Wavefront construction (WF) ray tracing in tetrahedral models –Application to 3-D traveltime and ray path computations

Zhaobo Meng and Norman Bleistein
Center for Wave Phenomena Colorado School of Mines

ABSTRACT

For the purpose of building a fast and accurate tool for the computation of traveltimes and ray paths for 3-D depth imaging, we combine the techniques of tetrahedral model representation and 3-D wavefront construction (WF) ray tracing. The scheme is robust and efficient in the presence of complex earth structures.

To efficiently represent a 3-D complex earth structure, a 3-D tetrahedral model as well as an auxiliary Cartesian cubic grid is used. In each layer, the "sloth" (the square of the slowness) is required to vary continuously, and specifically, in each cell, the sloth gradient is constant. The layers are separated by smoothly varying horizons. The tetrahedral model representation is based on the triangulation of the smooth horizons.

WF ray tracing is used in the scheme. In WF ray tracing, rays are traced stepwise in traveltime through the model and are maintained by a *triangular network*, (i) to form the wavefront at each traveltime step, (ii) to evaluate quantities, such as traveltimes and ray paths at the receivers, and (iii) to interpolate new rays, whenever certain criteria are met. The quantities, such as traveltimes and ray paths, evaluated on a Cartesian cubic grid, are useful for depth imaging. For our kinematic ray tracing, which gives correct traveltimes and ray paths, an efficient alternative for dealing with caustics is presented.

Key words: Tetrahedral model representation, WF ray tracing, traveltime and ray path computations

Introduction

Computation of traveltimes and ray paths is essential to many 3-D seismic depth imaging processes: Kirchhoff prestack and poststack migration/inversion, migration velocity analysis, tomography, and Kirchhoff datuming. In this report, we discuss the approach to combining a tetrahedral earth model representation and wavefront construction (WF) ray tracing for the purpose of building a fast and accurate tool for computing traveltimes and ray paths for 3-D depth imaging.

The 2-D WF ray tracing was first introduced by Vinje *et al.* (1993). Chilcoat and Hildebrand (1995) and Vinje *et al.* (1996a) (1996b) extended the 2-D algorithm to 3-D, using Cartesian grid models or models with triangular networked interfaces (Vinje *et al.*, 1996a). The WF ray tracing algorithm is robust, in part because it maintains a reasonably consistent low ray density without making *a priori* estimates of the number of rays needed. This is attractive in complex 3-D applications, such as sub-salt imaging. The rays are maintained by a triangular network, such that the representation of wavefronts, interpolation of new rays, and evaluation of quantities at receivers, become fairly simple and efficient.

On the other hand, tetrahedral model representation (Albertin & Wiggins, 1994), including model building with triangular networked interfaces (Vinje *et al.*, 1996a) and (non-WF) ray tracing in tetrahedral models

(Stankovic & Albertin, 1995), have shown high efficiency in representing complex models with a reasonable number of cells when compared to Cartesian grid modeling. For a complex model, where 3-D imaging is necessary, a Cartesian grid model usually gives a poor and inefficient description of the rays and wavefronts. As a result, for Cartesian grid models, fine grids have to be used and excessive smoothing must be applied. This leads to higher memory cost and/or inaccurate estimates of quantities (such as traveltimes, and, even worse, for ray paths) for 3-D complex models. It also leads to lower resolution in the imaging. Hence, tetrahedral models are needed for fast and efficient ray tracing and wavefront construction.

In the following sections we describe highlights of our approach to developing a tetrahedral model builder and WF construction ray tracing scheme.

Tetrahedral Representation of 3-D Complex Models

In this section, we discuss how earth models are built. 3-D ray tracing is not the only process that requires efficient representation of an earth model. Earth models also play an important role in other modern 3-D depth image processing, such as migration, datuming and tomography, that rely on accurate representation of geologic models to obtain good images.

Model building must combine geologic objects, such as *horizons* and *regions*. Horizons are surfaces that represent the medium discontinuities–in particular, velocity–and the regions are domains of continuous medium. Horizons and regions are two fundamental objects of 3-D models.

It has been well-known that the simplexes (segment in 1-D, triangles in 2-D and tetrahedra in 3-D) are the most efficient elements to construct complex structures. In this scheme, we will use triangles to represent horizons and wavefronts, and tetrahedra to represent regions between horizons.

Tetrahedra are robust in representation of velocity discontinuities across horizons. However, in regions of continuous velocity, it is Cartesian grids that are more efficient; calculations of intersections (by solving polynomial equations) of ray paths with tetrahedral faces are more costly than with cubic grids. Thus, in this approach, a tetrahedral model and a Cartesian cubic grid are both used. The tetrahedral model or the Cartesian grid can be accessed as needed, e.g. using the tetrahedral model in representing velocity discontinuities across horizons, and using the Cartesian grid in regions between horizons. This facilitates fast and accurate ray tracing.

We note that the capacity to represent the discontinuities of the model in 3-D depth imaging is important. The reason is that we use high frequency methods–ray methods–that are only valid if the physical parameters in the medium vary slowly over a wavelength; thus, they break down at discontinuities. The use of Snell's law across the discontinuities overcomes this problem.

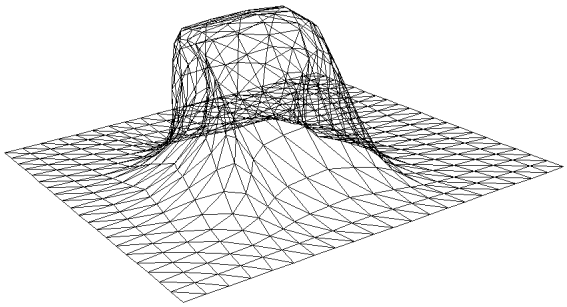

Figure 1. A salt-dome defined by three functions, $x(u,v)$, $y(u,v)$ and $z(u,v)$. Note that the grid spacing in x, y and z is related to the local curvature, and there are dips above and below 90^o.

Horizon triangulation

To build a tetrahedral model, we first need to triangulate the horizons. An efficient triangulation of a horizon should achieve a sufficiently precise representation of a complex horizon with a "cheapest" mesh.

To begin, we represent a horizon by three functions, $x = x(u,v)$, $y = y(u,v)$ and $z = z(u,v)$, of two parameters, u and v. Parameterizations are useful to represent a multi-valued horizon using single-valued functions.* Usually, u and v can be sampled uniformly. The discretization of the horizon in x, y and z should be dense enough to adequately describe the variation on the horizon, i.e., finer grids in x, y and z should be applied to areas with more rapid variations along the horizon. Then, by simply connecting all the grid points on the horizon that have the same parameter value u, and then, v, quadrilaterals are obtained. Finally, by cutting each quadrilateral into two triangles, the horizon is represented by triangles. As an example, Figure 1 is a salt-dome defined by such a parameterization. Note that, in the figure, there are dips both above and below 90^o. The grid spacing of the horizon in x, y and z is related to the curvatures, while x, y and z, in u and v, are uniformly sampled (the three $x(u,v)$, $y(u,v)$ and $z(u,v)$ grids are not shown in the figure).

Secondly, some auxiliary information about the horizons must be provided. For kinematic ray tracing in

* If the horizon is too complicated, more levels of parameterizations can be used, that is, to parameterize u and v as functions of new parameters, and so on.

this approach, we need the normal, a linear function of position, to each triangular tile. Providing such auxiliary information in advance will speed up the ray tracing calculations, compared to the case in which the information is computed as needed.

The smoothness requirement of the triangulated horizons

The smoothness of the triangulated horizons is important in ray tracing. The degree of smoothing that one uses is a compromise between the quality of the output and the cost of the computations. To correctly calculate the travel times and the ray paths, the normal, denoted by $\mathbf{n}$, should be continuous (i.e. $\mathbf{n} \in C^0$). This guarantees that the ray paths, as well as the traveltimes, vary continuously, depending on the variation of the horizons. If dynamic ray tracing or true amplitude evaluation is applied, the normals should be continuously differentiable (i.e. $\mathbf{n} \in C^1$); in other words, the horizons should be C^1 for travel time and ray path computation, and C^2 (continuously second order differentiable) for dynamic ray tracing (Cervény, 1987). This leads to different requirements for the orders of interpolation for the surface positions and the normals when applying R/T (reflection/transmission) laws in the ray tracing, in order to make the modeling well-conditioned to the degree necessary.

We note that, higher order of smoothing of horizons will be much more costly. For example, C^1-smooth horizons used in this approach are represented by piecewise quadratic polynomials. The ray coordinates in a linear sloth model are quadratic functions of the ray running parameter, σ (see Appendix A). Thus, solving for the intersection of a ray and a triangular tile on the horizon requires the solution of a 4th order polynomial equation.

An efficient scheme for solving for the intersection of a ray with a triangular tile starts from a close initial guess for the intersection. Assuming the ray is straight at first, we solve for the intersection of the straight ray with the smoothed triangular tile, which only requires solution of a quadratic equation. From this initial guess, the *four*th order polynomial equation can be quickly solved iteratively. Note that a C^2-smooth horizon (e.g. for dynamic ray tracing and true amplitude computations) requires the solutions of 6th order of polynomial equation. Although the same technique can be applied, solving a 6th order equation in this manner could be more costly and problematic.

Here, we limit our approach to C^1-smooth horizons, which gives correct traveltimes and ray paths. However, the higher order quantities, such as amplitudes may not be sufficiently accurately determined. Given a C^0-smooth horizon, we have to address the problem of numerically constructing a C^1-smooth horizon. Of course, there are different ways to do this. Our approach is to carry out the smoothing in three steps, as shown in Figure 2. Before the smoothing, the horizon is C^0-smooth, each triangular tile is planar. In the first step, we assign the normal to the planar triangular tile to its centroid. In the second step, the normal to each vertex is interpolated using the normals to the centroids of the nearest neighboring (usually of six) triangular tiles. The last step is to linearly interpolate the normal to the (shaded in Figure 2) triangular tile. The first two steps are carried out in model building, the last step is carried out during WF ray tracing. After this linear interpolation, the normal to the horizon becomes C^0-smooth. Of course, the linearly interpolated normals are continuous across the boundaries of the triangles. Once a piece-wise linear C^0 normal function is defined, a piece-wise quadratic C^1 triangulated horizon is a simple integration of the normal over the space.

Tetrahedral regions

A simplified tetrahedral region between two triangulated horizons can be obtained by straight line connections of the corresponding nodes (with the same indices in u and v) on the two triangulated horizons, e.g., as shown in Figure 3. A pair of triangles (shaded), one from each horizon, makes three tetrahedra by the following steps. First, we connect the apices of the triangles with straight lines. Then, two triangular tiles are made by cutting through apices 1-5-6 and 1-5-3. Finally, three tetrahedra are formed and named by apices 1-5-2-3, 1-4-5-6 and 1-5-3-6. The shortcoming of this simplification is that the number of nodes in each parameter (u or v) on all horizons must be the same. The advantage is that the tetrahedra and the triangulated horizons can be stored using "arrays" that are more memory and computation efficient than "lists" that are used otherwise. So, again, this is a compromise between the computation cost and the quality. In Figure 4, a tetrahedral layer is formed by connecting the two triangulated horizons. In Figure 5, a lens is described by connecting the two triangulated horizons which come together near the edges of the figure.

WF Ray Tracing

Vinje, *et al.* (1993) (1996a) (1996b), Chilcoat and Hildebrand (1995) introduced 2-D and 3-D WF ray tracing algorithms, either on Cartesian grids or models with triangular interfaces. Perhaps the most attractive feature of WF ray tracing is that, it interpolates new rays whenever certain criteria are met, such as, when the triangular wavefront section determined by its three

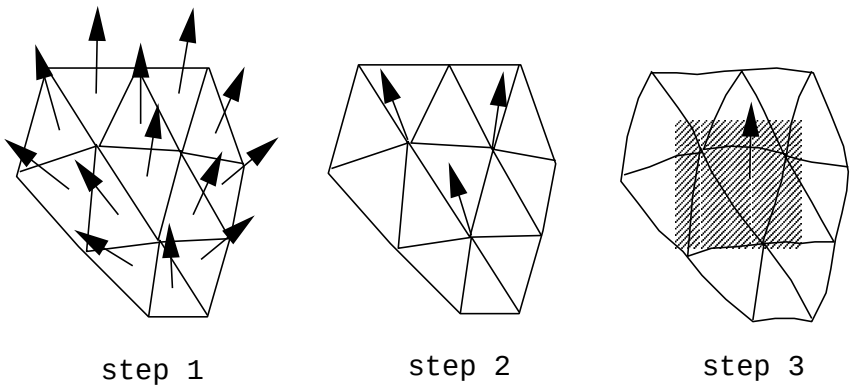

Figure 2. The three steps to smooth the normals to C^0-smooth (and thus the horizons to C^1-smooth). Step 1, assign the normals to each centroid; step 2, linearly interpolate the normals to the vertexes using the normals to the centroids of the (six in general) neighboring triangular tiles; and step 3, interpolate the normal to the triangular tile (shaded) using the normals to the three vertexes. C^1- horizon is obtained by integrating the C^0- normals over the space. After the smoothing, the edges of each triangles are quadratic polynomials (see step 3).

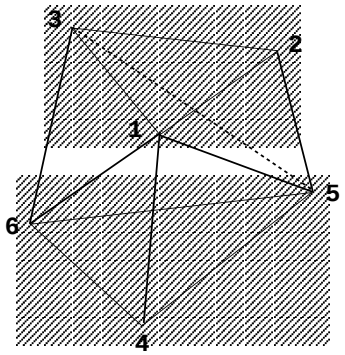

Figure 3. Three tetrahedra (1-5-2-3, 1-4-5-6 and 1-5-3-6) defined by a triangle from top horizon (shaded) and the other from the base horizon (shaded).

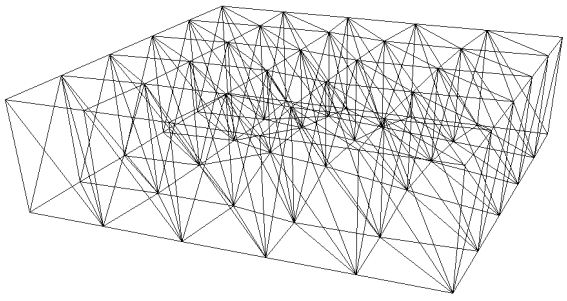

Figure 4. A tetrahedral layer defined by connecting two non-uniformly sampled triangulated horizons.

neighboring rays is too large, or the opening angle of its three neighboring rays is too wide. The WF algorithm also overcomes the problem of *artificial* shadow zones in conventional ray tracing. These arise because conventional ray tracing is initiated by a bundle of rays from a

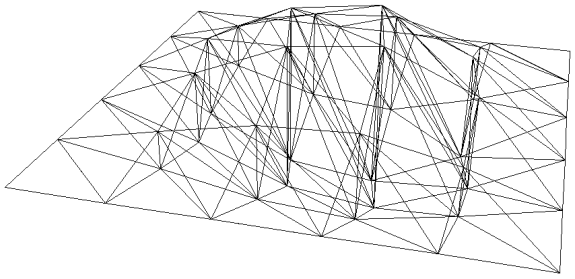

Figure 5. A tetrahedral lens defined by two horizons, which come together near the edges.

source point with angular distribution assigned *a priori*. Thus may cause the lack of arrivals for large geometrical spreading in non-shadow zones, thus creating the so called artificial shadow zones.

Propagation of wavefronts

WF ray tracing is typically done shot by shot. The basic idea is, for each shot, given the source position, at $t = 0$, for each node (i.e. the intersections between rays and wavefronts), initialize the slowness vectors and quantities such as traveltimes and ray paths, wave code (P or S) if the medium is not acoustic, etc.; then start to propagate the wavefronts. For each time step, for each ray, until the time increment Δt is reached, the ray is traced forward by increasing the parameter σ, using analytic formulas (see Appendix A). It is checked if this analytic ray intersects with any horizons within the small traveltime increment Δt. If it does, the R/T laws are applied to adjust the slowness vectors, ray and wavefront variables

at the intersections. When the Δt increment is reached for all rays, the wavefront is updated to the new position. The wavefronts for two consecutive time steps are kept in memory to facilitate interpolation of new rays and estimation of quantities at receivers.

Triangular networking of the rays and wavefronts

The above are the main steps. We did not address, here, how to numerically represent and maintain a wavefront. The standard 3D WF ray tracing scheme ((Chilcoat & Hildebrand, 1995), (Vinje *et al.*, 1996a) and (Vinje *et al.*, 1996b)) uses triangular networks connecting the rays on a 3-D wavefront, which makes it fairly simple for the interpolation of new rays and evaluation of quantities at receivers.

For example, Figure 6 gives a view of the triangular networked wavefront. At each time step, a wavefront consists of triangular tiles, the triangular tile is determined by its three neighboring nodes/rays. The three rays forming the triangular tile were the nearest (both in take-off angle and the azimuth) when they were shot off at $t = 0$ from the source, but not necessarily the nearest at any other traveltime $t \neq 0$.

At each time step, the distance and the angular difference of tangents between any two rays bound by the triangle are checked. If the distance or the angular difference of tangents become larger than a predefined quantity, a new ray must be added, and thus a new triangle is added as well, as shown in Figure 7. The new ray is interpolated using the three rays bound by that triangle. Note that both the distance and the angular difference of ray tangents increase unless the rays approach a caustic. The triangle and its three rays coexist, in a way that, when one ray among them goes out out of the model or runs out of the maximum traveltime, this triangle is eliminated.

For many applications, it is the quantities (such as traveltime and ray paths) that are needed. The quantities at a receiver $\mathbf{r}$ which falls in, or very close to, the triangular prism (Figure 8) can be obtained by evaluation of the quantities at the three rays $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$.

Handling caustics for kinematic ray tracing

When a caustic exists, the traditional (dynamic) WF ray tracing removes the self-crossed part of the wavefront, and adjusts the wavefront at the triplication, so that the higher order quantities needed in dynamic ray tracing can be correct (Vinje *et al.*, 1993). This can be very costly and problematic in numerical application, especially for a 3-D complex model. For kinematic ray tra-

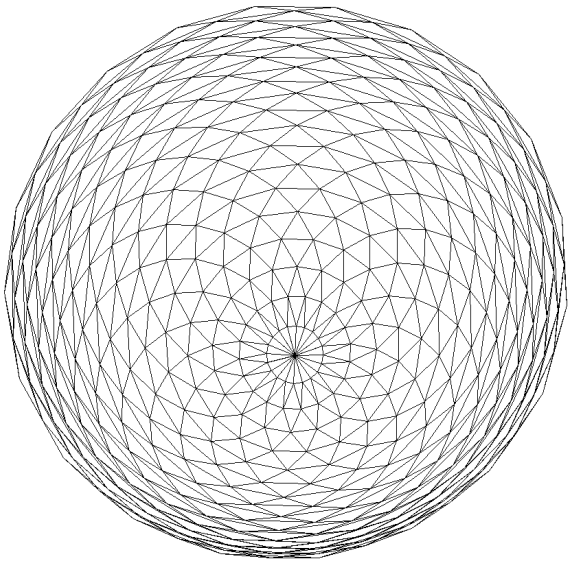

Figure 6. The wavefront at a certain time in a homogeneous medium. The wavefront consists of triangular tiles, each triangular tile is determined by its three neighboring nodes/rays.

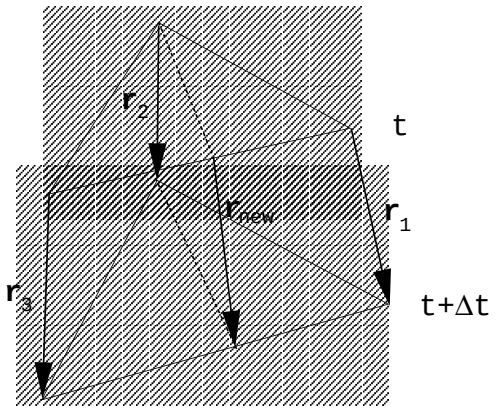

Figure 7. One new ray is added, and thus one triangle is added, as certain criteria are met.

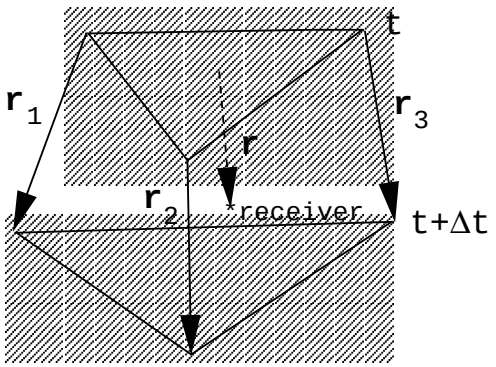

Figure 8. Quantities at a receiver $\mathbf{r}$ which falls in the triangular prism are interpolated using the three rays $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$.

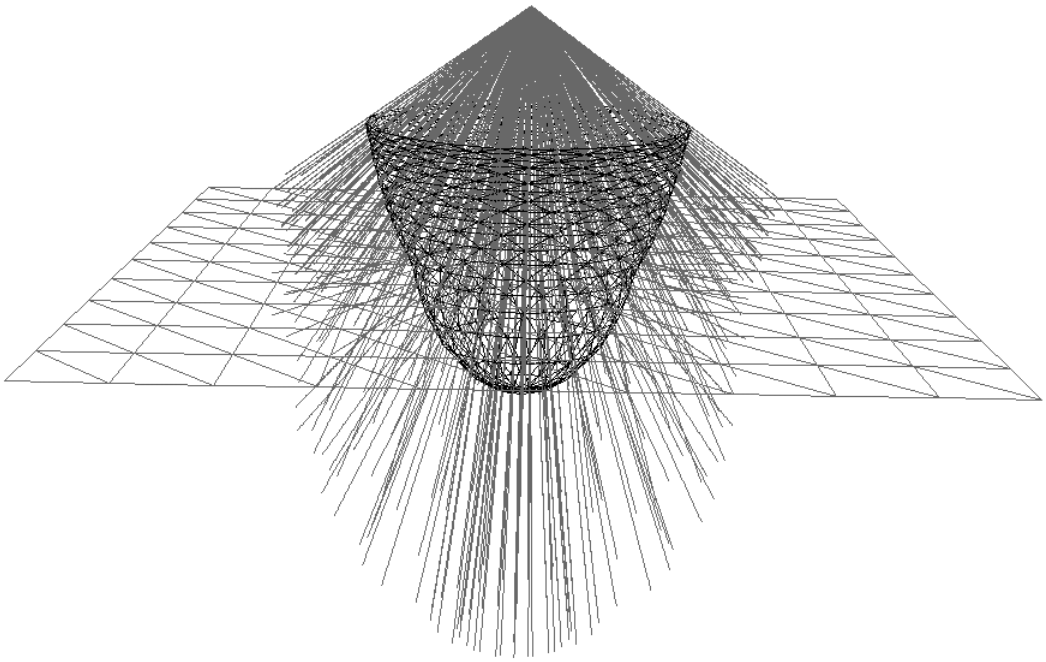

Figure 10. Ray tracing in a model of two layers. Velocity is 2km/s above the horizon and 3km/s below the horizon. The wavefront is distorted and the rays bend when crossing the horizon.

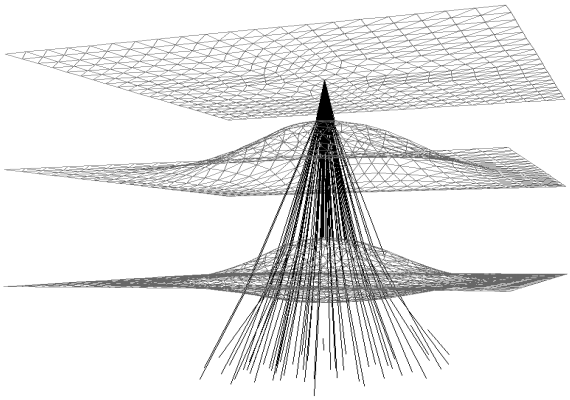

Figure 9. Ray tracing in a layered model consisting of 4 layers including a lens (the third layer). Some interpolated rays in the 4th layer can be seen.

cing, we present here an efficient alternative for dealing with caustics. We use the current wavefront (at time t) and the previous wavefront (at time $t - \Delta t$), to evaluate quantities (such as traveltime and ray paths) at receivers (usually a cubic grid). If quantities, such as traveltimes and ray paths, are evaluated more than once (which is easy to check numerically by setting up flags), only the one with smallest traveltime is kept; if the traveltime is the same, then check the ray path; only the one with the shortest ray path is kept.

Conclusions

For the purpose of fast and accurate traveltime and ray path computations, we have developed a tetrahedral model building algorithm and a WF ray tracing algorithm. The tool will be used for 3-D depth imaging, including prestack and poststack Kirchhoff migration, Kirchhoff datuming, tomography and tomographic migration velocity analysis, etc.

We have described here some of the features of the model building and the ray tracing algorithm. The tetrahedral model representation for complex models uses a reasonably small number of elements with more accuracy, as compared to Cartesian grids. The WF ray tracing maintains a reasonably consistent ray density by a triangular network. Thus, the combination of the two techniques is robust and efficient.

Several different models have been built for each of which ray tracing has been carried out. A 3-D prestack Kirchhoff migration algorithm using the ray tracer based on this report is currently in testing mode.

Acknowledgments

The first author is grateful to Fanlin Meng (Utah Tomography and Modeling/Migration Consortium) for useful discussions and Jérôme Le Rousseau (visiting CWP from

Elf Aquitaine Production) for suggestions about an efficient interpolation scheme within a triangular prism.

References

Albertin, Uwe K., & Wiggins, Wendell. 1994. Embedding geologic horizon surfaces in tetrahedral meshes for geologic modeling. *64th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts*, **94**, 502–505.

Cervény, V. 1987. Raytracing algorithms in three-dimensional laterally varying layered structures. *Nolet, G., Ed., Seismic Tomography*, 99–133.

Chilcoat, Steven R., & Hildebrand, Steve T. 1995. Wavefront construction in 3-D. *65th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts*, **95**, 1255–1257.

Stankovic, Goran M., & Albertin, Uwe K. 1995. Raytracing in topological tetrahedral models. *65th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts*, **95**, 1247–1250.

Vinje, V., Iversen, E., & Gjöystdal, H. 1993. Traveltime and amplitude estimation using wavefront construction. *Geophysics*, **58**(8), 1157–1166.

Vinje, V., Iversen, E., Astebol, K., & Gjöystdal, H. 1996a. Estimation of multivalued arrivals in 3D models using wavefront contruction–Part I. *Geophysical Prospecting*, **44**, 819–842.

Vinje, V., Iversen, E., Astebol, K., & Gjöystdal, H. 1996b. Estimation of multivalued arrivals in 3D models using wavefront contruction–Part II. *Geophysical Prospecting*, **44**, 843–858.

APPENDIX A: Formulations for linear sloth model

In this appendix, we discuss the analytic ray tracing solutions for linear sloth model. Consider the 3-D acoustic wave eikonal equation,

$$\mathbf{p}^2 = s, \tag{A1}$$

where $\mathbf{p} = \mathbf{p}(\mathbf{x}, \sigma)$ is the slowness vector, $s(\mathbf{x}) = 1/v^2(\mathbf{x})$ is the sloth, and $v(\mathbf{x})$ is the propagation speed. We use σ as the ray tracing parameter, which allows us to represent the ray path as a simple quadratic function (A8), in a linear sloth model. Then the ray tracing system can be written as,

$$\frac{d\mathbf{x}}{d\sigma} = \mathbf{p}, \tag{A2}$$

$$\frac{d\mathbf{p}}{d\sigma} = \frac{1}{2}\nabla s, \tag{A3}$$

$$\frac{dt}{d\sigma} = s. \tag{A4}$$

The initial values are the starting point $\mathbf{x}_0$ and the initial time t_0 at initial value of the ray parameter $\sigma = 0$. In addition

$$\mathbf{p}(\mathbf{x}_0, \sigma = 0) = \mathbf{p}_0. \tag{A5}$$

As mentioned above, we assume the sloth to be linear in each cell and we define

$$U = s(\mathbf{x}_0), \tag{A6}$$

$$\mathbf{A} = \nabla s(\mathbf{x}) = \nabla s(\mathbf{x}_0), \tag{A7}$$

where $\mathbf{A}$ is a constant vector in each cell. The ray tracing problem can now be substantially simplified. Because sloth $s(\mathbf{x})$ is the only function appearing in the formulas, the solution can be written as,

$$\mathbf{x}(\sigma) = \mathbf{x}_0 + \sigma\mathbf{p}_0 + \frac{\sigma^2}{4}\mathbf{A}, \tag{A8}$$

$$\mathbf{p}(\sigma) = \mathbf{p}_0 + \frac{\sigma}{2}\mathbf{A}, \tag{A9}$$

$$t(\sigma) = t_0 + \sigma U + \frac{\sigma^2}{2}\mathbf{A}\cdot\mathbf{p}_0 + \frac{\sigma^3}{12}\mathbf{A}\cdot\mathbf{A}. \tag{A10}$$

Here $\mathbf{x}_0$, $\mathbf{p}_0$ and t_0 are determined by continuity (or R/T) conditions applied to the incident and transmitted wavefronts.

APPENDIX B: Best fit for linear velocity to linear sloth models

In this appendix, we discuss the representation of a linear velocity model in a linear sloth model. Since the linear velocity model is popular among geophysicists, the transform of a linear velocity model to a linear sloth model is useful. In a linear velocity model,

$$v(\mathbf{x}) = v(\mathbf{x}_0) + \mathbf{B}\cdot(\mathbf{x}-\mathbf{x}_0) + O(|\,\mathbf{x}-\mathbf{x}_0\,|^2). \tag{B1}$$

where $\mathbf{B}$ is the coefficient for linear velocity. Then

$$\frac{1}{v^2(\mathbf{x})} = \frac{1}{v^2(\mathbf{x}_0)} - \frac{2}{v^3(\mathbf{x}_0)}\mathbf{B}\cdot(\mathbf{x}-\mathbf{x}_0), \tag{B2}$$

from which it follows that

$$\mathbf{A} = -\frac{2}{v^3(\mathbf{x}_0)}\mathbf{B}(1 + O(|\,\mathbf{x}-\mathbf{x}_0\,|)) \sim -\frac{2\mathbf{B}}{v^3(\mathbf{x}_0)}. \tag{B3}$$

Thus, the coefficient for linear sloth $\mathbf{A}$ can be approximately obtained from formula (B3).