Fast-marching eikonal solver in the tetragonal coordinates

Yalei Sun and Sergey Fomel, Stanford University*

SUMMARY

Accurate and efficient traveltime calculation is an important topic in seismic imaging. We present a fast-marching eikonal solver on the tetragonal coordinates (3-D) and trigonal coordinates (2-D), *tetragonal (trigonal) fast-marching eikonal solver* (TFMES), which can significantly reduce the first-order approximation error without greatly increasing the computational complexity. In the trigonal coordinates, there are six equally-spaced points surrounding one specific point and the number is twelve in the tetragonal coordinates, whereas the numbers of points are four and six respectively in the Cartesian coordinates. This means that the local traveltime updating space is more densely sampled in the tetragonal (or trigonal) coordinates, which is the main reason that TFMES is more accurate than its counterpart in the Cartesian coordinates. Compared with the fast-marching eikonal solver in the polar coordinates, TFMES is more convenient since it needs only to transform the velocity model from the Cartesian to the tetragonal coordinates for one time. Potentially, TFMES can handle the complex velocity model better than the polar fast-marching solver. We also show that TFMES can be completely derived from Fermat's principle. This variational formulation implies that the fast-marching method can be extended for traveltime computation on other nonorthogonal or unstructured grids.

INTRODUCTION

Traveltime map generation is a computationally expensive step in 3-D Kirchhoff depth imaging. Most approaches proposed are either based on ray tracing equations or on eikonal equation (Červený, 1987; Beydoun and Keho, 1987; Vidale, 1990; van Trier and Symes, 1991). Popovici and Sethian (1997) proposed a fast-marching finite-difference eikonal solver in the Cartesian coordinates, which is very efficient and stable. The efficiency is based on the heap-sorting algorithm. A similar idea has been used previously by Cao and Greenhalgh (1994) in a slightly different context. The remarkable stability of the method results from a specifically chosen order of finite-difference evaluation, which resembles the method used by Qin et al. (1992).

Alkhalifah and Fomel (1997) implemented the fast-marching algorithm in the polar coordinates, which is more accurate than its implementation in the Cartesian coordinates. However, the polar implementation requires velocity to be transformed from the Cartesian to the polar coordinates for each source location, which makes it inefficient. The spatial variation of grid size in the polar coordinates also makes it more difficult to handle strong velocity variation.

We present a new scheme based on the tetragonal eikonal equation. Because of the specialty of the tetragonal coordinates we have chosen, this new algorithm is more accurate than the Cartesian implementation. Meanwhile, it is free of the problems associated with the polar implementation.

We first derive the tetragonal eikonal equation and explain why it is more accurate than the Cartesian fast-marching eikonal solver. Then we show how to derive the same approach from Fermat's principle using a variational formulation, which is important for extending the fast-marching method to unstructured grids. We present 2-D and 3-D results, from simple to complex models, to support our explanation.

TETRAGONAL EIKONAL EQUATION

In the 3-D Cartesian coordinates, the eikonal equation is expressed as

$$\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial y}\right)^2 + \left(\frac{\partial t}{\partial z}\right)^2 = s^2, \tag{1}$$

where t stands for traveltime and s for slowness. The 2-D counterpart is given by omitting one term from the above equation.

$$\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial z}\right)^2 = s^2. \tag{2}$$

The Cartesian expressions have no crossing terms because of the orthogonality. If we define a tetragonal coordinate which has a transform relation (3) with the Cartesian coordinates (Figure 1)

$$\left\{ \begin{array}{lcl} x & = & \frac{\sqrt{6}}{3}u \\ y & = & \frac{\sqrt{3}}{6}u + \frac{\sqrt{3}}{2}v \\ z & = & \frac{1}{2}u + \frac{1}{2}v + w \end{array} \right. \tag{3}$$

and then substitute equation (3) into (1) and (2), we can get the following eikonal equation in the tetragonal (3-D) and trigonal (2-D) coordinates.

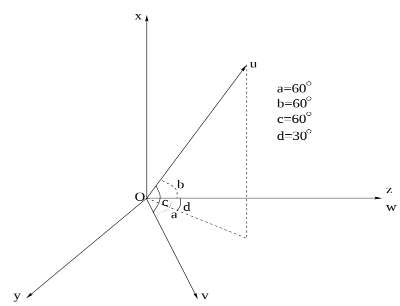

Figure 1: The tetragonal coordinates used in this paper. It reduces to the trigonal coordinates by omitting axis **u**.

$$\frac{3}{2}\left(\frac{\partial t}{\partial u}\right)^2 + \frac{3}{2}\left(\frac{\partial t}{\partial v}\right)^2 + \frac{3}{2}\left(\frac{\partial t}{\partial w}\right)^2 - \left(\frac{\partial t}{\partial u}\right)\left(\frac{\partial t}{\partial v}\right) - \left(\frac{\partial t}{\partial u}\right)\left(\frac{\partial t}{\partial w}\right) - \left(\frac{\partial t}{\partial v}\right)\left(\frac{\partial t}{\partial w}\right) = s^2 \tag{4}$$

$$\frac{4}{3}\left(\frac{\partial t}{\partial v}\right)^2 + \frac{4}{3}\left(\frac{\partial t}{\partial w}\right)^2 - \frac{4}{3}\left(\frac{\partial t}{\partial v}\right)\left(\frac{\partial t}{\partial w}\right) = s^2. \tag{5}$$

The fast-marching algorithm is an upwind first-order discretization of the above eikonal equation. In the next section, we show that it reduces to solving a quadratic equation. The key feature of this algorithm is a carefully selected order of traveltime evaluation. In the Cartesian coordinates, each point with known traveltime can update four equally-spaced neighboring points in 2-D and six in 3-D, as shown in Figures 3 and 4. In the trigonal and tetragonal coordinates, these two numbers are six and twelve respectively, as shown in Figures 5 and 6. Since the fast-marching eikonal solver is based on the plane wave assumption, more equally-spaced neighboring points mean a better approximation to the assumption. Therefore, TFMES should be more accurate than its Cartesian counterpart.

Alkhalifah and Fomel (1997) have shown that the fast-marching algorithm in the polar coordinates is also more accurate than the Cartesian implementation. The reason is that the circle (2-D) or sphere (3-D) axis in the polar coordinates closely matches the wavefront when the media are relatively smooth. However, the polar implementation needs to transform the velocity model from the Cartesian to the polar coordinates for each single source, which makes it inconvenient. The grid size in the polar coordinates becomes larger and larger with the increase of radius. Therefore, some of the detailed velocity variation can be missed easily. Free of these problems, TFMES is more flexible and efficient than the polar implementation.

VARIATIONAL FORMULATION OF THE FAST-MARCHING ALGORITHM

The fast-marching algorithm consists of two parts: minimum traveltime selection and a local traveltime updating scheme. The selection scheme is essentially based on Fermat's principle. The local traveltime updating scheme can also be derived from the same principle using a local linear interpolation, which provides the first-order accuracy. For simplicity, let us focus on the 2-D case. Consider a line segment with the end points A and B, as shown in Figure 2. Let t_A and t_B denote the traveltimes from a fixed distant source to points A and B, respectively. Define a parameter ξ such that $\xi = 0$ at A, $\xi = 1$ at B, and ξ changes continuously on the line segment between A and B. Then for each point of the segment, we can approximate the traveltime by the linear interpolation formula

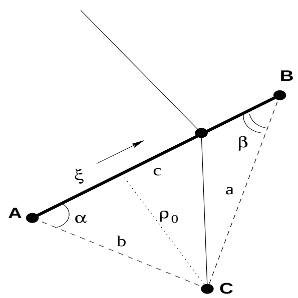

Figure 2: A geometrical scheme for the traveltime updating procedure in two dimensions.

$$t(\xi) = (1-\xi)t_A + \xi t_B \ . \tag{6}$$

Now let us consider an arbitrary point C in the vicinity of AB. If we know that the ray from the source to C passes through some point ξ of the segment AB, then the total traveltime at C is approximately

$$t_C = t(\xi) + s_C \sqrt{|AB|^2(\xi-\xi_0)^2 + \rho_0^2} \ , \tag{7}$$

where s_C is the local slowness, ξ_0 corresponds to the projection of C to the line AB (normalized by the length $|AB|$), and ρ_0 is the length of the normal from C to ξ_0.

Fermat's principle states that the actual ray to C corresponds to a local minimum of the traveltime with respect to raypath perturbations. According to our parameterization, it is sufficient to find a local extreme of t_C with respect to the parameter ξ. Equating the ξ derivative to zero, we arrive at the equation

$$t_B - t_A + \frac{s_C\,|AB|^2\,(\xi-\xi_0)}{\sqrt{|AB|^2(\xi-\xi_0)^2+\rho_0^2}} = 0 \ , \tag{8}$$

which contains (as a quadratic equation) the explicit solution for ξ:

$$\xi = \xi_0 \pm \frac{\rho_0\,(t_A - t_B)}{|AB|\,\sqrt{s_C^2\,|AB|^2 - (t_A-t_B)^2}} \ . \tag{9}$$

Finally, substituting the value of ξ from (9) into equation (7) and selecting the appropriate branch of the square root, we obtain the formula

$$c\,t_C = \rho_0\,\sqrt{s_C^2 c^2 - (t_A-t_B)^2} + a\,t_A\,\cos\beta + b\,t_B\,\cos\alpha \ , \tag{10}$$

where $c = |AB|$, $a = |BC|$, $b = |AC|$, angle α corresponds to $\widehat{BAC}$, and angle β corresponds to $\widehat{ABC}$ in the triangle ABC (Figure 2).

The above general procedure can be greatly simplified when applied to some regular grids, such as the rectangular grid or the tetragonal grid. This expression is even valid for unstructured grids. As pointed out by Fomel (1997), unstructured grids have some attractive computational advantages over regular ones. Moreover, the derivation provides a general principle, which can be applied to derive analogous algorithms for other eikonal-type (Hamilton-Jacobi) equations and their corresponding variational principles.

NUMERICAL RESULTS

We implement TFMES in both 2-D and 3-D cases. 3-D Constant velocity medium is used as a benchmark to verify the accuracy of the new algorithm. In order to make a fair comparison, we use the same sampling interval in both the Cartesian and the tetragonal coordinates. As shown in Figure 7, the Cartesian implementation tends to over-estimate the traveltime in the diagonal direction, while the tetragonal result matches the analytical result very accurately. More complex Marmousi and SEG/EAGE saltdome models are used to test its stability when handling complex models. Figure 8 is the test of 2-D Marmousi model. The source is located on the surface at coordinates (x=4100m, z=0m). In most areas, the two results match each other. When passing through the complex structure in the middle, they begin to deviate from each other. The trigonal result is not as smooth as the Cartesian result. This is because the trigonal result has six neighboring points instead of four points as in the Cartesian coordinates, which makes it more capable of simulating complicate wavefronts. The Cartesian implementation over-estimates the traveltime compared with the trigonal result, which is similar to the conclusion reached by Alkhalifah and Fomel (1997). Figures 9 and 10 show two traveltime slices from the SEG/EAGE saltdome model. The source is located at coordinates (x=7200m, y=7320m, z=1680m). Figure 9 was obtained with a constant depth z=1560m, while Figure 10 is extracted in the diagonal direction. In general, the Cartesian implementation tends to over-estimate the traveltime in more areas than the tetragonal implementation.

CONCLUSION

In this paper, we extend the fast-marching eikonal solver from the Cartesian coordinates to the tetragonal coordinates. Compared with the Cartesian implementation, the tetragonal (trigonal) fast-marching eikonal solver can reduce the first-order approximation error. It is also more efficient than the polar implementation. Following the plane wave assumption, we derive the same algorithm using Fermat's principle, which implies that it is possible to extend the algorithm to unstructured grids (triangle in 2-D; tetrahedron in 3-D model).

REFERENCES

Alkhalifah, T., and Fomel, S., 1997, Implementing the fast marching eikonal solver: Spherical versus cartesian coordinates: SEP–**95**, 149–171.

Beydoun, W. B., and Keho, T. H., 1987, The paraxial ray method: Geophysics, **52**, no. 12, 1639–1653.

Cao, S., and Greenhalgh, S. A., 1994, Finite-difference solution of the eikonal equation using an efficient, first-arrival wavefront tracking scheme: Geophysics, **59**, no. 4, 632–643.

Červený, V., 1987, Ray tracing algorithms in three-dimensional laterally varying layered structures *in* Nolet, G., Ed., Seismic Tomography:: Riedel Publishing Co., 99–134.

Fomel, S., 1997, A variational formulation of the fast marching eikonal solver: SEP–**95**, 127–147.

Popovici, A. M., and Sethian, J. A., 1997, Three dimensional traveltime computation using the fast marching method: 67th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1778–1781.

Qin, F., Luo, Y., Olsen, K. B., Cai, W., and Schuster, G. T., 1992, Finite-difference solution of the eikonal equation along expanding wavefronts: Geophysics, **57**, no. 3, 478–487.

van Trier, J., and Symes, W. W., 1991, Upwind finite-difference calculation of traveltimes: Geophysics, **56**, no. 6, 812–821.

Vidale, J. E., 1990, Finite-difference calculation of traveltimes in three dimensions: Geophysics, **55**, no. 5, 521–526.

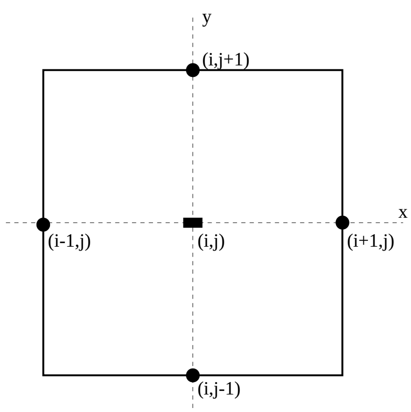

Figure 3: 2-D updating scheme in the Cartesian coordinates. Traveltime at point (i, j) is known and four equally-spaced neighboring points' traveltimes are candidates for updating.

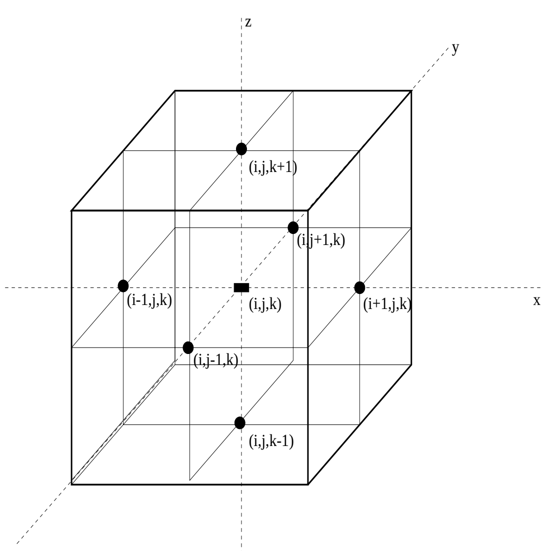

Figure 4: 3-D updating scheme in the Cartesian coordinates. Traveltime at point (i, j, k) is known and six equally-spaced neighboring points' traveltimes are candidates for updating.

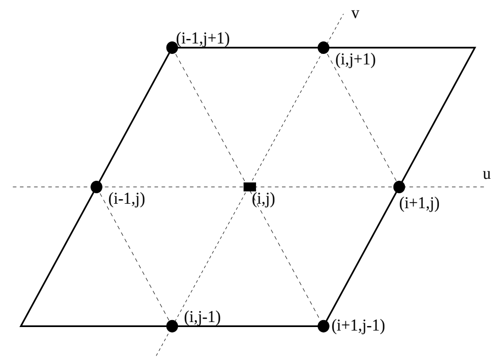

Figure 5: 2-D updating scheme in the trigonal coordinates. Traveltime at point (i, j) is known and six equally-spaced neighboring points' traveltimes are candidates for updating.

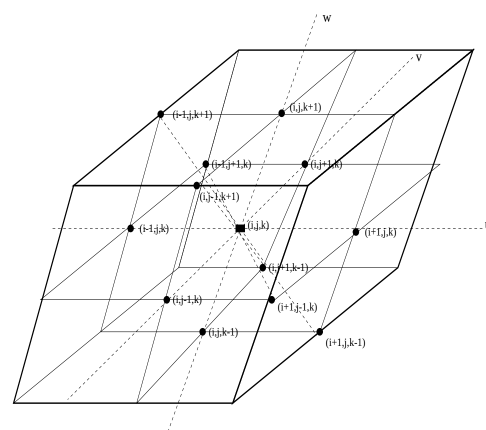

Figure 6: 3-D updating scheme in the tetragonal coordinates. Traveltime at point (i, j, k) is known and twelve equal-spaced neighboring points' traveltimes are candidates for updating.

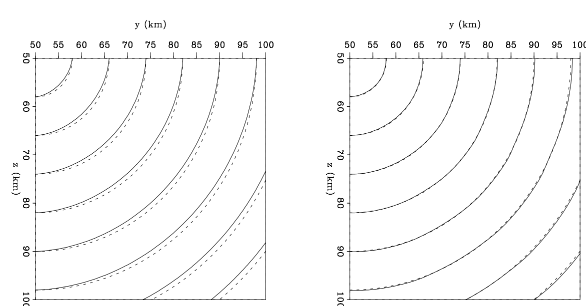

Figure 7: **Left**: Cartesian. **Right**: tetragonal. Traveltime slice from a 3-D constant velocity model. The source is located at the upper-left corner and the spatial sampling interval is 1 km in all the three directions. The dash line represents the analytical solution. The solid line on the left stands for the Cartesian implementation and the solid line on the right for the tetragonal case. The Cartesian result has large errors in the diagonal direction. The tetragonal result matches the analytical result very accurately.

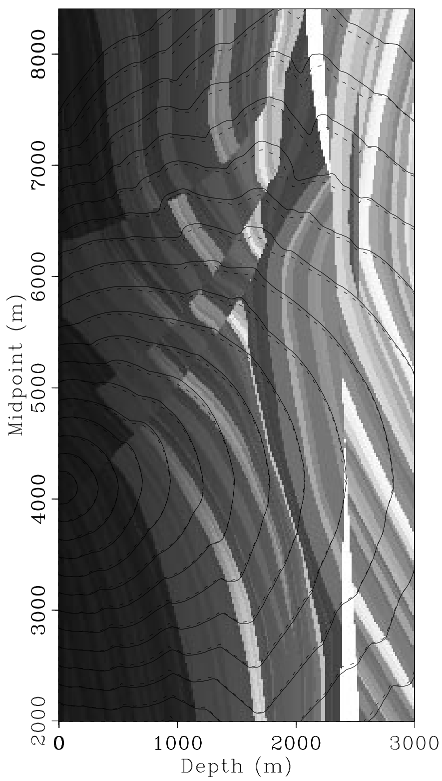

Figure 8: Traveltime slice of 2-D Marmousi model. Solid line stands for the tetragonal result and dash line for the Cartesian result. The Cartesian implementation tends to over-estimate the traveltime.

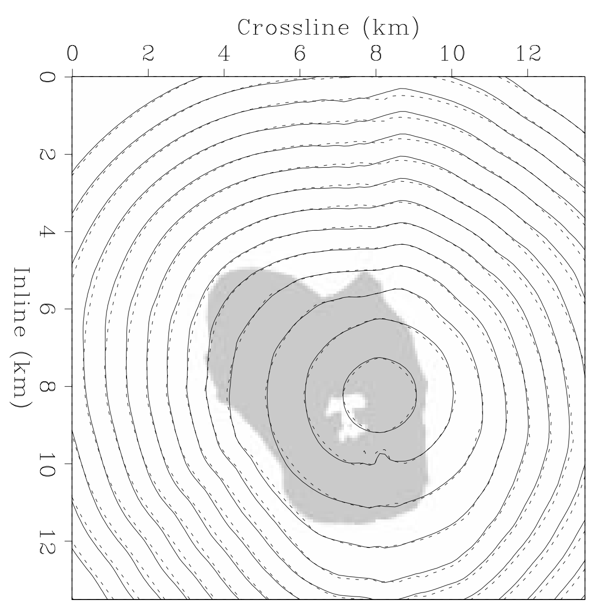

Figure 9: Traveltime slice of the 3-D SEG/EAGE saltdome model for a constant depth $z = 1560m$. The solid line represents the tetragonal result and the dash line stands for the Cartesian result. For the most part, the Cartesian result tends to over-estimate the traveltime.

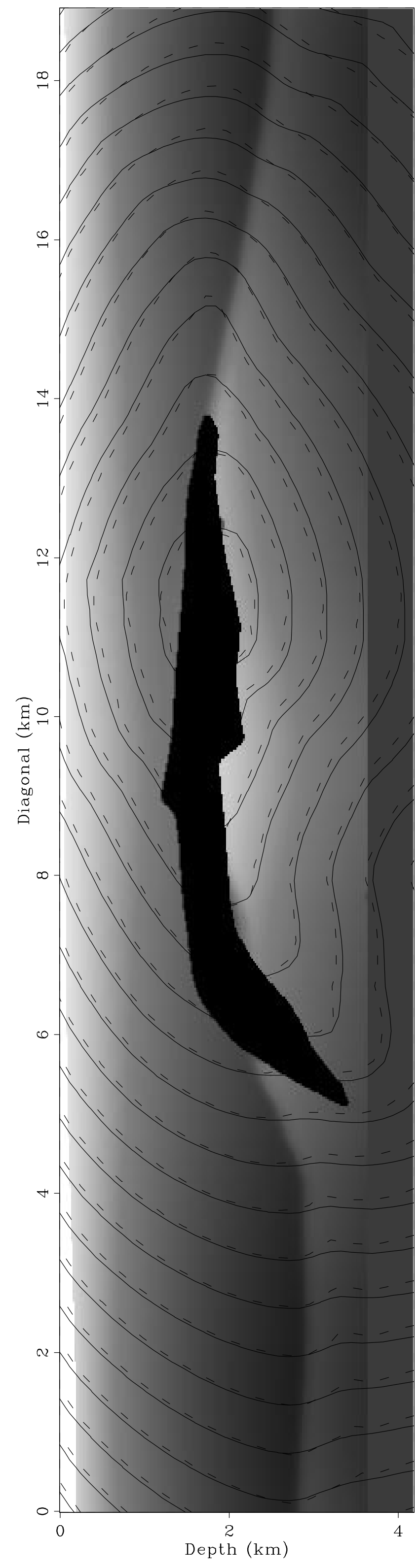

Figure 10: Traveltime slices of 3-D SEG/EAGE saltdome model in the diagonal direction.