

Ray tracing by simulated annealing: Bending method

Andrej Bóna¹, Michael A. Slawinski², and Peter Smith³

ABSTRACT

We propose a new ray-tracing method based on the concept of simulated annealing. Using this method, we find rays between fixed sources and receivers that render traveltimes globally minimal. With our method, we are able to construct rays and their associated traveltimes to satisfactory precision in complex media. Furthermore, our algorithm can be modified to calculate rays of locally minimum traveltime, such as reflected rays, by constraining the ray to pass through a set of points that we are free to specify.

INTRODUCTION

The shooting and bending methods are two commonly used numerical approaches to solve boundary-value ray-tracing problems. The shooting method is based on solving ray equations as initial-value problems by specifying the takeoff angle. The takeoff angle is varied until the ray passes through the receiver position. The shooting method works well to find rays connecting sources and receivers in simple 2D media. It breaks down in areas where ray equations break down, such as shadow zones, and in complicated media where a slight variation in the takeoff angle might result in a significantly different ray, thus causing difficulty in connecting sources to receivers (Červený, 2001).

The bending method addresses intrinsically the problem of connecting sources to receivers. One begins by connecting source S to receiver R with an initial path. This initial path is bent according to a prescribed method based on minimizing traveltime according to Fermat's principle until the desired ray is obtained. If we search for the ray corresponding to the global minimum of the traveltime, our method allows us to avoid obtaining only local minima. We can calculate paths that correspond to local minima of traveltime when we constrain them to pass through an interface.

The bending method has been used extensively. Bending methods used by Julian and Gubbins (1977) and Pereyra et al. (1980) use finite-difference methods to solve ray equations, and to construct rays constrained to pass through a specified receiver. Um and Thurber (1987) and Prothero et al. (1988) take a different approach: each shows how rays can be constructed by perturbing the shape of the ray itself. Um and Thurber start with a three-point guess to the ray and then use geometric interpretation of the ray equations to perturb it. Prothero et al. make an initial guess and then perturb it locally to find the ray. The approach of Um and Thurber is more efficient, although less accurate numerically than that of Prothero et al. In addition, Koketsu and Sekine (1998) show that the bending method of Um and Thurber is more stable numerically than the results of Pereyra et al. (1980), especially in the presence of velocity-model discontinuities. As shown by Pereyra et al. (1980), accuracy of the pseudobending method decreases as the number of discontinuities of the model increases. These methods depend on the initial path; if the path is not chosen carefully, one can end up with a ray corresponding to a local minimum traveltime.

Other ray-tracing methods exist. Moser (1991) utilizes network theory to calculate the entire rayfield in the shortest-path method. This method always finds the global minimum traveltime, but accuracy of the results depends on geometry of the underlying network. Sadeghi et al. (1999) propose the microgenetic algorithm, which combines the pseudobending method and a genetic algorithm to calculate rays. By combining these algorithms, they avoid complications that arise with discontinuities in the velocity function. However, their method is dependent on the initial guess.

As an alternative to these search algorithms, Vinje et al. (1993) and Vinje et al. (1996a, 1996b, 1999) propose a method in which wavefronts are calculated by ray tracing short elements in two and three dimensions, respectively. This method uses ray tracing to calculate the shapes of elementary wavefronts that propagate in a smoothly varying inhomogeneous medium. Although fast and robust in practice, its main disadvantages are difficulty with mapping irregularly spaced data onto regular grids (Gibson et al., 2005) and ray tracing in caustic regions (Lambaré et al., 1996).

Manuscript received by the Editor 23 October 2007; revised manuscript received 27 July 2008; published online 18 February 2009.

¹Curtin University of Technology, Department of Exploration Geophysics, Perth, Australia. E-mail: a.bona@curtin.edu.au.

²Memorial University, Department of Earth Sciences, St. John's, Newfoundland and Labrador, Canada. E-mail: mslawins@mun.ca.

³Simon Fraser University, Department of Physics, Burnaby, British Columbia, Canada. E-mail: psa15@sfu.ca.

© 2009 Society of Exploration Geophysicists. All rights reserved.

Our purpose is to expand on the work of Smith (2006) in a new ray-tracing method based on bending paths according to minimization of associated traveltime using simulated annealing. Simulated annealing is used in mathematics and physics to obtain an optimal solution to problems subject to constraints (Kirkpatrick et al., 1983; Černý, 1985). Velis and Ulrych (1996) use this method in the context of ray tracing, developing an algorithm to search for the takeoff angle of the ray that minimizes traveltime in a 2D medium. The authors extend their ray-tracing algorithm to calculate rays in 3D media (Velis and Ulrych, 2001). The key difference between their algorithm and ours is that their algorithm is based on varying the takeoff angle of the initial ray, whereas ours is based on varying the path itself.

Our method overcomes two common shortcomings of other bending methods. First, solutions calculated using our algorithm are independent of the initial path. Second, our algorithm does not require using smoothly varying velocity models. We demonstrate the applicability of our method with two complex velocity models. Using a linearly inhomogeneous and elliptically anisotropic model, we show that our method can construct rays in smooth media and it can be generalized to a three-point problem, which is applicable for calculating rays constrained to pass through multiple interfaces. Then we study the Marmousi model, calculating rays in complex velocity structures with accuracy comparable to existing ray-tracing methods.

BACKGROUND

Rays

A ray describes a signal path. Properties of the medium are given by function $V(\mathbf{x})$, which describes the velocity of a signal at any point \mathbf{x} within the medium. In this work, we consider 2D media, where we denote a single point in the medium by coordinates $\mathbf{x} = (x, z)$.

The traveltime along a ray γ depends on the velocity and path, namely,

$$T(V(\mathbf{x}); \gamma) = \int_{\gamma} \frac{ds}{V(\mathbf{x})}, \quad (1)$$

where ds is an arc-length element and γ connects the fixed endpoints. The problem of finding rays can be solved as Fermat's variational problem of finding a path of stationary traveltime between a source and a receiver (Červený, 2001; Bóna and Slawinski, 2003).

Simulated annealing

In physics, annealing is a physical process in which a solid in a heat bath is melted by increasing the temperature of the heat bath to a value at which all particles arrange themselves randomly in the liquid phase. This is followed by cooling the particles back into the solid phase by slowly lowering the temperature of the heat bath. As the liquid cools, it must be allowed to reach thermal equilibrium with the heat bath to ensure the particles settle in the state of minimum energy upon reaching the freezing point. This is referred to as the ground-state configuration of the system or the ground state (Chaikin and Lubensky, 1995). If the cooling process is carried out too fast, one runs the risk of freezing defects in the structure of the solid.

Simulated annealing is based on the analogy between simulation of the annealing of solids and the problem of solving large combinatorial problems (Van Laarhoven and Aarts, 1987). This method, de-

veloped by Černý (1985) and Kirkpatrick et al. (1983), applies statistical mechanics reasoning to solve optimization problems. The goal is to construct the optimal solution (the ground state) for a particular problem (the system) by varying a trial solution according to a specific protocol (heating, then cooling the system) with respect to a particular parameter (temperature). We use the Metropolis algorithm to carry out the perturbations at each temperature. For a complete description of the Metropolis algorithm and its application to simulations of statistical mechanical systems, see Metropolis et al. (1953) and Newman and Barkema (2002). To implement the simulated annealing algorithm as a method for ray tracing, let us consider analogies between ray theory and statistical mechanics.

Analogies between ray theory and statistical mechanics

The goal of simulated annealing is to find the ground state of the system under consideration. We associate system configurations with paths connecting sources and receivers. Let $\gamma(S, R)$ represent a particular path connecting source S to receiver R , and let $T(\gamma)$ denote the traveltime associated with this path.

Our goal is to find the path minimizing traveltime between source and receiver. In this context, it is natural to associate energy with traveltime along a path. As a result, the ground state is the path that minimizes traveltime. Hence, the ray corresponds to the ground state.

To represent the role of the heat bath in ray tracing, we use parameter τ , which has units of time. We lower the value of this parameter in the same way one would lower the temperature of the heat bath in annealing.

The transition probability A of moving from path γ_i to γ_j at τ is

$$A(\gamma_i \rightarrow \gamma_j) = \begin{cases} 1 & \Delta T \leq 0 \\ \exp\left(-\frac{\Delta T}{\tau}\right) & \Delta T > 0 \end{cases}, \quad (2)$$

where $\Delta T = T(\gamma_j) - T(\gamma_i)$ is the change in traveltime of the signal associated with perturbation of the path.

Any perturbation that lowers traveltime along the path is accepted automatically, although any perturbations that increase traveltime are accepted with a probability that decreases with temperature. For example, perturbations that increase the traveltime along the path by $\Delta T(\gamma_j) \approx \tau$ have a 30% probability of acceptance at that particular τ . When τ is lowered, this probability decreases, so perturbations of this size are less likely to be accepted.

RAY-TRACING METHODOLOGY

Media and rays: Discretization

In our method, we represent a 2D medium by a rectangular grid \mathcal{D} of $N = L_x \times L_z$ points, where L_x and L_z denote the number of grid points along the x -axis and z -axis, respectively. We let parameters h_x and h_z denote the spacing between points on \mathcal{D} . We are not required to adhere to a grid representation of the medium. Instead, we are free to move points on the grid to other areas of the medium where ray tracing is frustrated by grid geometry, e.g., mapping points in a constant velocity region to lie on an interface where tracing rays is hindered by lattice geometry. The grid representation is natural because it covers the medium uniformly.

We represent a path by a spline of straight lines that approximates its shape. Let S be a source, R a receiver, and $\gamma(s; S, R)$ be the path

connecting these points. Arc-length parameter s traces the shape of the path within the medium. Let $\bar{\gamma}(s; S, R)$ be a piecewise-continuous spline of M straight-line segments with endpoints on \mathcal{D} that approximates $\gamma(s; S, R)$:

$$\gamma(s; S, R) \approx \bar{\gamma}(s; S, R) = \bigcup_{k=1}^M \bar{\gamma}_k(s). \quad (3)$$

Term $\bar{\gamma}_k(s)$ in expression 3 represents the k th segment of the path in splined representation $\bar{\gamma}$. It is a linear function defined on interval $\mathcal{I}_k = [s_0^k, s_f^k]$:

$$\bar{\gamma}_k(s) = \mathbf{m}_k(s - s_0^k) + \mathbf{p}_k, \quad (4)$$

where \mathbf{m}_k is the direction of the k th line segment in the space of arc-length parameterization and \mathbf{p}_k corresponds to its initial point.

Traveltime: Calculations on grids

In view of approximating rays by equation 3, we approximate traveltime l as the sum of traveltimes over each segment:

$$T(V(\mathbf{x}); \gamma) \approx \sum_{k=1}^M T(V(\mathbf{x}); \bar{\gamma}_k), \quad (5)$$

where $T(V(\mathbf{x}); \bar{\gamma}_k)$ denotes the traveltime of a signal along the k th segment of the ray and $V(\mathbf{x})$ is the signal velocity. In our calculations, velocity is scaled according to the dimensionless distances on the grid and is expressed in units of s^{-1} .

The traveltime of a signal propagating along straight line $\bar{\gamma}_k$ is

$$T(V(\mathbf{x}); \bar{\gamma}_k(s)) = \int_{s_0^k}^{s_f^k} \frac{ds}{V(x(s), z(s))}. \quad (6)$$

We can integrate expression 6 numerically using the composite Simpson’s rule. In contrast to Moser et al. (1992), we include the support points (in our case, nodal points) to calculate traveltime along a line segment. Inclusion of nodal points is necessary to determine the locations of reflections along an interface. Furthermore, we integrate over the entire line segment rather than just at its support points.

Ray tracing by simulated annealing

For $S = (x_S, z_S)$ and $R = (x_R, z_R)$, we begin with a guess of a ray $\bar{\gamma}^0(\xi)$ that has M segments. The parameterization is determined by expression 4, and its traveltime is calculated by expression 5. For simplicity, we choose a straight line as an initial guess. We begin the annealing process at a high temperature so all perturbations to the system are accepted, and we cool the system until it settles into a stationary value. Hence, our solution does not depend on the initial guess, which is annihilated upon starting the calculations.

At each step, we choose a point randomly, $q = (x_q, z_q) \in \mathcal{D}$, that connects two segments on $\bar{\gamma}^0$. Then, we choose one of three possible perturbations randomly: the removal of the point, the addition of another point, or the shift of the point to another point on \mathcal{D} .

If we choose to remove the point, we connect the points adjacent to the chosen point q , as illustrated in Figure 1. If we choose to add another point, we split the segment to the left or right of q and connect it to a randomly chosen point $q' \in \mathcal{D}$, as illustrated in Figure 2. A shift of point q to q' is illustrated in Figure 3.

Let ΔT denote the change in traveltime corresponding to the chosen perturbation: addition, removal, or shift of a point along the path. For change in traveltime ΔT at τ , where τ has units of time and is the analog of temperature in our formulation, we accept this perturbation using transition probability A , given by expression 2.

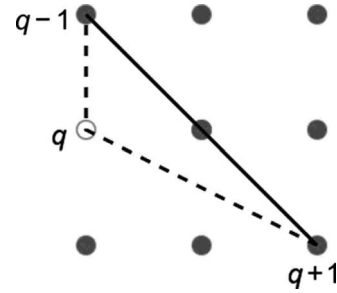


Figure 1. Perturbation that results in the removal of a point $q \in \bar{\gamma}^0$, where the new path is the line joining points $q - 1$ and $q + 1$.

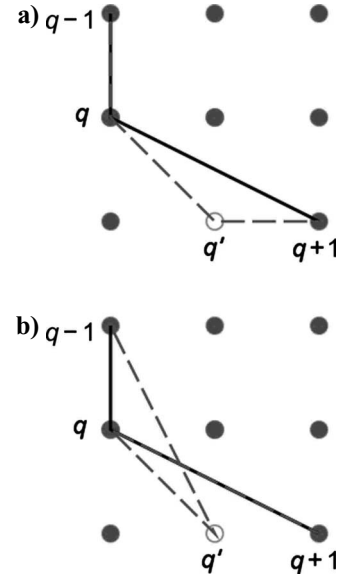


Figure 2. Perturbation to the path resulting in the addition of a point. Points $q \in \bar{\gamma}^0$ (open circle) and $q' \in \mathcal{D}$ (gray point) are chosen at random, and the original path is perturbed by splitting the segments joining (a) $q - 1$ to q and (b) q to $q + 1$ and joining each at q' , yielding the dashed and dot-dashed paths, respectively. Then we select the path of smallest traveltime for ensuing calculations.

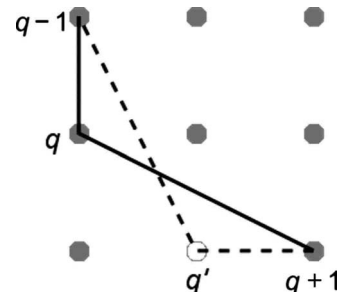


Figure 3. Perturbation to the path via shifting point q to q' .

Equilibration times and cooling schedules

By definition, a system is in equilibrium with its environment when each possible state is sampled according to the Boltzmann distribution. To equilibrate the path properly with respect to τ , we need to choose the number of paths ν with care. We choose ν to be an integer multiple of the number of grid points N , so we select each point on the grid at least once per temperature on average. If we choose ν to be small, the system will not equilibrate and we will end up with a configuration that is not optimal. Conversely, if we choose ν to be large, we sacrifice efficiency for accuracy. For the results presented in this work, we set $\nu = 10N$. Smaller values of ν yield results of similar accuracy; however, we keep ν large enough to ensure we sample all points on the grid sufficiently without sacrificing computational cost.

To simulate the cooling process in the annealing method, one can follow several schedules. Nourani and Andersen (1998) investigate the properties of several cooling schemes implemented in various simulated annealing studies. They consider constant thermodynamic speed, exponential, logarithmic, and linear cooling schedules. Comparisons are based on the amount of entropy produced during the annealing process for given initial and final states, and a fixed number of iterations. The cooling scheme yielding the least amount of entropy is deemed the best. In their study, the constant thermodynamic speed schedule performed the best; it always found the global minimum of the system in the prescribed number of iterations.

For our simulations, we use the exponential cooling schedule, which results in the following sequence:

$$\tau(i) = \tau_0 \exp(-\alpha i), \quad (7)$$

where $i = 0, 1, 2, \dots$ denotes the i th cooling step in the annealing process, τ_0 corresponds to the initial temperature, and α corresponds to the rate of cooling. Although this schedule did not yield the rapid convergence of the constant thermodynamic speed schedule, it did yield the best results of all fixed-rate cooling schedules considered. Implementing adaptive schedules such as the constant thermodynamic speed schedule can be studied in future work.

The final step before running the simulations is to choose appropriate values of τ_0 and α . The value of τ_0 is chosen for each model such that the acceptance rate is near unity at the beginning of the calculation. For our cooling schedule, we choose $\alpha = 0.5$. We cool for 25 temperatures. This allows us to cool our system by several orders of magnitude of temperature. Larger values of α result in rapid cooling schemes, where the possibility of quenching comes into play; smaller values result in a cooling scheme that samples many more temperatures at a cost of computation time needed to cool the system. Optimization of the value of α can be studied in future work.

We derive our chosen cooling schedule by constructing 100 rays between a given source and receiver pair and observing what set of parameters give the best balance between accuracy and efficiency. With the schedule outlined above, we construct a set of rays for which traveltimes differ on the order of milliseconds or less. Longer cooling schedules with larger values of ν are more accurate, at the expense of efficiency. Hence, we can say the data presented here are accurate on the scale of experimental error.

Three-point ray tracing

The main advantage of the bending method is that one needs to specify only the endpoints to calculate the ray and associated travel-

time. Thus, our method can be modified to find rays corresponding to reflections or refractions at an interface. Such rays are associated with local traveltime minima.

To modify the method, we calculate the paths that pass through a set of points on an interface. The constrained path yielding minimum traveltime is the solution to the three-point problem. This method is generalized easily to an $(M + 2)$ -point ray-tracing problem, where the ray is constrained to pass through M interfaces or areas. If the number of constrained points per interface or area is on the order of L , this results in $M \times L$ possible paths to calculate. Here we consider the simplest case, where $M = 1$, and propose a straightforward generalization to an $(M + 2)$ point scheme.

EXAMPLES AND DISCUSSION

In this section, we use our method to construct rays and compute traveltimes in two velocity models: layered elliptically anisotropic and Marmousi. We present results for two-point and three-point ray-tracing calculations connecting a single source and receiver.

At the beginning of each calculation, we calculate the traveltime between each possible set of points on the grid. As one increases the number of points, the time taken to calculate traveltime scales as $N(N - 1)/2$, where N is the number of points on the square grid.

We need to perform this calculation only once per simulation, as long as no modifications are made to the underlying grid. This is crucial when N is large. However, the amount of available memory imposes strict limits on the number of points we can consider in our calculations. In addition, this step is not equivalent to calculating all possible rays; the purpose of this calculation is to provide a look-up table to be used throughout the annealing process when calculating perturbations to a path.

For each model, we construct rays between the source and all the possible pairs. Then we use this data to qualitatively reconstruct the shapes of the wavefronts by plotting all points that lie on the isochron $T = t$ using Mathematica software. Before constructing the isochron data for each model, we constructed 100 rays connecting the source to a specific receiver to acquire insight into how our algorithm would behave in the media considered. Results obtained from the algorithm could be viewed as accurate to one part in 10^4 . Traveltimes of constructed rays in our simulation agree with each other up to a certain point. Although our algorithm does not yield the absolute minimum traveltime with 100% accuracy, the difference in path between the minimum and maximum observed traveltime at zero temperature is small.

All simulations were performed using a serialized code on a 2-GHz AMD Opteron dual-core 285 SE processor. Using the annealing process outlined previously, calculating each ray took about 2 s of CPU time. It took approximately 8 hours to calculate a set of rays starting from a specified source to all points on the grid. Citing the results of Smith (2006) and the results corresponding to the linearly inhomogeneous and elliptically anisotropic medium for rays not passing through an interface, our method is highly accurate for sufficiently smooth media.

In addition, we will compare the results from the Marmousi model to results published in the literature and comment on how our results agree. Our primary concern with this work is to illustrate that our method is accurate. To ensure this, we use a 101×101 grid to sample the domain of each model. We make no appeal to comparisons of CPU time, other than to cite the amount of time taken to perform our

calculations, because our algorithm is still in the process of optimization.

Model 1: Linearly inhomogeneous and elliptically anisotropic medium

The first model we consider combines a smoothly varying anisotropic velocity model in one layer with two constant-velocity layers, separated by interfaces with nonzero slope. The model we consider is given by

$$v(x,z) = \begin{cases} (3.0 + 0.75z)f(\theta) \text{ s}^{-1}, & z < \bar{z}_1(x) \text{ and } z < \bar{z}_2(x) \\ 6.00 \text{ s}^{-1}, & \bar{z}_1(x) \leq z < \bar{z}_2(x) \\ 4.25 \text{ s}^{-1}, & z \geq \bar{z}_2(x) \end{cases}, \tag{8}$$

where f depends on the direction of signal propagation, measured by θ with respect to the z -axis. Equations $\bar{z}_1(x) = 0.5 + 0.25x$ and $\bar{z}_2(x) = 0.75 - 0.25x$ represent interfaces within the medium.

In the layer $z < 0.75 - 0.25x$, signal velocity depends on the position and direction of propagation of the signal. This model corresponds to a medium that is linearly inhomogeneous along z and elliptically anisotropic. The anisotropy is given by $f(\theta) = \sqrt{(1 + 2\chi)/\sqrt{(1 + 2\chi \cos^2 \theta)}}$, where χ measures the ellipticity of the wavefronts. In our calculations, we set $\chi = 0.3$ (Slawinski, 2003). The analytical formula for traveltimes is given by Rogister and Slawinski (2005).

The grid space of our medium lies on a square of side length 1.0. For a 101×101 -point grid, this corresponds to a grid spacing of 0.01. In this section, all units corresponding to space are dimensionless. This proposed grid layout presents a problem: The interfaces we specify have slopes not equal to unity. Thus, the number of points along the interface for the raypath to pass through can reduce the accuracy of our algorithm in the sense of adhering to Snell's law along the interfaces. To remedy this, we add extra points to interfaces $\bar{z}_1(x)$ and $\bar{z}_2(x)$. The spacing between points along each interface is $\Delta s \approx 5.0 \times 10^{-3}$ units. This allows for increased accuracy in ray tracing through interfacial boundaries. These points are used later to calculate reflections off the interfacial boundaries.

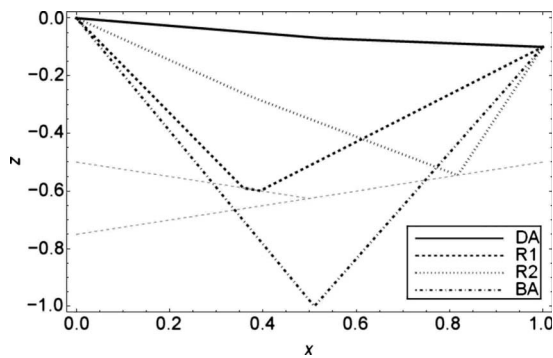


Figure 4. Rays in the complex medium defined by expression 8 corresponding to the direct arrival (DA), reflected arrival from the interface given by $\bar{z}_1(x)$ (R1), reflected arrival from the interface given by $\bar{z}_2(x)$ (R2), and the reflected arrival from the bottom (BA). Source located at (0, 0) and receiver located at (1.0, 0.1). Dashed gray lines correspond to the interfaces.

In calculating reflections at the interfaces $\bar{z}_1(x)$, $\bar{z}_2(x)$, and $z = 1.0$, we are able to determine the points corresponding to local minima along each interface to within 5×10^{-3} units of distance. When calculating reflections along the bottom, we reduce grid spacing along x to 5×10^{-3} units, corresponding to a range of [0.25, 0.75]. We plot the rays corresponding to the direct and reflected arrivals of a signal propagating from (0, 0) in Figure 4. The ray corresponding to the reflection of the signal along $\bar{z}_1(x)$, R1, contains elements of a head wave, which is a result of numerical optimization of traveltimes along the path constrained to pass through a point on this interface. Furthermore, our method finds multiple optimal constraints in this region, all corresponding to the same ray and all located along the section of the interface through which the ray passes. Thus, our method finds multiple optimal constraints in areas where head waves optimize traveltimes along the path.

Table 1 contains traveltimes calculated for all considered paths. We see good agreement between numerically obtained and analytically calculated traveltimes of the direct arrival. In Figure 5, we plot isochrons corresponding to a wave propagating from a source at (0, 0) in intervals of 0.05 s. This graph clearly shows the expected shape of the wavefronts, including the ellipticity of wavefronts in the upper region of the medium.

Table 1. Traveltimes (in seconds) in the complex medium defined by expression 8, corresponding to the direct arrival (DA), reflected arrival from the interface given by $\bar{z}_1(x)$ (R1), reflected arrival from the interface given by $\bar{z}_2(x)$ (R2), and the reflected arrival from the bottom (BA). The source is located at (0, 0) and receiver at (1.0, 0.1).

S	Exact	Calculated
DA	0.26194	0.26199
R1	N/A	0.41495
R2	N/A	0.40758
BA	N/A	0.56330

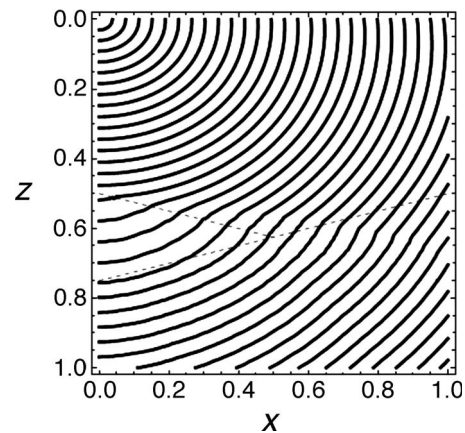


Figure 5. Isochrons in the complex medium defined by expression 8 corresponding to the elementary wavefront propagating from a source located at (0, 0). Isochrons are spaced in 0.01-s intervals. Dashed gray lines correspond to interfaces.

Model 2: Marmousi model

Next, we consider the 2D Marmousi model (Versteeg and Grau, 1991). We calculate rays and their associated traveltimes using a smoothed version of Marmousi. Given the irregular nature of the velocity field, we use an underlying square grid of 101×101 points, which gives us the best possible approach to sampling the velocity

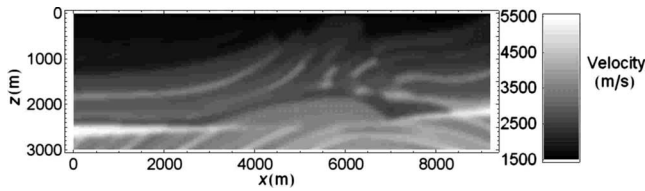


Figure 6. Smoothed Marmousi model: velocity field plotted in Mathematica using a set of 384×121 data points.

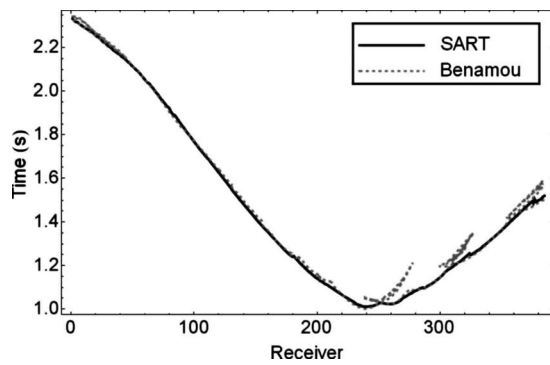


Figure 7. Comparison of results obtained by our simulated annealing method to the big ray-tracing method of Benamou (1996). Plot of traveltime versus receiver in the Marmousi model; source is located at (6000 m, 2800 m), and 384 receivers are spaced at 24-m intervals along the surface.

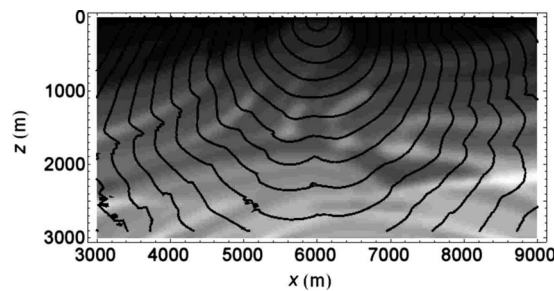


Figure 8. Wavefronts calculated using simulated annealing method; source located at (6000 m, 0 m). Isochrons are plotted in 0.1-s intervals.

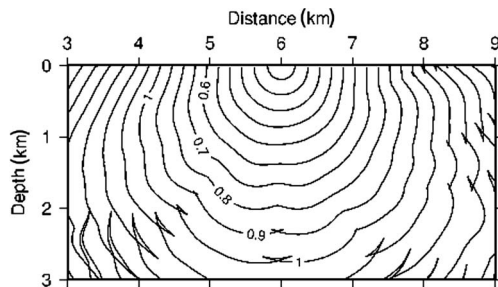


Figure 9. Wavefronts calculated using wavefront-oriented ray tracing. Time spacing between wavefronts is 0.1 s. From Coman and Gajewski (2005), their Figure 8.

model uniformly. Traveltimes between all points on the grid were calculated from a data set of 384×122 points sampling the velocity field at 24-m intervals along x and z . When determining the traveltime along a segment, we calculate slowness according to the nearest velocity point. The smoothed velocity model is plotted in Figure 6.

In Figure 7, we plot a set of traveltimes corresponding to first-arrival traveltimes from a source at (6.0 km, 2.8 km) to a set of 384 receivers, spaced evenly along the surface at 24-m intervals. We compare our results for a 101×101 grid to the results of Benamou (1996) in Figure 7 and see that results from our method are in close agreement. We notice slight deviations from the results of Benamou near the end of the line of receivers only. Specifically, our method seems to get caught in a local minimum near the end of the line of receivers, where Benamou's results seem to indicate crossings between wavefronts. We expect this can be remedied by increasing the number of sweeps or varying the cooling schedule. However, this is the only instance in which one observes this discrepancy; our result matches well with those of Benamou throughout the calculation. Also, we do not consider multipathing in our results. Without extra constraints, our method is suitable for calculating traveltime corresponding to first arrivals only. This property is intrinsic to the method of simulated annealing.

In addition, we compare shapes of wavefronts found here to the shapes of wavefronts calculated by Coman and Gajewski (2005) using wavefront-oriented ray tracing. In Figure 8, we plot the data points corresponding to the predicted shapes of wavefronts in 0.1-s intervals. Predicted shapes of wavefronts using our gridpoint data match closely with that of Coman and Gajewski, shown in Figure 9, aside from observing wavefront crossings.

Thus, although the rays we calculate are not as smooth as those calculated using other methods, traveltimes calculated by our method are at least as accurate as those calculated in other works using different methods.

CONCLUSION

Our primary purpose is to propose a new ray-tracing method based on simulated annealing. Velocity models presented here demonstrate the usability of the proposed method. Results from the Marmousi and elliptically anisotropic models show our method is suitable to calculate rays within smooth media to an accuracy above experimental error. Furthermore, we are able to exploit the fact that points on the grid can be mapped to specific areas of the model for increased accuracy — in this case, to points along an interface not parallel to the coordinate system of the underlying network. In other words, we are free to choose any grid pattern we desire. Results from the Marmousi model show that our method can produce accurate results in complex media as compared to results obtained from other methods.

Results presented in this work show that a bending method using simulated annealing shows promise. In view of the algorithm itself, extensions to 3D ray tracing are straightforward. The 3D Metropolis algorithm is, in principle, the same as the 2D algorithm; we specify an initial path and anneal the solution to the minimum by choosing points in a 3D grid, and we proceed through the Metropolis algorithm in the same manner.

The major drawback in moving to three dimensions with our algorithm can be seen by considering a cubic grid of side length L . Where the number of sweeps per temperature would scale with L^2 in two dimensions, our calculations will now scale like L^3 . Furthermore, the

amount of time necessary to calculate traveltimes between all points on the latter (and the amount of memory to store said calculations) would scale as L^6 . In our case, we would have to store more than 10,000 times more data before we could begin calculating rays. Given the amount of time needed to construct the rays for each model here, the algorithm in its current form is unsuitable for direct application to 3D media. We suggest an improvement to the algorithm in which one starts with a small number of grid points to obtain an estimate of the ray. Then one can increase the number of points around the ray gradually and use the previously calculated ray as the initial guess. In this way, we can solve for the desired ray recursively.

The current version of our code is not parallelized. An extension of our method to parallel computing would reduce the overall time needed to calculate a set of rays. One simple scheme would be to split the work needed to calculate a set of rays between each of the processors. For P processors, this would decrease the overall computational time to calculate N rays by P . Schemes that involve calculating one ray using P processors are not as straightforward, as they require extensive communication between each of the processors, which can reduce the effectiveness of parallel computing. Extensions to parallel computing will become important when we move to calculating rays in three dimensions.

Another extension to our method relates to the problem of constraining rays to pass through specified interfaces. Once these constraints are specified, we can choose configurations randomly, anneal each segment, and decide whether to accept a configuration according to the Boltzmann distribution. As one can see, our method can be generalized easily to M constraints in two or three dimensions by isolating specific areas to study. Furthermore, this need not be restricted to interfaces. We can specify a set of points to lie within a low-velocity region where rays that correspond to first arrivals would not pass through.

In addition, special cases such as a focusing reflector, in which the rays render the stationary traveltime as maximum, can be accommodated easily in our method by changing the transition probability to accept paths with longer traveltimes always. Furthermore, we have not taken into account the amplitudes of waves traveling along the rays. This would suggest the amplitude of the wavefront along a ray segment could be used as another parameter in the Metropolis algorithm. However, one would need to take into account how the amplitude of the entire wavefront would change by local perturbations to the ray. Amplitude considerations in this algorithm will be the focus of future work.

In addition, quenching at certain points during the algorithm can be used to obtain local minima of the traveltime; thus, the algorithm can be modified to find rays later than first arrivals.

We are aware that the presented algorithm can be adjusted further to suit particular applications better by optimizing the cooling schedule and the required equilibration times. Such improvements are not our purpose here; our main goal is to bring this new approach in ray tracing to the attention of interested researchers in the field.

ACKNOWLEDGMENTS

Research presented in this paper has benefited from the insightful comments, suggestions, and revisions of Phil Bording, David Dalton, Thormod Johansen, Tony Kocurko, Edward S. Krebes, Michael Rochester, and John Whitehead. In addition, this paper benefited from the editorial work of Leslie McNab. This research has been ac-

complished in the context of the Geomechanics Project at Memorial University. Research of AB and MAS was funded partially by the National Science and Engineering Research Council. Numerical results were obtained by using computational facilities hosted on the Mahone2 server of the ACEnet computer network.

REFERENCES

- Benamou, J. D., 1996, Big ray tracing: Multivalued travel time field computation using viscosity solutions of the eikonal equation: *Journal of Computational Physics*, **128**, 463–474.
- Bóna, A., and M. A. Slawinski, 2003, Fermat's principle for seismic rays in elastic continua: *Journal of Applied Geophysics*, **54**, 445–451.
- Černý, V., 1985, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm: *Journal of Optimization Theory and Applications*, **45**, 41–45.
- Červený, V., 2001, *Seismic ray theory*: Cambridge University Press.
- Chaikin, P., and T. Lubensky, 1995, *Principles of condensed matter physics*: Cambridge University Press.
- Coman, R., and D. Gajewski, 2005, Traveltime computation by wavefront-oriented ray tracing: *Geophysical Prospecting*, **53**, 23–36.
- Gibson, R. L., Jr., V. Durussel, and K.-J. Lee, 2005, Modeling and velocity analysis with a wavefront-construction algorithm for anisotropic media: *Geophysics*, **70**, no. 4, T63–T74.
- Julian, B. R., and D. Gubbins, 1977, Three-dimensional seismic ray tracing: *Journal of Geophysics*, **43**, 95–114.
- Kirkpatrick, S., C. D. Gelatt, Jr., and M. P. Vecchi, 1983, Optimization by simulated annealing: *Science*, **220**, 671–680.
- Koketsu, K., and S. Sekine, 1998, Pseudo-bending method for three-dimensional seismic ray tracing in a spherical earth with discontinuities: *Geophysical Journal International*, **132**, 339–346.
- Lambaré, G., P. S. Lucio, and A. Hanyga, 1996, Two-dimensional multivalued traveltime and amplitude maps by uniform sampling of a ray field: *Geophysical Journal International*, **125**, 584–598.
- Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, 1953, Equation of state calculations by fast computing machines: *Journal of Chemical Physics*, **21**, 1087–1092.
- Moser, T. J., 1991, Shortest path calculation of seismic rays: *Geophysics*, **56**, 59–67.
- Moser, T. J., G. Nolet, and R. Snieder, 1992, Ray bending revisited: *Bulletin of the Seismological Society of America*, **82**, 259–288.
- Newman, M., and G. Barkema, 2002, *Monte Carlo methods in statistical physics*: Oxford University Press.
- Nourani, Y., and B. Andersen, 1998, A comparison of simulated annealing strategies: *Journal of Physics A: Mathematical and Theoretical*, **23**, 8373–8385.
- Pereyra, B., W. H. K. Lee, and H. B. Keller, 1980, Solving two-point seismic-ray tracing problems in a heterogeneous medium: Part 1 — A general adaptive finite difference method: *Bulletin of the Seismological Society of America*, **70**, 79–99.
- Prothero, W. A., W. J. Taylor, and J. A. Eickemeyer, 1988, A fast, two-point, three-dimensional ray tracing algorithm using a simple step search method: *Bulletin of the Seismological Society of America*, **78**, 1190–1198.
- Rogister, Y., and M. A. Slawinski, 2005, Analytic solution of ray tracing equations for linearly inhomogeneous and elliptically anisotropic velocity model: *Geophysics*, **70**, no. 5, D37–D41.
- Sadeghi, H., S. Suzuki, and H. Takenaka, 1999, A two-point, three-dimensional seismic ray tracing using genetic algorithms: *Physics of the Earth and Planetary Interiors*, **113**, 355–365.
- Slawinski, M. A., 2003, *Seismic waves and rays in elastic media*: Elsevier Pergamon.
- Smith, P., 2006, On forward and inverse modeling in seismology: Raytracing in inhomogeneous media: M. S. thesis, Memorial University.
- Um, J., and C. Thurber, 1987, A fast algorithm for two-point seismic ray tracing: *Bulletin of the Seismological Society of America*, **77**, 972–986.
- Van Laarhoven, P., and E. Aarts, 1987, *Simulated annealing: Theory and applications*: D. Reidel.
- Velis, D., and T. Ulrych, 1996, Simulated annealing two-point ray tracing: *Geophysical Research Letters*, **23**, 201–204.
- , 2001, Simulated annealing ray tracing in complex three-dimensional media: *Geophysical Journal International*, **145**, 447–459.
- Versteeg, R., and G. Grau, eds., 1991, *The Marmousi experience: Proceedings of the 1990 EAEG workshop on practical aspects of seismic data inversion*.

- Vinje, V., K. Astebøl, E. Iversen, and H. Gjøystdal, 1996a, Estimation of multivalued arrivals in 3D models using wavefront construction — Part I: Geophysical Prospecting, **44**, 819–842.
- , 1996b, Estimation of multivalued arrivals in 3D models using wavefront construction — Part II: Tracing and interpolation: Geophysical Prospecting, **44**, 843–858.
- , 1999, Three-dimensional wave modeling by wavefront construction in open models: Geophysics, **64**, 1912–1919.
- Vinje, V., E. Iversen, and H. Gjøystdal, 1993, Traveltime and amplitude estimation using wavefront construction: Geophysics, **58**, 1157–1166.