowness models exhibited the same error characteristics.

6. Realistic Examples

Although homogeneous and one-layer models are excellent for accuracy analysis, tests on geologically realistic slowness distributions provide the best test of robust behavior. We ran our linearized eikonal solver on a synthetic salt-model with aspects similar to several Gulf coast salt features. We tested the Cartesian-to-ray mapping scheme on both the salt model and the classic Marmousi model.

As a preliminary test case we developed a relatively simple salt model. The background was a smoothly varying $v(z)$ field with several different velocity gradients. We chose to add in the salt feature as a perturbation, located at a central depth of 2500 m. While adding a salt feature is an unrealisticly large slowness perturbation for either tomographic or residual migration iteration, the abrupt addition both stresses the algorithm and provides an example of the difference between the linear and non-linear eikonal solutions. To compute the non-linear times we used a modified version of Sethian's Fast Marching algorithm [13] with second-order FD operators. Figure 7 shows the velocity model. In regions where the salt-body introduces significant ray-bending (far right) the linear and non-linear solutions diverge, as would be expected. Figure 8 shows the traveltime perturbation due to the salt body.

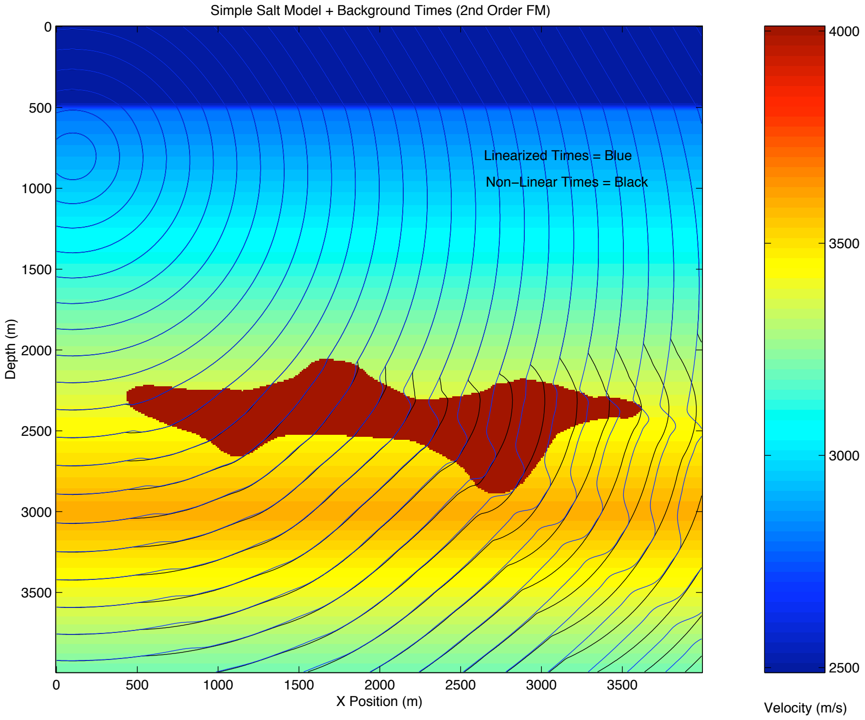

Fig. 7. A simple salt model with superimposed contours indicating both linear (blue) and non-linear (black) eikonal solutions

The results of the take-off angle calculation applied to the composite salt-model are also interesting. In figure 9 we see some of the focusing effects produced by the salt body's curvature.

Another test of our take-off angle and arc-length solvers was the popular Marmousi velocity model, developed from studies in Angola's Cuanza Basin. The model includes a heavily faulted anticline with high velocity salt features and has become a standard test model for traveltime calculation. From an imaging perspective, the desired target is a small reservoir within the lower section of the anticline. The model is composed of 384x122 samples with a grid unit of 24 meters in both the x and z dimensions.

Of the two panels in figure 10, the top one displays the Marmousi velocity model with 2nd order non-linear eikonal traveltimes superimposed, while the bottom shows the results of calculating take-off angle and arc-length using update operators (4.18) and (4.19). Several interesting features are evident in the lower panel: due to head wave generation, the angular family of rays implicitly followed by the eikonal solver is very narrow. At the far

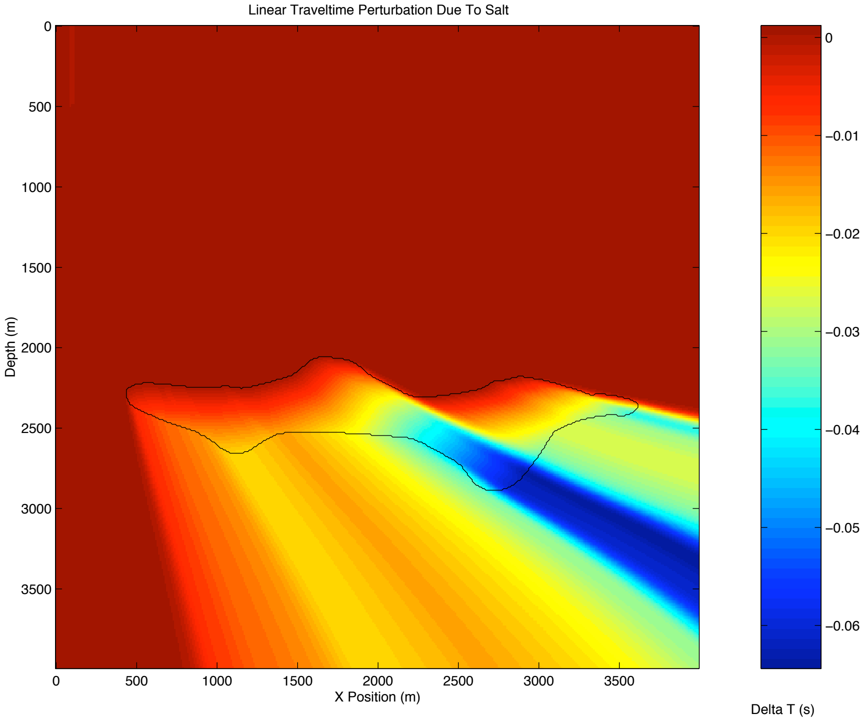

Fig. 8. Traveltime perturbations due to addition of the salt-body. Note that the values are negative indicating a time speed-up due to the salt feature's high velocity.

right edge of the model, almost all of the rays followed are from within a 1 degree fan. Discontinuities in arc-length delineate ray "families" and offer a rough guide of how the first-arrivals are partitioned in space. Mathematically, these discontinuities indicate shocks where multiple characteristics converge. They occur in regions where the time field should be non-differentiable i.e. when "corners" exist in the contours. This phenomenon suggests that the arc-length map might be useful as a computational shock or caustic detection tool.

7. Performance Analysis

Some preliminary examination of performance suggests that the linearized eikonal equation is only useful as an update operator when the pre-sorting cost is amortized over multiple updates or is determined during traveltime computation of the background field (a natural by-product of using a Fast Marching solver for t_0).

The cost of this type of FD algorithm can be divided into the marching costs ($O(n \lg n)$ in this case) and the stencil evaluation costs ($O(n)$). While our linearized eikonal solver is computationally equivalent to a non-linear Fast Marching eikonal solver in an asymptotic or "Big O" sense, the constants associated with evaluating the linearized FD stencil are substantially smaller due to the absence of square-root operations. The constants associated with quicksort are also substantially lower than those of the dynamic heapsort use in Fast

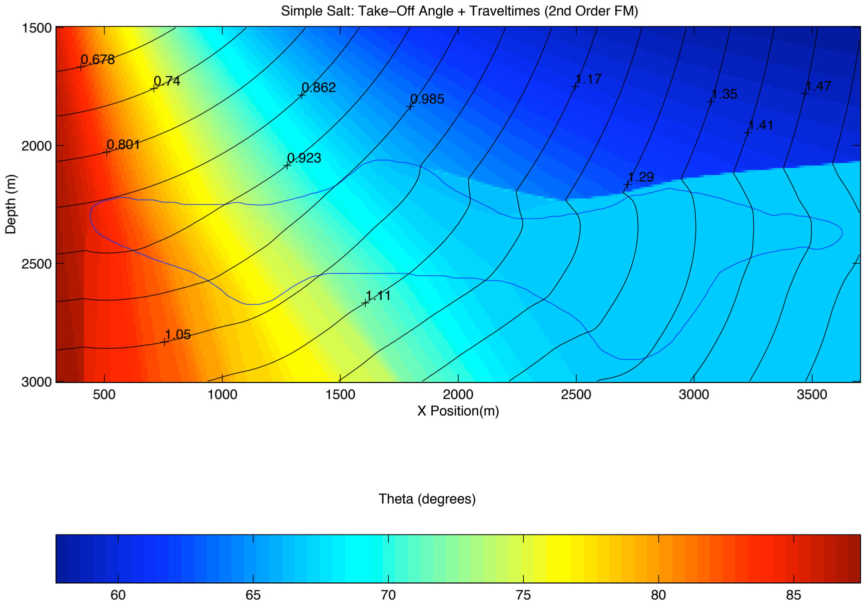

Fig. 9. Take-off angle map with superimposed 2nd order non-linear eikonal traveltimes

Marching. Figure 11 shows a performance comparison for linear and non-linear eikonal algorithms solving for times (or perturbations) over a 500 by 500 2D model.

8. Remarks And Conclusions

Care must be given to the proper treatment of initial state and near-source conditions. Since a singularity exists in the traveltime field at the point source, direct application of finite-differences in the near-source region will introduce 1st order error into any scheme, regardless of formal order. We tackled this problem using a combination of straight-ray tracing and LUMR or Locally Uniform Mesh Refinement [14] although elegant solutions based on fully adaptive grids are available [2], if slightly more difficult to develop. We experienced some difficulties in using our Cartesian-to-ray coordinate transform as the basis for a geometric amplitude estimation code. This might be due to insufficiently smooth traveltime derivatives.

We have developed a high-order FD scheme for solution of the linearized eikonal equation. The method is efficient, stable, and free from aperture limitations. Comparison to the analytic solution for simple three-layer slowness models has quantitatively verified the method's accuracy while tests on more complicated structures, including the Marmousi model, have yielded reasonable results. Performance analysis suggests that the solver is most useful in situations where computation of the marching order can be amortized over multiple evaluations of δt. Use of similar techniques for computing Cartesion-to-Ray coor-

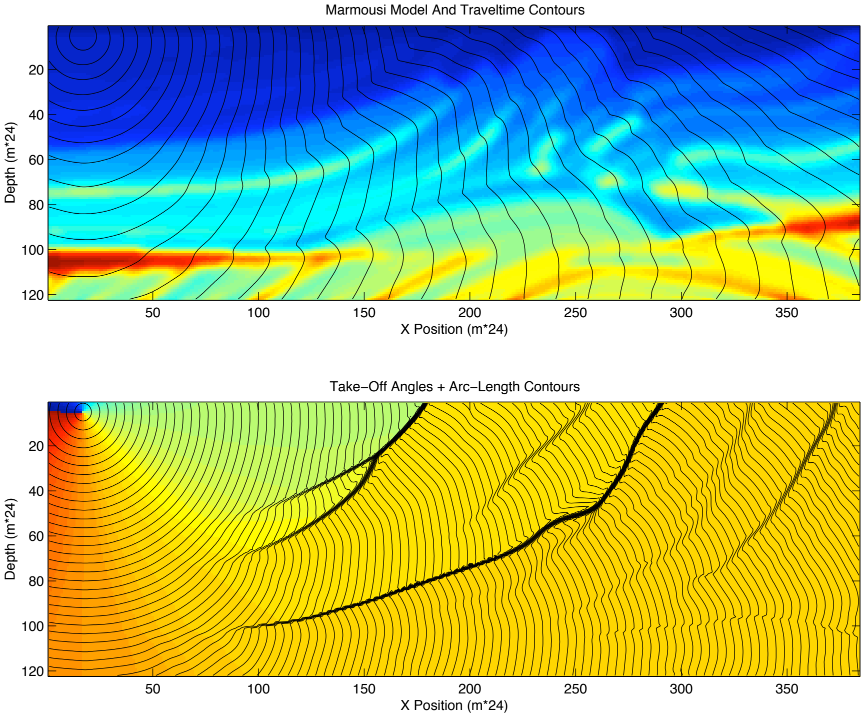

Fig. 10. The top panel shows the Marmousi velocity model with 2nd order non-linear traveltime contours superimposed. The bottom panel shows take-off angle calculations for the same model with superimposed arc-length contours.

dinate transformations offer an interesting way of viewing the results of gridded traveltime calculation algorithms.

Appendix A

For completeness, we derive our second-order upwind finite-difference operator by differentiating a Lagrange interpolating polynomial. A polynomial of degree $N-1$ can interpolate N points: assume the form $\tau_1 = \tau(x_1)$, $\tau_2 = \tau(x_2), ..., \tau_n = \tau(x_n)$ and the interpolating polynomial is given explicitly by Lagrange's formula,

$$\tau(x) = \frac{(x-x_2)(x-x_3)...(x-x_N)}{(x_1-x_2)(x_1-x_3)...(x_1-x_N)}\tau_1 + \frac{(x-x_1)(x-x_3)...(x-x_N)}{(x_2-x_1)(x_2-x_3)...(x_2-x_N)}\tau_2 + \\ ... + \frac{(x-x_1)(x-x_2)...(x-x_{N-1})}{(x_N-x_1)(x_N-x_2)...(x_N-x_{N-1})}\tau_N \qquad \text{(A.1)}$$

We are interested in a second-order FD approximation, or equivalently, the derivative of an interpolating quadratic polynomial in x with respect to τ. A quadratic fit requires three points and can be written as an instance of Lagrange's formula

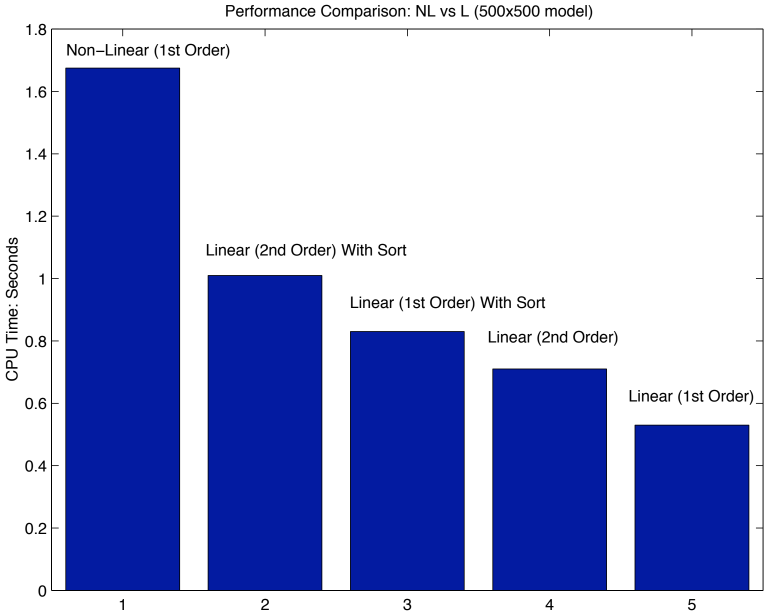

Fig. 11. Performance comparison for non-linear and linear eikonal solvers of various order with and without the pre-sorting step. The algorithms were executed on a single processor 233mhz Pentium 1 (MMX) with 48 megabytes of RAM and no substantial exterior system load.

$$\tau(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}\tau_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}\tau_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}\tau_3. \qquad \text{(A.2)}$$

In this expression we have values of τ at locations x_1 , x_2, and x_3. Differentiating with respect to x and evaluating the expression at x_1 yields

$$\frac{\partial\tau(x_1)}{\partial x} = \frac{(x_1-x_2)+(x_1-x_3)}{(x_1-x_2)(x_1-x_3)}\tau_1 + \frac{(x_1-x_3)}{(x_2-x_1)(x_2-x_3)}\tau_2 + \frac{(x_1-x_2)}{(x_3-x_1)(x_3-x_2)}\tau_3, \quad \text{(A.3)}$$

assuming that $x_1 = a$, $x_2 = a - \Delta x$, and $x_3 = a - 2\Delta x$ (equal spacing between sampled points) we can evaluate to obtain the second-order difference operator used in our method,

$$\frac{\partial\tau(a)}{\partial x} = \frac{\Delta x + 2\Delta x}{2(\Delta x)^2}\tau_1 + \frac{2\Delta x}{(-\Delta x)(\Delta x)}\tau_2 + \frac{\Delta x}{(-2\Delta x)(-\Delta x)}\tau_3 = \frac{3\tau_1 - 4\tau_2 + \tau_3}{2\Delta x} \qquad \text{(A.4)}$$

1. D. Aldridge. Linearization of the eikonal equation. *Geophysics*, 59(19), 1994.
2. C.D. Belfi and W.W. Symes. An adaptive eno algorithm for the eikonal equation. *Annual Report 1998, TRIP (The Rice Inversion Project)*, 1998.
3. T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction To Algorithms.* McGraw Hill and MIT Press, 1990.
4. J.H. Ferziger and M. Peric. *Computational Methods For Fluid Dynamics.* Springer Verlag, 1996.

5. S. Fomel. Traveltime calculation with the linearized eikonal equation. *Technical Report: Stanford Exploration Project (SEP)*, 94, 1997.
6. M.H. Holmes. *Introduction To Perturbation Methods*. Springer Verlag, 1995.
7. P. Podvin and I. Lecomte. Finite difference computation of traveltimes in very contrasted media: a massively parallel approach and its associated tool. *Geophysical Journal International*, 105, 1991.
8. F. Qin, Y. Luo, K. Olsen, W. Cai, and G. Schuster. Finite-difference solution of the eikonal equation along expanding wavefronts. *Geophysics*, 57, 1992.
9. M. Reshef and D. Kosloff. Migration of common shot gathers. *Geophysics*, 51:324–331, 1986.
10. S.Cao and S.Greenhalgh. Finite-difference solution of the eikonal equation using an efficient, first-arrival, wavefront tracking scheme. *Geophysics*, 59, 1994.
11. A. Sei and W. Symes. Convergent finite-difference traveltime gradient for tomography. *Proceedings Of The 65th Annual SEG Meeting*, pages 1258–1261, 1995.
12. J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Of Sci.*, 93:1591–1595, 1996.
13. J.A. Sethian and A. Mihai Popovici. 3-d traveltime computation using the fast marching method. *Geophysics*, 64(2):516–523, 1999.
14. S.Kim and R.Cook. 3d traveltime computation using second-order eno scheme. *Annual Report 1998, TRIP (The Rice Inversion Project)*, 1998.
15. W. Symes, R. Versteeg, and A. Sei. Kirchoff simulation, migration, and inversion using finite difference traveltimes and amplitudes part i: Eikonal and transport solvers. *Technical Report: The Rice Inversion Project (TRIP)*, 1995.
16. J. van Trier and W.W. Symes. Upwind finite-difference calculation of traveltimes. *Geophysics*, 56, 1991.
17. J. Vidale. Finite-difference calculation of travel times. *Bull. Seis. Soc. Of America*, 78:2062–2076, 1988.
18. J. E. Vidale and H. Houston. Rapid calculation of seismic amplitudes. *Geophysics*, 55(11):1504–1507, 1990.
19. Lin Zhang. Imaging by the wavefront propogation method. *Phd. Thesis, Stanford Univesity, SEP*, 1993.