Stanford Exploration Project, Report 93, November 11, 1997, pages 63–75

Three dimensional dynamic ray tracing in complex geological structures

Yalei Sun, Robert G. Clapp, and Biondo Biondi[1]

keywords: *ray tracing, modeling, dynamic, velocity*

ABSTRACT

We implement a robust three dimensional dynamic ray tracing algorithm, that can be applied to complex geological structures. We use a Runge-Kutta solver to solve the dynamic ray tracing system. This solver has the ability to adapt its step length in accordance with the local gradient of the slowness field. We applied the ray tracing method to three synthetic models. The results were accurate and robust when the model was smoothed with a 3-D triangle filter to make the models more "ray valid".

INTRODUCTION

3-D imaging and velocity analysis methods are computationally intensive. One reason is that they are iterative processes, which need a forward modeling algorithm as a kernel component. For example, 3-D Kirchhoff-type imaging usually needs a 3-D Green's function as input. There are many different ways of estimating Green's function. They can be roughly categorized into two families according to both accuracy and efficiency: *limited bandwidth methods* and *infinite frequency methods* (Audebert et al., 1994).

Limited bandwidth methods usually deal with frequency-dependent, complex-valued Green's function. These methods are typically the extrapolation or modeling algorithms working in the time domain or the temporal frequency domain, e.g., phase-shift or finite-difference downward continuation. Nichols (1994) presented a method of calculating band-limited traveltime explicitly, which is essentially a compromise between full wavefield modeling and ray tracing. In terms of accuracy, it is a reduced version of a full wavefield modeling; in terms of efficiency, the cost of this new method is lower than that of full wavefield modeling, but higher than ray tracing method. One main weakness with this method is that no explicit raypaths are calculated and stored. In many applications, e.g., tomographic velocity analysis, raypaths are indeed necessary.

[1]**email:** yalei@sep.stanford.edu, bob@sep.stanford.edu, biondo@sep.stanford.edu

Ray tracing method is the main sub-family of the infinite frequency methods. The simplest and also the fastest ray tracing algorithm is *geometric ray tracing*. It transfers the eikonal equation to the ray equation using the characteristic method and then estimates the raypaths by solving the ray equation. Geometric ray tracing can only provide traveltime and cannot provide any amplitude and phase-shift information. It has been proved that first arrivals cannot be used to image complex structures correctly (Rekdal and Biondi, 1994). *Dynamic ray tracing* (Červený, 1985), is an extension to geometric ray tracing. Dynamic ray tracing (including paraxial ray approximation and Gaussian Beam approximation) can provide not only multiple traveltimes but also amplitude and phase-shift information. Therefore it can also be regarded as a compromise between full wavefield modeling and ray tracing modeling.

In this paper, on the basis of Červený's work, we develop an accurate and robust 3-D dynamic ray tracing system. First, our implementation is independent of the model gridding scheme, which can be layer, cube, or tetrahedron, etc. Therefore, it is easy to incorporate our implementation into different 3-D velocity model representations. Second, since dynamic ray tracing is a high-frequency approximation, it has some requirements over the smoothness of the model. In order to make the model more "ray valid", we apply a triangle smoothing filter to the model along the x, y, z axis. The smoothed model makes not only our numerical solver more stable, but also the results from dynamic ray tracing more like band-limited seismic data. We use a Fortran90 version Runge-Kutta solver in our application (Brankin and Gladwell, 1995). This solver has the ability to adapt its step-length in accordance with the local gradient of the slowness, which makes it more accurate than constant step-length solver and also enables us to deal with more complex structures without smoothing it dramatically. Another advantage is that Fortran90 is easily parallelized and we expect our implementation would be more efficient when implemented on a parallel computer.

In the next several sections, we first discuss our representation of the model and then present a brief version of the dynamic ray tracing theory. After that, we show three results from applying our dynamic ray tracing program to synthetic models. The linear model verifies that our implementation is correct when there is a theoretical solution. The layer model shows the importance of smoothing the model. The saltdome model demonstrates our method's robustness.

MODEL PRESENTATION

3-D dynamic ray tracing is a forward problem. It requires that a physical model should be known in advance. Constructing a 3-D slowness model is an open topic. Different model gridding schemes have different characteristics. In order to be more general and portable, our dynamic ray tracing program is designed to be independent of the model gridding scheme. It only requires the local velocity information in the vicinity of the central ray be known. We can choose different types of gridding schemes, such as cube, layer, or tetrahedron, according to the characteristics of the geological structure. We use cubic scheme in our application only because it is easy to implement. we use slowness instead of velocity in this paper.

Since the model has already been gridded, the model construction is transferred into an expression of the slowness within each small cube. We select a trilinear interpolation function.

$$s(x, y, z) = s_0 + s_{\mathrm{x}}x + s_{\mathrm{y}}y + s_{\mathrm{z}}z + s_{\mathrm{xy}}xy + s_{\mathrm{xz}}xz + s_{\mathrm{yz}}yz + s_{\mathrm{xyz}}xyz \tag{1}$$

where x, y, z are Cartesian coordinates. The advantage of trilinear functions is that the slowness is continuous from one cube to another. Therefore the direction of the raypaths are continuous when crossing the cube boundary. The coefficients of (1) can be solely determined from the eight slowness value of the cube. In our application, we first associate the ray point with eight grid points which construct a cube containing the ray point. From the slowness values of eight grid points, we get a trilinear expression for this cube. Then we can estimate the slowness at any point inside the cube.

THEORY OF DYNAMIC RAY TRACING

Conceptually, dynamic ray tracing can be divided into two steps. The first step is the same as geometric ray tracing, from which we get raypaths and traveltimes along the raypaths. These raypaths are called "central ray". The second step is to estimate traveltime, amplitude, and phase-shift in the vicinity of each central ray. In other words, each central ray represents a 3-D "ray tube".

The feature that ray tracings are high-frequency approximation requires that the variation of velocity should be smooth enough within one wavelength. Therefore, smoothing the model is necessary to make the model more "ray valid". Furthermore, smoothing the model can make ray tracing more like band-limited solutions (Biondi, 1992).

There are many references on the theory of dynamic ray tracing (Červený, 1985; Červený, 1987; Červený et al., 1982; Červený and Psenćik, 1983). In order to make this paper complete, we give a brief version of the dynamic ray tracing theory by skipping the deductions.

3-D geometric ray tracing

Assume that a ray is specified by the parametric equation

$$x_{\mathrm{i}} = x_{\mathrm{i}}(\tau), \qquad i = 1, 2, 3 \tag{2}$$

where τ is the traveltime along the ray, with $\tau = 0$ at the location of the source. The independent variable is not necessarily τ. It can be any other variable which increases monotonously along the ray, e.g., arclength is a common choice in the ray tracing algorithm. Here we choose traveltime because seismic data are sampled along time axis. Therefore, it is more convenient to compare the ray tracing results with seismic data.

The three coordinates are the solution of geometric ray tracing system

$$\frac{dx_i}{d\tau} = s^2 p_i, \qquad \frac{dp_i}{d\tau} = \frac{1}{s}\frac{\partial s}{\partial x_i} \tag{3}$$

where s is the slowness along the ray. The reason we use slowness, not velocity, is to follow the convention of tomography. Here, p_i are the Cartesian components of the slowness vector $\vec{p}$ and

$$p_i = \frac{\partial t}{\partial x_i}, \qquad i = 1, 2, 3 \tag{4}$$

Equation (3) is first-order ordinary differential equations. Usually, we cannot find an analytical solution for a general velocity model. It is more common to use some numerical algorithms, e.g., the Runge-Kutta method. In this paper, we use a Fortran90 version Runge-Kutta solver. In order to fit our expression to the convention of *rksuite90* package, the geometric ray tracing system is expressed as

$$\frac{d}{d\tau}\begin{bmatrix} x_1 & p_1 \\ x_2 & p_2 \\ x_3 & p_3 \end{bmatrix} = \begin{bmatrix} s^2 p_1 & \frac{1}{s}\frac{\partial s}{\partial x_1} \\ s^2 p_2 & \frac{1}{s}\frac{\partial s}{\partial x_2} \\ s^2 p_3 & \frac{1}{s}\frac{\partial s}{\partial x_3} \end{bmatrix} \tag{5}$$

The initial condition is

$$\begin{bmatrix} x_1^0 & p_1^0 \\ x_2^0 & p_2^0 \\ x_3^0 & p_3^0 \end{bmatrix} = \begin{bmatrix} x_{10} & s_0 \sin\alpha_0 \cos\beta_0 \\ x_{20} & s_0 \sin\alpha_0 \sin\beta_0 \\ x_{30} & s_0 \cos\alpha_0 \end{bmatrix} \tag{6}$$

where α_0 and β_0 are the two initial take-off angles, as shown in Figure 1.

3-D dynamic ray tracing

Dynamic ray tracing will provide us dynamic information like amplitude and phase-shift, not only kinematic information like raypaths and traveltimes. More specifically, we can estimate the traveltime, amplitude, and phase-shift in the vicinity of the central ray. In order to do this, we need to introduce another two coordinate systems, i.e., ray-centered coordinate system (q_1, q_2, q_3) and ray parameter coordinate system (γ_1, γ_2, γ_3).

The ray-centered coordinate system is a curvilinear orthogonal coordinate system. Let us consider an arbitrarily selected ray Ω, which is specified by ray parameter γ_1, γ_2. The ray-centered coordinates q_1, q_2, q_3 connected with the ray Ω are defined in

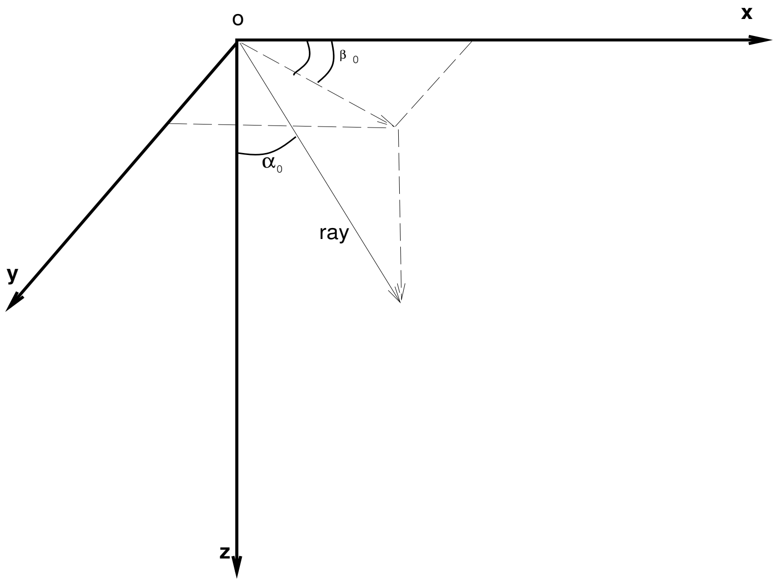

Figure 1: The initial condition of each central ray. (x, y, z) forms a Cartesian coordinate system. O is the source location. yalei2-take-off [NR]

the following way: q_3 corresponds to the traveltime along the ray Ω. q_1 and q_2 form a 2D orthogonal Cartesian coordinate system in the plane perpendicular to Ω, as shown in Figure 2.

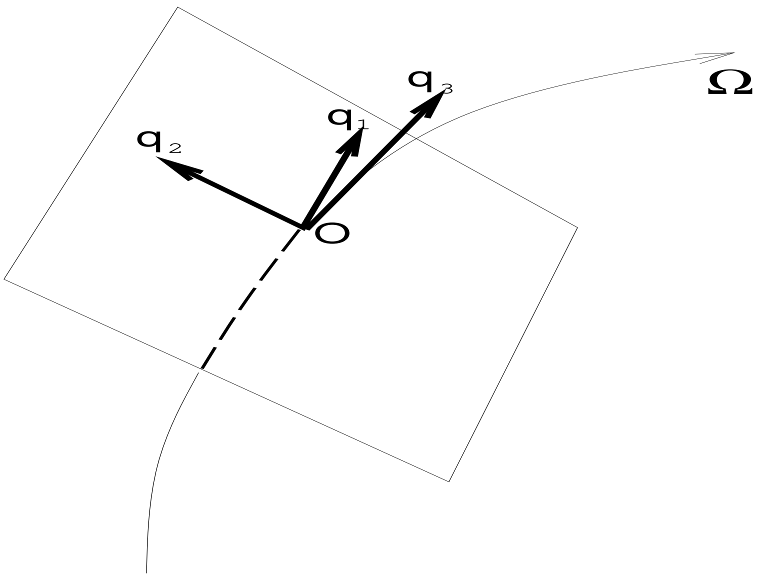

Figure 2: The ray-centered coordinate system. q_3 is tangent to the raypath. q_1 and q_2 forms a plane perpendicular to q_3. yalei2-ray-center [NR]

The physical meaning of the ray parameter coordinate system is very obvious. γ_1 and γ_2 are two ray parameters in the x_1 and x_2 direction, which are constant along the whole ray and variable from one ray to another. Their definitions are as follows

$$\gamma_1 = \frac{\partial t}{\partial x_1}, \qquad \gamma_2 = \frac{\partial t}{\partial x_2} \tag{7}$$

γ_3 is chosen as q_3. It increases monotonously along the ray.

Dynamic ray tracing system can be expressed as

$$\frac{d}{dt}\mathbf{Q} = s^{-2}\mathbf{P}, \qquad \frac{d}{dt}\mathbf{P} = -s\mathbf{V}\mathbf{Q} \tag{8}$$

where $\mathbf{Q}$, $\mathbf{P}$, $\mathbf{V}$ are 2×2 matrices.

$$\mathbf{Q} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}, \qquad \mathbf{P} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \tag{9}$$

The definition of each component is

$$Q_{\mathrm{IJ}} = \frac{\partial q_{\mathrm{I}}}{\partial \gamma_{\mathrm{J}}}, \qquad P_{\mathrm{IJ}} = \frac{\partial p_{\mathrm{I}}}{\partial \gamma_{\mathrm{J}}}, \qquad V_{\mathrm{IJ}} = \frac{\partial v}{\partial q_{\mathrm{I}} \partial q_{\mathrm{J}}} \tag{10}$$

where $q_{\mathrm{I}}(I = 1, 2)$ is the ray-centered coordinate. $\gamma_{\mathrm{I}}(I = 1, 2)$ is the ray parameter associated with each ray. $v = \frac{1}{s}$ is the medium velocity.

Mathematically, $\mathbf{Q}$ and $\mathbf{P}$ can be interpreted as transformation matrices: $\mathbf{Q}$ is a transformation matrix from the ray parameters γ_1, γ_2 to the ray-centered coordinate q_1, q_2, $\mathbf{P}$ is a transformation matrix from the ray parameters γ_1, γ_2 to the slowness vector component in the ray-centered coordinate system. Physically, matrix $\mathbf{Q}$ measures the derivations of paraxial rays from ray Ω and is also referred to as the matrix of geometrical spreading. Matrix $\mathbf{P}$ has no obvious physical meaning. But we can define a 2×2 matrix $\mathbf{M}$ of second derivatives of the traveltime field with respect to the ray-centered coordinate q_1, q_2 by the relationship

$$M_{\mathrm{IJ}} = \frac{\partial^2 \tau}{\partial q_{\mathrm{I}} \partial q_{\mathrm{J}}}. \tag{11}$$

Then we can write

$$\mathbf{M}(\tau) = \mathbf{P}\mathbf{Q}^{-1} \tag{12}$$

In order to fit the requirement of Runge-Kutta package, equation (8) is expressed as

$$\frac{d}{d\tau}\begin{bmatrix} Q_{11} & P_{11} \\ Q_{12} & P_{12} \\ Q_{21} & P_{21} \\ Q_{22} & P_{22} \end{bmatrix} = \begin{bmatrix} s^{-2}P_{11} & -s\frac{\partial^2 v}{\partial q_1^2}Q_{11} - s\frac{\partial^2 v}{\partial q_1 \partial q_2}Q_{21} \\ s^{-2}P_{12} & -s\frac{\partial^2 v}{\partial q_1^2}Q_{12} - s\frac{\partial^2 v}{\partial q_1 \partial q_2}Q_{22} \\ s^{-2}P_{21} & -s\frac{\partial^2 v}{\partial q_1 \partial q_2}Q_{11} - s\frac{\partial^2 v}{\partial q_2^2}Q_{21} \\ s^{-2}P_{22} & -s\frac{\partial^2 v}{\partial q_1 \partial q_2}Q_{12} - s\frac{\partial^2 v}{\partial q_1^2}Q_{22} \end{bmatrix} \tag{13}$$

The initial condition is chosen as

$$\begin{bmatrix} Q_{11}^0 & P_{11}^0 \\ Q_{12}^0 & P_{12}^0 \\ Q_{21}^0 & P_{21}^0 \\ Q_{22}^0 & P_{22}^0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \tag{14}$$

Here we set matrix $\mathbf{Q^0}$ and $\mathbf{P^0}$ to be 2×2 identity matrix. The reason for choosing an identity matrix is to make sure that $\mathbf{M}$ is a symmetric matrix.

The paraxial ray tracing solution of the pressure field can be written as

$$\psi(\mathbf{x}) = C\left(\frac{\alpha}{\alpha_0 \; det\mathbf{Q}}\right)^{1/2} e^{i\omega(\tau(\mathbf{0}) + \frac{1}{2}\mathbf{q}^T\mathbf{M}\mathbf{q})} \tag{15}$$

Sub-index 0 denotes a value taken at the source. C is a factor dependent upon the source, and is constant along the ray.

RESULTS AND DISCUSSION

As a new implementation, our dynamic ray tracing needs some tests to prove its accuracy. The first synthetic model is designed for this purpose. It is a linear velocity model. We can get theoretical solution for this model. The second model is a two-layered model. We use this model to show the importance of smoothing the model. We also show the change of amplitude caused by the interface between the two layers. The third one is a synthetic saltdome model, which is issued by EAEG/SEG velocity modeling committee. This model is much more complicated and therefore presents a real challenge. The result from this model also demonstrates the effectiveness of our approach.

Synthetic model (linear velocity)

In order to prove our program "works", we need some "true" results from the same model to check the accuracy of our results. Since dynamic ray tracing is a high-frequency approximation. Traveltime is usually more appropriate than amplitude and phase-shift to be chosen to verify a new algorithm. Fortunately, we know the theoretical solutions for some simple models, e.g., constant velocity or linear velocity. Here we design a linear velocity model and use it to test our new approach. The size of the whole model is $100 \times 100 \times 120$. The size of each cube is $12.5m \times 12.5m \times 12.5m$. The linear velocity model is as following:

$$v(x, y, z) = v_0 + az \tag{16}$$

where $v_0 = 2000m/s$ and $a = 2$.

The theoretical solution of wavefront for this model can be found in Slotnick (1959).

$$\tau = \frac{1}{a}\cosh^{-1}[1 + \frac{a^2(\Delta x^2 + \Delta y^2 + \Delta z^2)}{2v_0(v_0 + az)}] \tag{17}$$

where

$$\Delta x = x - x_0, \quad \Delta y = y - y_0, \quad \Delta z = z - z_0 \tag{18}$$

(x_0, y_0, z_0) is the location of the source.

Figure 3 shows the numerical result from dynamic ray tracing and the theoretical solution. The ray tracing result agrees with the theoretical result satisfactorily.

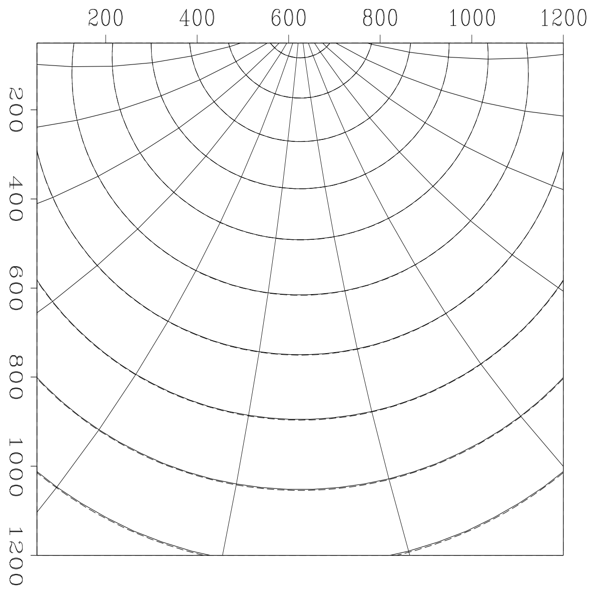

Figure 3: Solid lines represent numerical result (including ray and wavefront). The dash line represents theoretical result. The wavefront from ray tracing is consistent with the wavefront from equation (17). The raypaths also bend upwards because of the increase of velocity with depth. yalei2-model1 [CR]

Synthetic model (two layers, velocity within each layer is constant)

This synthetic model is essentially composed of two constant velocity layers with a transient zone between the two layers. The geometrical size of this model is the same as last one. The velocity of upper layer is $2000m/s$ and that of lower one $3000m/s$. The interface will make the Runge-Kutta algorithm fail if the model is not smoothed reasonably. We apply a triangle smoothing filter to the model along `x, y, z` axis. The result is shown in Figure 4.

EAEG/SEG saltdome model

This is a synthetic saltdome velocity model issued by EAEG/SEG velocity modeling committee. The size of the model is $400 \times 400 \times 200$ grids and the geometric size of each cube is $20m \times 20m \times 20m$. First, we smooth the model using the same smoothing filter we used in the last model. Figure 5 shows the portion where the raypaths hit

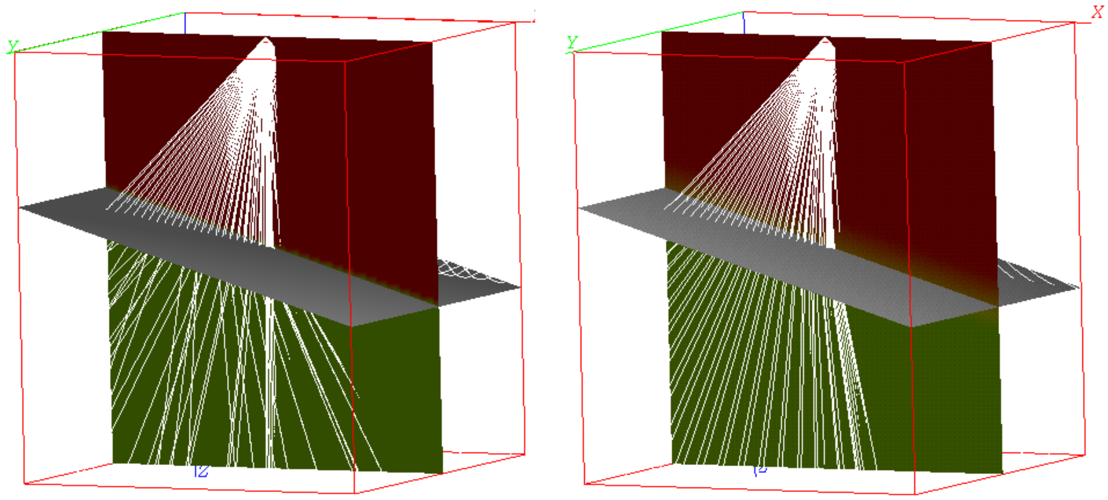

Figure 4: **Left**: The sharp interface makes the Runge-Kutta algorithm fail. The rays propagate in random directions when passing through the interface. Some of them are even reflected back to the upper layer. **Right**: The interface has been smoothed by the triangle smoothing filter. The rays' propagating directions follow Snell's law. All the rays pass through the transition zone and enter the lower layer. yalei2-model2-smooth [CR]

the flank of the saltdome. Since the saltdome velocity is much higher than the other parts. Most of the rays are "bounced" back by the flank except those rays hitting the flank with small incident angle. We also show a small portion of the corresponding wavefront. The color of the wavefront represents the magnitude of the amplitude. The amplitude is homogeneous before the wavefront hits the saltdome. When the wavefront hits the flank of saltdome, the amplitude changes quickly because of the wave resistance. Figure 6 is generated using AVS's clipping function. We use one model slice and one clipping plane (transparent) to extricate a ring of wavefront. The model slice shows the saltdome's structure clearly. We can see several triplications on the ring. This phenomenon is caused by the complicated velocity model. Therefore it is obvious that first arrival approach cannot image such a complicated structure. We need to do some interpolation to get the maximum energy traveltime.

CONCLUSION AND FUTURE WORK

In this paper, we develop an accurate and robust 3-D dynamic ray tracing system. Our implementation is independent of the model gridding scheme and is easy to be incorporated into 3-D velocity model representation methods. We use a triangle smoothing filter to smooth the model to make the model more "ray valid" and the

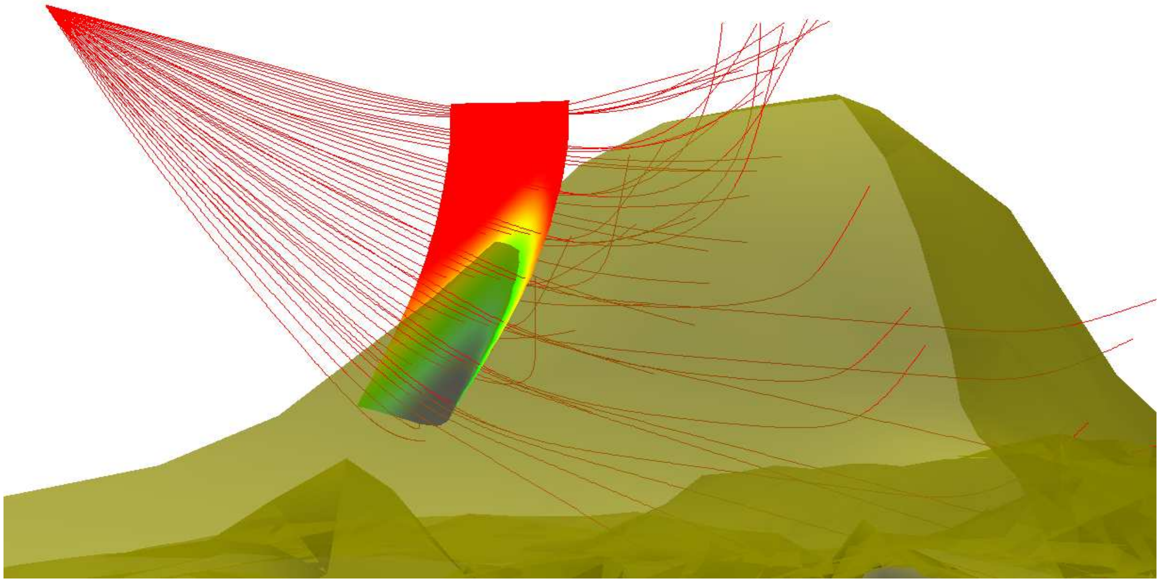

Figure 5: The raypaths diverge at the flank of saltdome. Variation in amplitude along the wavefront as it enters the saltdome are also shown, with red; representing high amplitudes, blue low amplitudes. yalei2-model3-ray-front [CR]

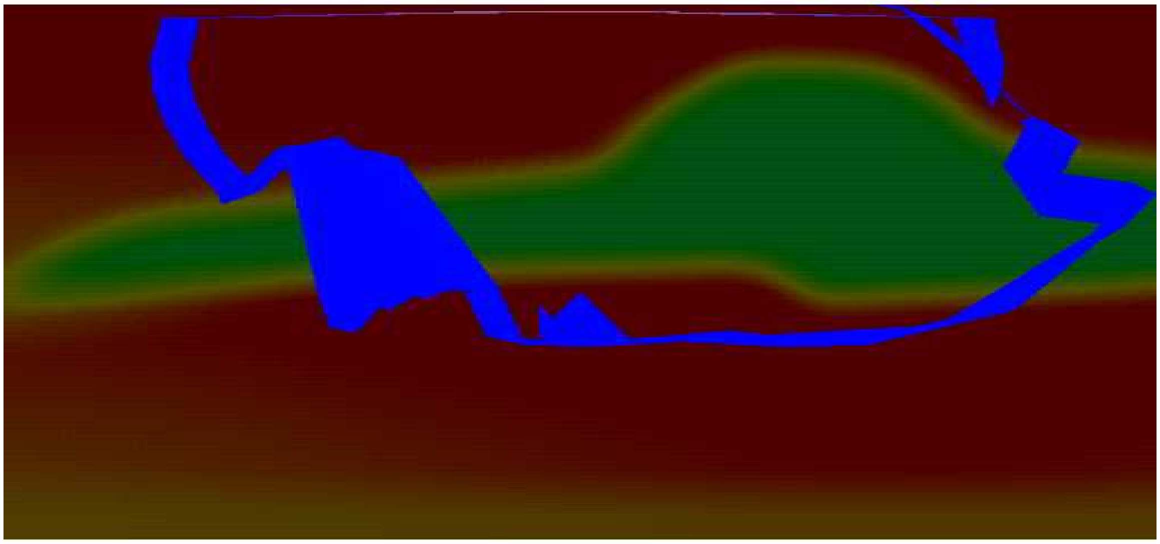

Figure 6: Because of the influence of the saltdome, the wavefront becomes complicated. yalei2-model3-clip [CR]

numerical solution more stable. The Fortran90 version Runge-Kutta solver *rksuite90* has the ability to adapt its step-length to fit the requirement of the local gradient of the slowness. The results in this paper verifies that our program is correct and also demonstrates our method's capability of handling complex model.

We have not calculated the phase-shift along the central ray. It is straight-forward after we get **Q**, **P**, and **M**, as shown in equation (15). Theoretically, dynamic ray tracing theory can be further categorized into *paraxial ray tracing* and *Gaussian beam approximation*. Gaussian beam approximation can deal with complex structures more accurately, but it is also more time-consuming. In the past, researchers always considered these two approximations separately. We will investigate the possibilities of combining them together to achieve an accurate result with reasonable costs.

Our final aim for this project is to generate a regularly gridded 3-D Green's function for complex structures, which is essential to 3-D imaging and velocity analysis. The result from dynamic ray tracing is irregularly distributed on the model. So we need some interpolation schemes to bridge the gap between spatial regularly-sampled Green's function and spatial irregularly-sampled ray tracing result. Our idea is to first find the nearest-neighborhood ray tracing results for each grid point. One possible candidate for the searching scheme is K-D tree, which is an extension to binary tree data structure. Then we design a linear operator to link the regular grids and the ray tracing results together. This is actually a large-scale inverse problem. We can choose different iterative solvers to interpolate the kinematic and dynamic information on the grid from the information on the raypaths.

REFERENCES

Audebert, F., Biondi, B., Lumley, D., Nichols, D., Rekdal, T., and Urdaneta, H., 1994, Marmousi traveltime computation and imaging comparisons: SEP–**80**, 47–66.

Biondi, B., 1992, Solving the frequency-dependent Eikonal equation: 62nd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1315–1319.

Brankin, R., and Gladwell, I., 1995, RKSUITE90: Software for ODE IVPS: ftp://seas.smu.edu.

Červený, V., and Psenćik, I., 1983, Gaussian beams and paraxial ray approximation in three-dimensional elastic inhomogeneous media: J. Geoph., **53**, 1–15.

Červený, V., Popov, M. M., and Psenćik, I., 1982, Computation of wave fields in inhomogenous media: Geophys. J. Roy. Astr. Soc., **70**, 109–128.

Červený, V., 1985, The application of ray tracing to the numerical modeling of seismic wavefields in complex structures.: Seismic Shear Waves, Part A, Theory, 1–124.

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

Nichols, D. E., 1994, Imaging complex structures using band-limited Green's functions: Ph.D. thesis, Stanford University.

Rekdal, T., and Biondi, B., 1994, Ray methods in rough models: SEP–**80**, 67–84.

Slotnick, M. M., 1959, Lessons in seismic computing *: Soc. Expl. Geophys., Tulsa.