GEOPHYSICS, VOL. 71, NO. 3 (MAY-JUNE 2006); P. T41–T51, 16 FIGS., 1 TABLE.
10.1190/1.2192948

Block modeling and segmentally iterative ray tracing in complex 3D media

Tao Xu[1], Guoming Xu[2], Ergen Gao[2], Yingchun Li[2], Xianyi Jiang[3], and Kaiyun Luo[3]

ABSTRACT

We propose using a set of blocks to approximate geologically complex media that cannot be well described by layered models. Interfaces between blocks are triangulated to prevent overlaps or gaps often produced by other techniques, such as B-splines, and to speed up the calculation of intersection points between a ray and block interfaces. We also use a smoothing algorithm to make the normal vector of each triangle continuous at the boundary, so that ray tracing can be performed with stability and accuracy. Based on Fermat's principle, we perturb an initial raypath between two points, generally obtained by shooting, with a segmentally iterative ray-tracing (SIRT) method. Intersection points on a ray are updated in sequence, instead of simultaneously, because the number of new intersection points may be increased or decreased during the iteration process. To improve convergence speed, we update the intersection points by a first-order explicit formula instead of traditional iterative methods. Only transmitted and reflected waves are considered. Numerical tests demonstrate that the combination of block modeling and segmentally iterative ray tracing is effective in implementing kinematic two-point ray tracing in complex 3D media.

INTRODUCTION

Two-point ray tracing is crucial to seismic tomography, earthquake location, seismic acquisition, and survey design. Previously reported methods include shooting (Langan et al., 1985; Virieux and Farra, 1991; Sun, 1993; Sambridge et al., 1995) and bending (Julian and Gubbins, 1977; Aki and Richards, 1980; Thurber and Ellsworth, 1980; Um and Thurber, 1987; Prothero et al., 1988; Pereyra, 1992; Mao and Stuart, 1997; Gao et al., 1998). Other methods include wavefront techniques (Vinje et al., 1993, 1996), graph theory (Moser, 1991; Fischer and Lees, 1993; Zhang et al., 2000; Zhao et al., 2004) and simulated annealing (Velis and Ulrych, 1996, 2001).

Most but not all of the above methods are based on models parameterized in cells (or grids) (Langan et al., 1985; Moser, 1991) or layers (Zelt and Smith, 1992; GuiZiou et al., 1996; Mao and Stuart, 1997; Rawlinson et al., 2001). When divided into fine enough cells or grids, the model can be a good approximation to reality. However, usually the number of cells or grid points is reasonably large — for example, 11 875 (125 × 95) cells (Langan et al., 1985) and 2500 (50 × 50) grids (Moser, 1991) in two dimensions. The computation time of the algorithms is almost linearly proportional to the number of nodes in the models when traveltimes are calculated for all nodes (Moser, 1991); therefore, ray tracing can be very time consuming for these models, especially in three dimensions. In some situations, a horizontally layered model is fairly effective in describing geologic structures and is especially convenient for ray tracing. However, in such models, layer boundaries cannot cross each other. For example, in a layered model, a ray must pass through overburden layers in sequence until it meets the given reflecting interface; it then returns through the same layers in reverse order. Such a requirement cannot be met in situations such as the one in which a ray crosses the fault on the right side of point A in Figure 1. To overcome this difficulty, pseudointerfaces (dashed lines in Figure 1) are introduced, a process that sometimes suffers from overlapping boundaries. In the case of more complex geologic structures (Figure 2), layered parameterization is very difficult.

Instead, Gjøystdal et al. (1985) use a solid modeling technique to generate a 3D model. The term solid modeling refers to the fact that the internal geometrical properties of the model can be modeled as a combination of solids or volumes in 3D space. However, this algorithm defines complex regions using counterintuitive, set

Manuscript received by the Editor March 16, 2003; revised manuscript received August 24, 2005; published online May 19, 2006.

[1]Chinese Academy of Sciences, Institute of Geology and Geophysics, State Key Laboratory of Lithosphere Evolution, No. 19, Beituchengxi Road, Chaoyang District, Beijing 100029, China. E-mail: xutao@mail.iggcas.ac.cn.
[2]University of Science and Technology of China, School of Earth and Space Sciences, no. 96, Jinzhai Road, Anhui Province, Heifei 230026, China. E-mail: xugm@ustc.edu.cn.
[3]Bureau of Geophysical Prospecting, Geophysical Research Institute, CNPC, Zhuozhou, Hebei, China.

theoretical operations on the volumes limited by simpler surfaces — an algorithm improved and made more intuitive and natural by Pereyra (1992). Requiring patch linking without gaps, as described below, results in block models that describe relatively simple cases as well as some complex cases, such as pinch-out layers and salt domes with overhangs (mushroom model).

In many applications, realistic 3D models must be constructed to represent very complex structures. In this paper, we further develop the solid modeling method for space parameterization. Our approach is based on that of Pereyra (1992), and our major improvement is in the description of interfaces. Based on block models, we also develop a segmentally iterative method for ray tracing, which is a variant of the bending technique. The method is effective for ray tracing complex model structures.

MODEL PARAMETERIZATION

After long tectonic evolution, most geologic units become very complex, and in only fairly idealized cases can they be represented by relatively simple structures (Figure 2). In this model, most local volumes can be regarded as homogeneous, with changes occurring discontinuously at volume boundaries. Such media would be better approximated as so-called blocks as opposed to layers. In this paper, a block model is described as an aggregate of arbitrarily shaped blocks or volumes separated by interfaces.

Xu et al. (2001) have studied block parameterization in two dimensions. The structure of these media is represented hierarchically as area→element→edge→segment→point. In three dimensions, we describe block structure as volume→block→interface →triangle→point. Three-dimensional geologic media are described as an aggregate of geologic blocks, each with its own geologic attributes, such as density and seismic-wave velocity. Geologic blocks are separated by interfaces, which are described by several discrete points and are triangulated in our scheme. The geologic media parameterized in the structure mentioned above are represented uniquely, as described in the following section.

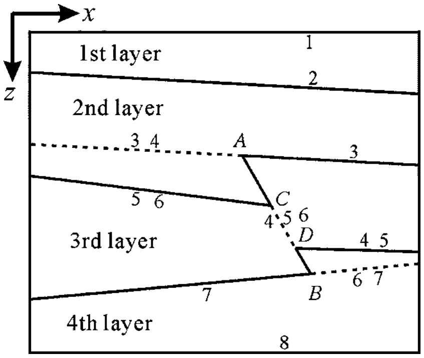

Figure 1. Pseudointerfaces (dashed lines) should be added for the layer-based reverse fault model. Note that the layer orders are described; some interfaces overlap.

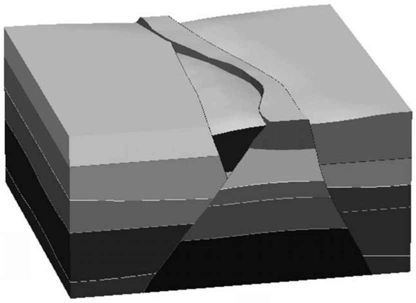

Figure 2. The complex model is an aggregate of blocks with triangulated interfaces. Different blocks are shown in different shades of gray. Layer parameterization is hardly possible in this complex model.

Interface representation

In block representation, the key implementation is interface (→triangle)→point, which can be done with Coons, Bezier, B-spline, and triangulated surface patches. Before going into detail, we list a few requirements for a good ray-tracing algorithm:

1) The complex 3D media can be efficiently parameterized.
2) Surface patches should be linked smoothly and gaps should not be generated in patch links.
3) Ray tracing should not be very time consuming, so the intersection point between a ray and a surface patch should be obtained rapidly.

Coons, Bezier, and B-spline surface patches are usually interpolated according to rectangularly spaced nodes, though they can be defined on nonrectangular domains in parametric form, such as a correspondence between the physical domain in the (x,y) plane and the unit square in parameter space (u,v) (Pereyra, 1996). Coons patches pass through all nodes, while Bezier and B-spline patches do not. B-spline patches have better geometric properties than Coons patches, and the polynomial order does not increase necessarily with the number of nodes, as it does with Bezier patches. Therefore, B-spline patches are widely applied to represent geophysical interfaces (for analytical expressions of such surface patches, see Pereyra, 1992, 1996). The main advantage of these representations is that the surfaces are everywhere continuous in curvature, since C_2 continuities are beneficial to ray-tracing methods that rely on nearby ray trajectories to vary smoothly to find an optimal solution.

However, there are two problems in the representation of these patches. One of them is the issue of interface linking. Consider the B-spline interface in which two interfaces are constructed based on rectangular nodes $n_1 \times m_1$ and $n_2 \times m_2$. In the linked direction, the number of control vertices should be equal (e.g., $n_1 = n_2$) and the line of control vertices identical. Such a situation results in only C_0 continuous linking, and smoother interfaces (e.g., C_2 continuous) require stricter constraints. Otherwise, gaps are generated (Wang and Yang, 1990) and ray tracing is difficult to implement. Vinje et al. (1999) present the open model to describe an interface containing holes or cracks. They also present a ray-tracing method with associated wavefront construction to address open models. However, the method is nonunique in determining geologic properties beside a hole or crack.

The other issue is that obtaining the intersection point between a ray and an interface interpolated with splines is very time consuming. Generally, intersection points are obtained by an iterative procedure, which is more time consuming than using analytical expressions. Some authors use strategies based upon a box hierarchy (Virieux and Farra, 1991) or a generalized Newton method (Rawlinson et al., 2001).

Triangulated interfaces

We avoid these problems above by using triangulation to represent geologic interfaces. For any given set of discrete nodes in a geologic interface, there is at least one scheme to construct an interface with a set of triangles of different sizes without any gaps, such as Delaunay triangulation (Delaunay, 1934). As an example, Figure 2 displays a complex block model, where different blocks are represented as different shades of gray. Interfaces in this model are constructed with 6676 triangles. Since the intersection between a line and triangle can be computed analytically, the large number of ray/interface intersections can be computed quickly. In addition, increasing or decreasing the number of nodes and reconstructing the interface with the remaining nodes is easy to implement, a process that is very useful for modifying or eliminating inaccurate geophysical nodes. Triangulated interfaces are also applied in the well-known GOCAD system (Mallet, 1989, 1992).

However, triangulated interfaces are less smooth. Since each triangle has a normal vector that is determined by the three vertices of the triangle, normal vectors vary abruptly across linked boundaries of two triangles that are not in the same plane. Ray tracing in such a situation is quite difficult, where a reflected or transmitted ray may change direction abruptly across linked boundaries.

To solve the problem, we propose a new algorithm to redefine normal vectors at arbitrary points on the interfaces. The scheme is implemented in two steps. First, the normal vector at each vertex is specified; then, the normal vector at an arbitrary point in a triangle is obtained by interpolation among the three normal vectors at the triangle's vertices. For similar techniques of smoothing normal vectors, see Zelt and Smith (1992) and Vinje et al. (1999).

The normal vector at each vertex is estimated by linked triangles on the interfaces. Figure 3 illustrates a simple interface. For example, vertex A is linked to six triangles marked with number symbols, and the normal vector in vertex A can be estimated by the formula

$$\mathbf{n}_A = \sum_{i=1}^{N} \left(\frac{s_i \mathbf{n}_i}{d_i} \right), \qquad (1)$$

where N is the number of linked triangles, which is equal to six for vertex A, $\mathbf{n}_i$ denotes the normal vector of each plane triangle, s_i denotes the area of each triangle, and d_i denotes the distance between vertex A and the central point of each triangle.

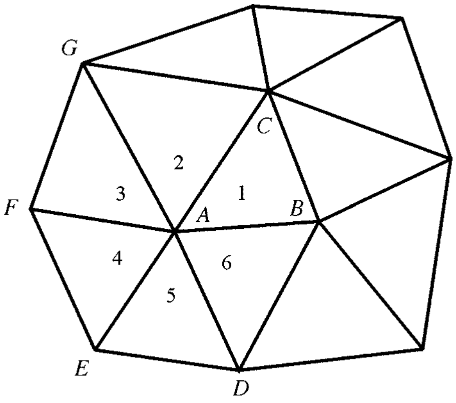

Figure 3. The normal vector in vertex A is determined by six linked triangles. The normal vectors are smoothed inside the triangles and across the boundary of two adjacent triangles.

Normal vectors at all vertices can be calculated in the same way. Note that the summation is composed of six items in the computation of vertex A, while it is composed of two items for vertex E and F, because only two triangles are linked to them on this interface.

The second step is linear interpolation to obtain the normal vector at an arbitrary point on the surface. For example, the normal vector at an arbitrary point in triangle ABC of Figure 3 can be described as $\mathbf{n} = \Sigma_{i=1}^{3}(u_i\mathbf{n}_i)$, where u represents the area coordinate of the point in triangle ABC. (For a description of area coordinates, see Appendix A and Chang, 1995), and $\mathbf{n}_i$ represents normal vectors at vertices A, B, C.) The normal vector in the triangle changes linearly with the change of the area coordinate u, which corresponds to the point's position. As a result, the normal vector changes smoothly in the triangle, as well as on the three edges of the triangle; e.g., the normal vector at a point on edge AB is the same, whether it is calculated from the triangle ABC or the triangle ABD. Therefore, normal vectors are continuously varying on the whole surface.

Vinje et al. (1999) use a similar representation of triangular interfaces without explaining how to obtain the normal vectors at the vertices. Linear interpolations are used to estimate the spatial position and normal vector within a triangle, which is essentially the same as interpolation by area coordinate, but the latter has good geometric attributes, as described in Appendix A.

Inevitably, the representation of a triangular interface has errors since it is not a true C_2 patch surface. It is only approximately C_2, and the computed oblique normal is not perpendicular to the plane triangles. Sometimes, for a sparsely sampled and rapidly fluctuating interface, an analytically interpolated normal at a vertex may have direction opposite to others, which may lead to an incident ray emerging on the wrong side of the interface (i.e., a transmitted ray passes back into the previous block or a reflected ray passes through into the next block). Generally, errors are rare if the interface is oversampled, but they are difficult to avoid completely.

RAY-TRACING METHOD

In ray tracing with the shooting method, updating take-off angles and computing new raypaths are very time-consuming tasks, especially in complex media. To enhance efficiency, we have developed a segmentally iterative ray-tracing (SIRT) method, based on Fermat's principle of stationary time, to suit block models (SIRT is a well-established name in geophysical inversion, but we still use this acronym). The SIRT algorithm is shown schematically in Figure 4. Consider the transmitted wave, for which ray tracing requires an initial raypath $RP_1P_2, \dots, P_{n-1}P_nS$. For the whole raypath to satisfy Fermat's principle, any successive three intersection points of the raypath must also satisfy the stationary traveltime principle. This observation leads to an approximate modifying formula. The formula is the same whether the midpoint is a reflection or a transmission point. From the source or receiver, three successive points R, P_1, P_2 are selected to obtain the new midpoint P_1' by the modifying formula. Replace P_1 with P_1', and the next three successive points P_1', P_2, P_3 are selected. Adjust the intersection points in the same way, and the adjusting process progresses until meeting the end point S. One iteration produces a new raypath $RP_1'P_2'$, $\dots, P_n'S$, and another iteration progresses in the same way. Ray tracing terminates when the modifying values of reflection (or transmission) points meet the accuracy requirement (typically a precision of 1 m is required in a 5000 × 5000 × 5000 m model).

SIRT is based on an initial-guess raypath, generally obtained by shooting. Our shooting scheme is similar to the first step of the four-step shooting method (Rawlinson et al., 2001). The scheme involves shooting an approximate spread of rays in constant increments of θ (typically $\sim 1°$) and ϕ (typically $\sim 5°$), the ray inclination and azimuth, respectively, at the source. The emergence points can form several triangles (Figure 5). Target receiver P falls inside triangle ABC, whose vertices are the emergence points of three adjacent rays. If the distance between receiver P and the nearest point B is less than the required accuracy (typically 1–5 m), the shooting raypath can be regarded as true. Otherwise, we substitute the position of the receiver for that of point B and implement SIRT. If the model is not too complex, an initial-guess reflection point can be obtained by some simple method, and the reflection point can be connected to the source and the receiver by straight lines. That means the initial raypath comprises straight lines to save shooting time.

SIRT cannot solve multiple paths. One initial-guess ray converges to only one ray. Furthermore, two emergence triangles constructed by adjacent rays may overlap, and a receiver can fall into two or more emergence triangles. Different initial trajectories from a receiver may converge to different minima or one minimum, which may be global or local. As a result, several paths from a receiver may result from our tracing scheme.

Modifying formula

As discussed above, application of Fermat's principle to three successive points along a raypath results in a formula for modifying the midpoint during SIRT. Figure 6 is a sketch of the modification of a midpoint.

The position in a triangle on an interface is described with two parameters s and t,

$$x_i = x_i(s,t), \quad (i = 1,2,3). \tag{2}$$

The parameters (s, t) correspond to (0, 0), (1, 0), and (0, 1) at the three vertices. Values of x and y (maybe y and z, z, and x) corresponding to parameters s and t lie in the triangle.

In Figure 6, the wave velocity of the upper block is v_1 and in the lower is v_2. Two successive segment lengths are described as l_1 and l_2, and traveltime is described as

$$T = \frac{l_1}{v_1} + \frac{l_2}{v_2},$$

where

$$l_1 = \left[\sum_{i=1}^{3} (x_i - x_i^{(1)})^2\right]^{1/2}, \quad l_2 = \left[\sum_{i=1}^{3} (x_i - x_i^{(3)})^2\right]^{1/2}. \tag{3}$$

In the expression $x_i^{(j)}$, j denotes three successive points, and i denotes three coordinates of one point. The conditions for stationary traveltime are

$$\left.\frac{\partial T}{\partial s}\right|_{(s=s^*,t=t^*)} = 0, \quad \left.\frac{\partial T}{\partial t}\right|_{(s=s^*,t=t^*)} = 0. \tag{4}$$

Using only the first term of a Taylor series, we obtain the final modifying formula, which is the same whether the midpoint is a reflection or transmission point:

$$\Delta s = \frac{U_{22}d_1 - U_{12}d_2}{\Delta}, \quad \Delta t = \frac{U_{11}d_2 - U_{21}d_1}{\Delta}. \tag{5}$$

(For details, see Appendix B.)

Substitute the position $(s + \Delta s, t + \Delta t)$ for the primary position (s,t) directly if the new midpoint falls on the original interface. Otherwise, additional judgment is needed to determine whether intersection points should be added or removed.

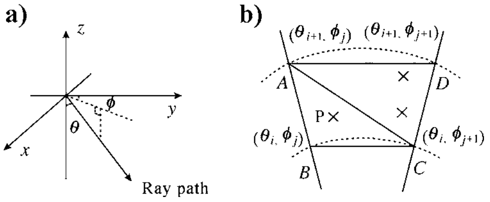

Figure 5. (a) Ray direction parameters at source θ (inclination) and ϕ (azimuth). (b) Receiver P (cross) falls into the triangle formed by adjacent emergence points A, B, and C.

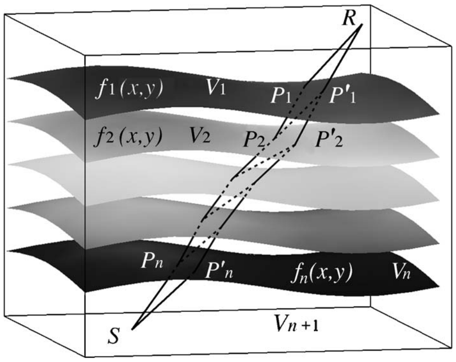

Figure 4. The sketch illustrating segmentally iterative ray tracing. The node points along the raypath are modified segmentally.

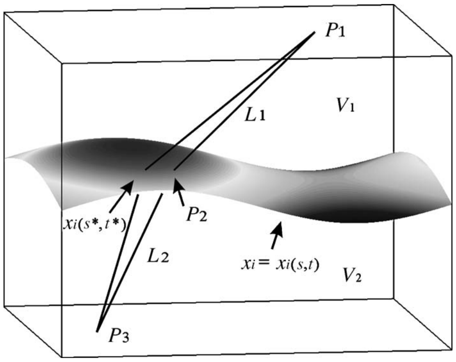

Figure 6. The sketch for modifying the middle point of two segments of a trajectory.

Adding or removing intersection points during iteration

In block models, some new problems arise from SIRT. The number of intersection points in a trajectory can be increased or decreased during the course of an iteration, a phenomenon that does not occur in layered models. Figure 7 indicates the trajectory change from $P_0P_1P_2$ to $P_0P_0'P_1'P_2$ after an iteration.

There are two blocks on each side of every interface, and one block beside a boundary interface can be regarded as null. Obviously, there is a common block beside two interfaces, which two successive intersection points determine. For example, block 0 is the common block beside interface 0 and interface 1. In the three successive intersection points $P_0P_1P_2$, midpoint P_1 situated on interface 1 is updated by the modifying formula and a new midpoint P_1' is obtained, which is situated on a new interface 2. If the blocks beside interface 1 and interface 2 were identical, we would not need to add or remove intersection points. However, such a situation does not exist in this case. Furthermore, since no common block is beside interface 0 and interface 2, intersection points must be added between point P_0 and P_1'. A ray connecting P_0 to P_1' intersects interface 4 at point P_0'. However, there is a common block (block 2) beside interface 2 and interface 3; hence, no intersection point should be added or removed. As a result, the segmental trajectory $P_0P_0'P_1'P_2$ is the updated raypath of $P_0P_1P_2$. Of course, further operations are required for more complex cases, such as when a lens (block 3) exists inside block 2. Then, the program needs to decide whether the straight line connecting point P_1' with point P_2 intersects other interfaces in block 2. If an intersection point exists, additional intersection points must be added.

The converse variation of changing the segmental trajectory from $P_0P_0'P_1'P_2$ to $P_0P_1P_2$ is similar but is not discussed here.

Adaptability of SIRT in block models

In general, bending methods involve perturbing an initial ray trajectory in some way, such that, by iterations, some criterion converges to a minimum. There are usually two cases. One is the issue of perturbing trajectories in a continuous medium. For example, Julian and Gubbins (1977) adopt finite-difference approximations to second-order differential equations; Pereyra et al. (1980) adopt nonlinear first-order differential equations and solved jump discontinuities across interfaces; and Um and Thurber (1987) develop an algorithm by using a geometric interpretation of the ray equations. The other issue involves perturbing trajectories across discontinuous boundaries. Mao and Stuart (1997) update all path points simultaneously by a quasi-linearization equation. Such an updating equation is not efficient in block models because the number of intersection points may vary during an iteration, as described above. To overcome this problem, we update intersection points segmentally and add or remove intersection points as necessary. SIRT has similar applicability to the perturbation scheme of Zhao et al. (1992). Based on Snell's law, intersection points are updated directly by a first-order explicit formula instead of using an iterative method (called the bisection method) in the case of discontinuous boundaries. Since ray-tracing schemes involve modifying many intersection points, our tracing scheme saves considerable time.

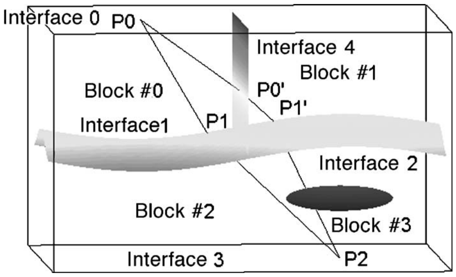

Figure 7. A segmental trajectory changes from $P_0P_1P_2$ to $P_0P_0'P_1'P_2$ after one iteration.

Blocks in this paper comprise homogeneous media with the same geologic attributes (density, seismic wave velocity, etc.), and rays are straight. Heterogeneous media can be approximated by several small blocks, each of which is homogeneous — a process that is inefficient if the media have constant gradients. In that case, the ray trajectory in that medium is a segment of arc and can be described analytically (Rawlinson et al., 2001). SIRT could also be applied in the same way; the main difference is the modification of midpoints, which is much more complex and is not discussed here.

SYNTHETIC DATA EXAMPLES

We present three typical block models (Figures 2 and 8) to illustrate the SIRT method. These models, which are contained within cubes, are all composed of several blocks separated by triangulated interfaces. Figure 8 shows a reverse fault model and a combination model with associated velocity structures. Both models have dimensions of 5000 × 5000 × 5000 m. Model 1 has 4 blocks and 3368 triangles. Model 2, composed of normal faults, reverse faults,

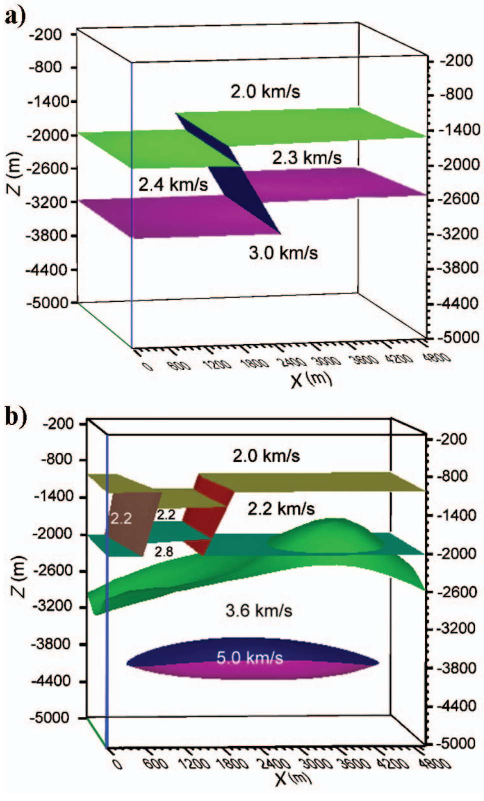

Figure 8. Two block models with associated velocity structures. (a) The reverse fault model 1 has 4 blocks and 3368 triangles. (b) The combination model 2, composed of normal faults, reverse faults, an intrusive mass, and a lens, has 7 blocks and 4649 triangles.

an intrusive mass, and a lens, has 4 blocks and 4649 triangles. Figure 2 shows a complex model (model 3) with a size of 10 000 × 10 000 × 5000 m; it has 18 blocks and 6676 triangles. To avoid undue complexity, the velocity structure is omitted here.

Ray-tracing results

We used three block models mentioned above for ray-tracing tests. Some of the source-receiver pairs designed for these models are shown in Figure 9. The source-receiver pairs are located on the earth's surface, and the locations of the sources and receivers are indicated by stars and triangles, respectively. The solid source-receiver pairs in Figure 9a are used for model 1, and the hollow pairs are used for model 2; the pairs used for model 3 are in Figure 9b. Note that the surface size is 10 000 × 5000 m in model 3. For each case, 360 receivers are arranged in a 6 × 60 rectangle. In our tests, two lower-layer interfaces in model 1 and the upper interface of the lens in model 2 are defined as reflecting interfaces.

Shooting rays are selected as initial raypaths for SIRT in our scheme. The approximate ranges of θ (inclination) and ϕ (azimuth) should be chosen differently for different models and associated source-receiver pairs. The ranges of θ and ϕ are typically 10° ∼35° and 60°∼50°, respectively, with increments of 1° and 5°, respectively, in model 1 and 3°∼10° and 100°∼240°, respectively, with the same increments in model 2.

In ray-tracing model 1 (Figure 10a), the blue part of each ray is the incident ray, and the red ray is the reflected ray. Figure 10b shows the associated traveltimes plotted against distance, which is the receiver y-coordinate. To illustrate the necessity of smoothing interfaces, we ray trace the same model with smoothed interfaces (Figure 11a) and unsmoothed interfaces (Figure 11c). In comparing the associated traveltimes, we note continuous results for smoothed interfaces (Figure 11b) and sporadic results for unsmoothed interfaces (Figure 11d). Without smoothing, the use of a triangular mosaic to describe an interface introduces ray shadow zones because of gradient discontinuities at sutures between adjacent triangles.

Figure 12 shows the raypaths and associated traveltimes for model 3. The second geologic layer (yellow) is defined as the reflecting interface. Note that the surface of model 3 is fluctuant, while those of model 1 and model 2 are planar.

Our modeling approach can be applied to seismic acquisition or survey design. This application requires software that performs ray tracing rapidly and can be applied to complex media. Various factors influence ray-tracing computation time in three dimensions — factors such as media complexity, tracing precision, the number of source-receiver pairs, and computer speed. In the ray-tracing examples above, the precision requirement is 1 m, which is the maximum modifying value equal to $\sqrt{(\Delta x)^2 + (\Delta y)^2}$, corresponding to $\sqrt{(\Delta s)^2 + (\Delta t)^2}$ in an entire trajectory.

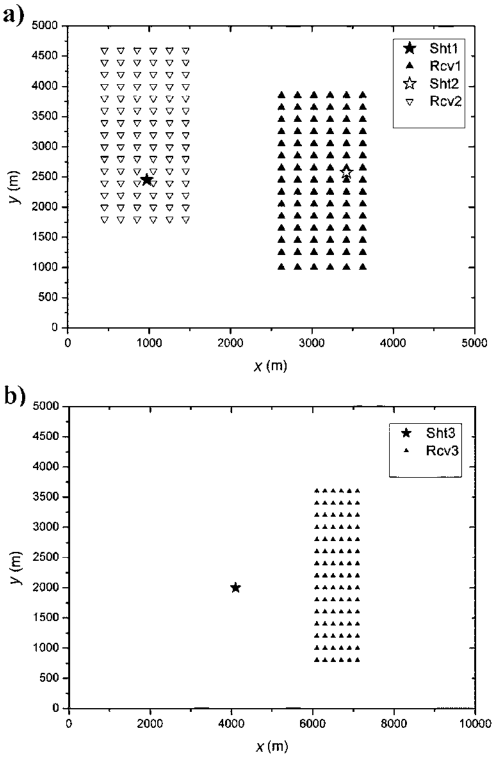

Figure 9. The locations of sources and receivers, indicated by stars and triangles, respectively. (a) The solid source-receiver pairs are designed for model 1 and the hollow pairs are designed for model 2. (b) The solid pairs are designed for model 3. Only a subset of the receivers is shown.

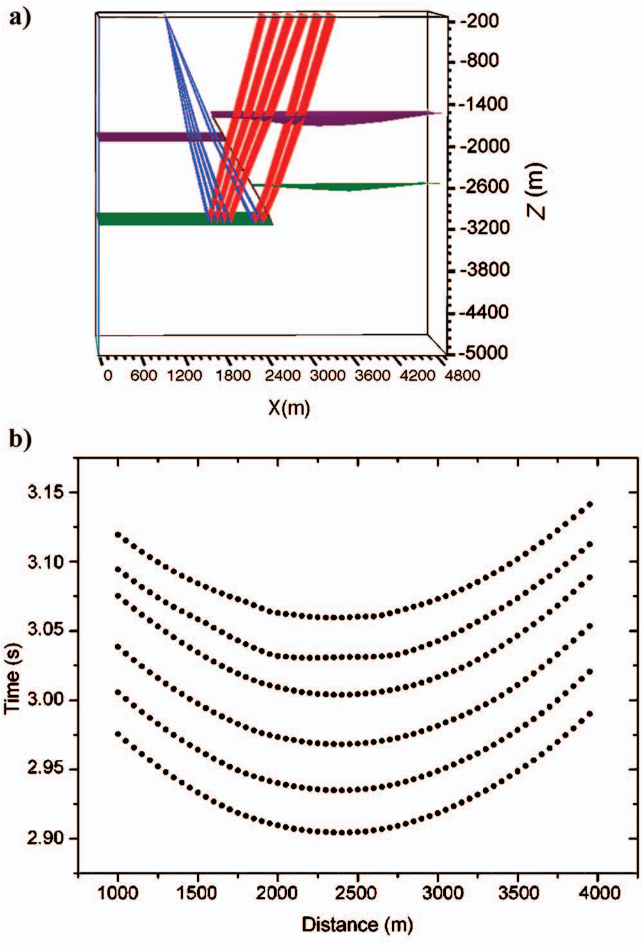

Figure 10. (a) The ray-tracing result using one source and 360 receivers in model 1. The blue part of a ray is incident, and the red is reflected. The lower layer interface is the reflecting interface. Negative values indicate depth beneath the surface. (b) Associated traveltimes.

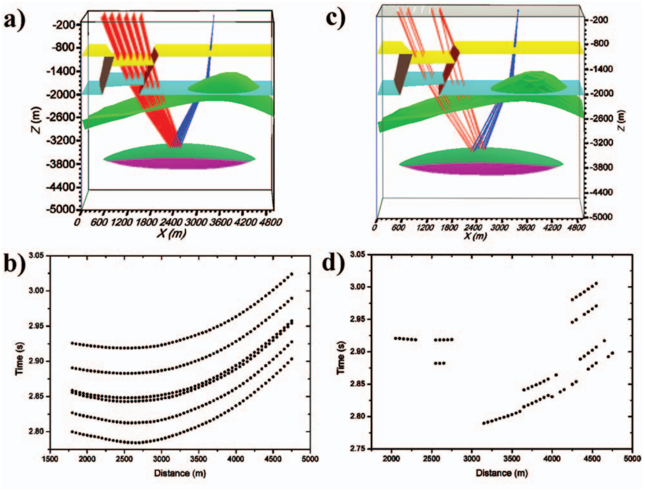

Figure 11. (a) Ray-tracing results with one source and 360 receivers in model 2. The upper interface of the lens is defined as the reflecting interface. (b) The associated traveltimes of (a). (c) The ray-tracing result with no interfaces smoothed. (d) The associated traveltimes of (c).

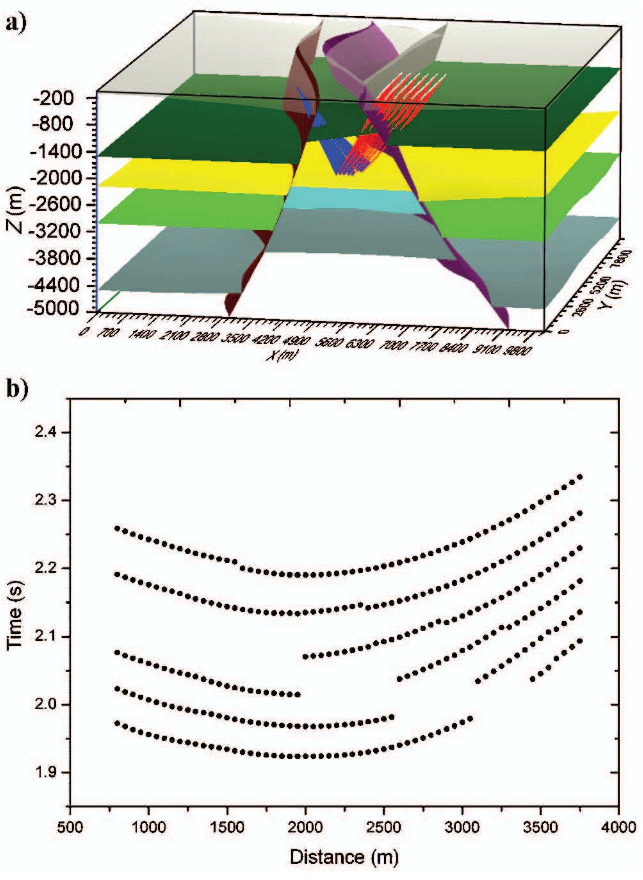

Figure 12. (a) The ray-tracing result with one source and 360 receivers in model 3. The semitransparent surface is fluctuant. The second layer (in yellow) is defined as the reflecting interface. (b) Associated traveltimes.

Next, we present a set of tests that compares the adaptability and calculating speed of SIRT with those of the shooting method.

Comparison with shooting method

It is very difficult to make a fair comparison between various methods in adaptability and calculating speed because different raypaths result from different model representations. We compare SIRT to the shooting method based on model 2 (Figure 8) and source-receiver pairs in Figure 13. The bottom receiver is labeled number one.

After the shooting procedure, the fifteen receivers are located at fifteen so-called primary triangles. The vertices of a primary triangle are three adjacent emergence points. The fifth, sixth, and seventh receivers are located in three primary triangles (Figure 13). Take-off angles are updated iteratively, a process that terminates when a ray ends within a given distance from its target. If the take-off angles for three vertices are described as (θ_i, ϕ_i), $i = 1,2,3$, and the area coordinate of a receiver in the triangle is described as u_i, then the strategy for updating take-off angles is given by

$$\theta = \sum_{i=1}^{3} \theta_i u_i, \quad \phi = \sum_{i=1}^{3} \phi_i u_i. \tag{6}$$

Table 1 shows the change in distance d with the number of iterations, where d is the distance between the emergence point and a receiver after an iteration.

SIRT selects the ray whose emergence point is nearest to a receiver and then replaces the endpoint position of the ray with that of the receiver. The modified ray is regarded as the initial-guess ray. In the subsequent segmental iteration, $(\Delta s_i, \Delta t_i)$, $i = 1,2, \ldots ,n$ are modifying values of midpoints, where $n + 2$ is the number of whole raypath points; $(\Delta x_i, \Delta y_i)$ are corresponding modifying values. A modifying distance d of a midpoint is described as $d = (\Delta x_i^2 + \Delta y_i^2)^{1/2}$, and the largest modifying distance $d_{\max}$ is the maximum modifying distance during an iteration. Table 1 shows the change in $d_{\max}$ with the number of iterations compared to distance d for shooting.

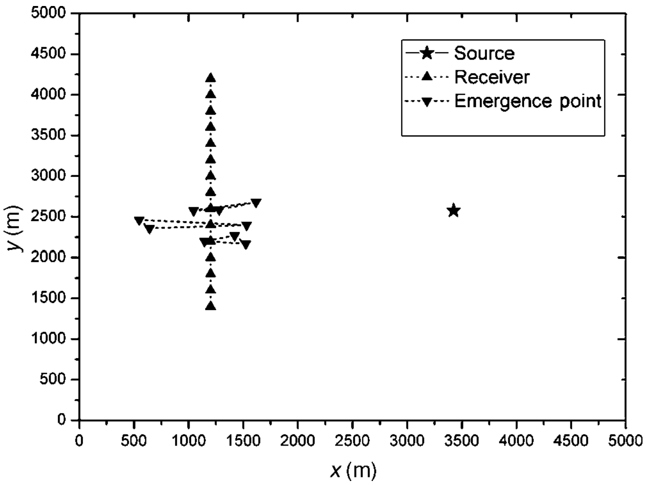

Figure 13. The locations of a source and 15 receivers are indicated by solid stars and upright triangles, respectively. The inverted triangles are primary emergence points

Table 1. The distance *d* between emergence points and a receiver in shooting compared to the maximum modifying distance d_{max} in SIRT after *N* iterations.

	Receiver 5		Receiver 6		Receiver 7	
Iteration times	d(m) Shooting	d_{max}(m) SIRT	d(m) Shooting	d_{max}(m) SIRT	d(m) Shooting	d_{max}(m) SIRT
1	93.9884	15.4335	252.918	106.216	125.612	14.9793
2	147.240	6.66932	212.620	17.3914	52.1205	10.1628
3	380.846	5.42493	200.481	12.9176	37.4033	8.34367
4	408.909	4.61834	522.152	12.0007	24.0301	7.12858
5	155.147	3.98158	509.200	11.0424	16.8187	6.18498
6	158.722	3.44501	506.336	10.0900	12.2859	5.39240
7	160.985	2.98441	505.692	9.17271	9.23171	4.70551
8	411.045	2.59024	505.547	8.64549	7.07327	4.10417
9	161.530	2.24708	505.514	8.43965	5.49594	3.57667
10	161.865	1.94813	505.507	8.19567	4.31469	3.11426

Figure 14 displays the relative traveltime errors at three receivers when calculated by shooting and SIRT, respectively. The relative errors are calculated using the formula $100 \times (t - t_{min})/t_{min}$, where minima t_{min} are obtained by running SIRT with 100 iterations and the largest modifying distance precision is equal to 0.01 m.

Table 1 displays ray-tracing performance for three receiver locations with increasing iterations. Note that SIRT converges more efficiently than does the shooting method. Shooting does not converge for receiver 5 and converges to a nonminimum value for receiver 6. The associated traveltime errors (Figure 14) show the same point. However, SIRT has good convergence for these receivers. For receiver 7, both methods converge, but SIRT is more accurate when more iteration times are chosen.

The updating scheme for the shooting method is linear modification of θ and ϕ, a scheme that sometimes does not converge robustly for a general complex medium. However, SIRT modifies the intersection points of an initial raypath segmentally. The local modification is relatively linear as opposed to shooting. A raypath converges faster when closer to the real one in SIRT.

The traveltimes at 15 receivers (Figure 15) show poor solutions from shooting in complex media. The upright triangles denote traveltimes obtained by SIRT and the reverse ones by shooting. Seven receivers have no ray-tracing results from shooting (Figure 15), while all receivers have ray-tracing results from SIRT. The traveltimes from the two methods are close for those receivers that have solutions.

The CPU time (PC, Pentium III 733 MHz) for the shooting method is 3.69 s and for SIRT is 1.81 s, showing that SIRT is faster than shooting in this case.

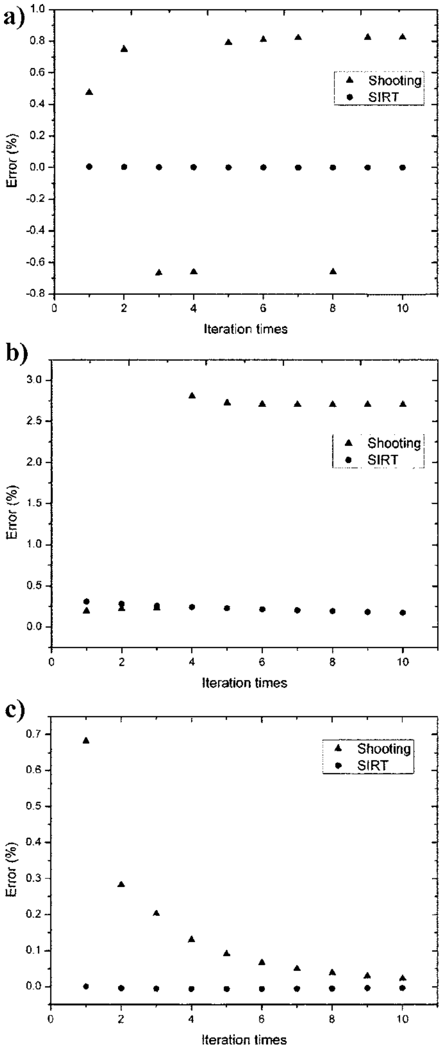

Figure 14. Traveltime relative errors at three receivers obtained by shooting and SIRT, respectively, as a function of the number of iterations: (a) receiver 5, (b) receiver 6, (c) receiver 7.

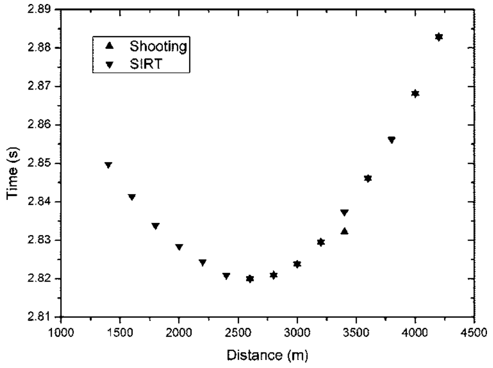

Figure 15. The traveltimes at 15 receivers obtained by shooting and SIRT, respectively. Upright triangles denote those by SIRT; inverted ones denote those by shooting. Seven receivers have no ray trace results from shooting.

Comparison with GOCAD ray-tracing scheme

Our model parameterization has similar generality to that of GOCAD in representing triangular interfaces. The main improvement is that our scheme involves smoothing normal vectors throughout interfaces.

Though GOCAD uses the discrete smooth interpolator (DSI) to modify exiting interfaces and interpolate some geologic nodes that are missing or inaccurate, it cannot deal with abrupt changes of normal vectors (Mallet, 1989, 1992). Therefore, ray tracing on top of the GOCAD modeler should be treated as an approximation. GuiZiou et al. (1996) use a ray perturbation technique and indicate that a raypath is bound to pass through the geologic nodes when the model is based on macrolayers. An initial trajectory is obtained by the shooting method; then a search of trajectories that can be constructed within the immediate neighborhood of the current trajectory is conducted for one with less associated traveltime. This ray-tracing procedure involves solving a nonlinear optimization problem and results in considerable errors. The accuracy can be improved by increasing the density of nodes on the surfaces, but the algorithmic computation is proportional to $(s + 1)^2$, where s is the number of the neighborhood and hence is much more time consuming. Furthermore, searching the immediate neighborhood of the current trajectory makes it converge easily to a local minimum, and the transmission and reflection points can be focused on one point or a local region, a situation that is fatal for seismic acquisition and survey design.

In our scheme, normal vectors are approximately C_2 continuous over an entire interface. Therefore, an initial ray can hit a model interface at an arbitrary point in SIRT. Furthermore, the modified midpoint can be located at an arbitrary point on the crossing interface. As a result, a trajectory traced in SIRT is more accurate than that in GuiZiou et al. (1996).

CONCLUSIONS

Complex media are described as aggregates of arbitrarily shaped blocks rather than layers and cells (grids). Using triangulated interfaces, complex media, such as faults, pinch-outs, intrusive tectonics, and lenses can be faithfully represented by blocks. We propose an algorithm for smoothing normal vectors over interfaces to enable more stable and accurate ray tracing.

Since the shooting method can be inefficient and, hence, time consuming, we develop a SIRT method based on Fermat's principle, which falls into the bending method category. Intersection points on an initial trajectory are modified segmentally instead of simultaneously. In a departure from traditional iterative methods, we update the midpoints by a first-order explicit modifying formula to speed up ray tracing greatly. In comparison tests with the shooting method, SIRT is faster and has better convergence in complex media.

Heterogeneous media can be approximated by a set of blocks, an approximation that is inefficient if a block has a constant gradient. SIRT could also be applied in a medium with constant gradients; the main difference is the modification of midpoints. Synthetic data examples show that the block modeling and SIRT method is suited to most complex media in three dimensions.

ACKNOWLEDGMENTS

We are indebted to Gengzhe Chang from the Department of Mathematics, University of Science and Technology of China, for helpful suggestions in the representation of interfaces. We are grateful to Johan Robertsson, Joachim Mispel, and three anonymous reviewers for critical comments and helpful suggestions that improved the quality of this work. We also thank Sidao Ni, Michael Schoenberger, and SWAT editors for improvements of language, grammar, and writing, and Yun Chen for modifying some of the figures. Key Foundation Project, National Natural Science Foundation of China (grant 40234044), National 973 Foundation of China (grant 2002CB412604), and National Natural Science Foundation of China (grants 40374010 and 40404009) jointly supported this work.

APPENDIX A

AREA COORDINATES

In the triangle $T_1T_2T_3$ (Figure A-1), if the coordinates of the three vertices T_1, T_2, and T_3 are given, then any point in the plane constructed with the three vertices has an area coordinate u_i $(i = 1,2,3)$. The area coordinate of point P in the triangle is described as

$$u_1 = \frac{[PT_2T_3]}{[T_1T_2T_3]}, \quad u_2 = \frac{[T_1PT_3]}{[T_1T_2T_3]}, \quad u_3 = \frac{[T_1T_2P]}{[T_1T_2T_3]}, \tag{A-1}$$

$[PT_2T_3]$ denotes the directional area of triangle PT_2T_3, which is the area of triangle PT_2T_3 when PT_2T_3 is counterclockwise; otherwise, $[PT_2T_3]$ is a negative value whose absolute value is equal to the area of the triangle.

Area coordinates have the following attributes:

- $u_1 + u_2 + u_3 \equiv 1$, when P is anywhere;
- $u_1 \geq 0, u_2 \geq 0, u_3 \geq 0, |u_1| + |u_2| + |u_3| = 1$, when P is inside the triangle or on the boundaries;
- $|u_1| + |u_2| + |u_3| > 1$, when P is outside the triangle.

APPENDIX B

MODIFYING FORMULA

The traveltime along raypath $P_1P_2P_3$ in Figure 6 can be computed by the equation

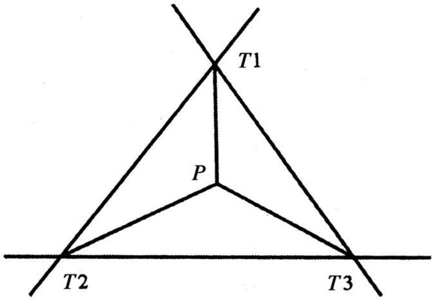

Figure A-1. Map of area coordinates.

$$T=\frac{l_1}{v_1}+\frac{l_2}{v_2}. \tag{B-1}$$

To satisfy the minimum traveltime principle, the partial derivative formulas at the new midpoint are

$$\left.\frac{\partial T}{\partial s}\right|_{(s=s^*,t=t^*)}=0, \qquad \left.\frac{\partial T}{\partial t}\right|_{(s=s^*,t=t^*)}=0. \tag{B-2}$$

Another form is

$$\begin{aligned}&\left(\sum_{i=1}^{3}\frac{x_i^*-x_i^{(1)}}{v_1l_1^*}+\sum_{i=1}^{3}\frac{x_i^*-x_i^{(3)}}{v_1l_2^*}\right)x_{is}^*=0,\\&\left(\sum_{i=1}^{3}\frac{x_i^*-x_i^{(1)}}{v_1l_1^*}+\sum_{i=1}^{3}\frac{x_i^*-x_i^{(3)}}{v_1l_2^*}\right)x_{it}^*=0,\end{aligned} \tag{B-3}$$

where

$$\begin{aligned}x_i^*&=x_i^{(2)}(s^*,t^*),\\x_{is}^*&=\frac{\partial x_i^{(2)}}{\partial s}(s^*,t^*),\\x_{it}^*&=\frac{\partial x_i^{(2)}}{\partial t}(s^*,t^*),\\l_1^*=l_1(s^*,t^*),&\quad l_2^*=l_2(s^*,t^*),\end{aligned} \tag{B-4}$$

After a Taylor series expansion, the variable formulas above are changed to the following with only first-order perturbation retained:

$$\begin{aligned}x_i^*&=x_i^{(2)}(s^*,t^*)=x_i^{(2)}(s+\Delta s,t+\Delta t)\approx x_i^{(2)}(s,t)+\frac{\partial x_i^{(2)}}{\partial s}\Delta s\\&\quad+\frac{\partial x_i^{(2)}}{\partial t}\Delta t=x_i^{(2)}+x_{is}^{(2)}\Delta s+x_{it}^{(2)}\Delta t,\end{aligned}$$

$$\begin{aligned}x_{is}^*&=x_{is}^{(2)}(s^*,t^*)=x_{is}^{(2)}(s+\Delta s,t+\Delta t)\approx x_{is}^{(2)}(s,t)+\frac{\partial x_{is}^{(2)}}{\partial s}\Delta s\\&\quad+\frac{\partial x_{is}^{(2)}}{\partial t}\Delta t=x_{is}^{(2)}+x_{iss}^{(2)}\Delta s+x_{ist}^{(2)}\Delta t,\end{aligned}$$

$$x_{it}^*=x_{it}^{(2)}(s^*,t^*)=x_{it}^{(2)}+x_{its}^{(2)}\Delta s+x_{itt}^{(2)}\Delta t,$$

$$\begin{aligned}l_1^*&=l_1(s^*,t^*)=l_1(s+\Delta s,t+\Delta t)=l_1(s,t)+\frac{\partial l_1}{\partial x_i}\left(\frac{\partial x_i^{(2)}}{\partial s}\Delta s\right.\\&\quad\left.+\frac{\partial x_i^{(2)}}{\partial t}\Delta t\right)=l_1+\frac{x_i^{(2)}-x_i^{(1)}}{l_1}(x_{is}^{(2)}\Delta s+x_{it}^{(2)}\Delta t),\end{aligned}$$

$$l_2^*=l_2(s^*,t^*)=l_2+\frac{x_i^{(2)}-x_i^{(3)}}{l_2}(x_{is}^{(2)}\Delta s+x_{it}^{(2)}\Delta t). \tag{B-5}$$

Elements in equation B-3 are substituted into equation B-5 to produce

$$\begin{aligned}&(x_{is}^{(2)}+x_{iss}^{(2)}\Delta s+x_{ist}^{(2)}\Delta t)[P_i+(x_{is}^{(2)}R-Q_{is})\Delta s\\&\quad+(x_{it}^{(2)}R-Q_{it})\Delta t=0,\\&(x_{it}^{(2)}+x_{its}^{(2)}\Delta s+x_{itt}^{(2)}\Delta t)[P_i+(x_{is}^{(2)}R-Q_{is})\Delta s\\&\quad+(x_{it}^{(2)}R-Q_{it})\Delta t=0.\end{aligned} \tag{B-6}$$

Expanding the equations and retaining only the first-order perturbation results in the ultimate modifying formulas

$$\begin{aligned}\Delta s&=\frac{U_{22}d_1-U_{12}d_2}{\Delta},\\\Delta t&=\frac{U_{11}d_2-U_{21}d_1}{\Delta},\\\Delta&=U_{11}U_{22}-U_{12}U_{21},\end{aligned} \tag{B-7}$$

where

$$\begin{aligned}U_{11}&=x_{is}^{(2)}(Q_{is}-x_{is}^{(2)}R)-x_{iss}^{(2)}P_i,\\U_{12}&=x_{is}^{(2)}(Q_{it}-x_{it}^{(2)}R)-x_{ist}^{(2)}P_i,\\U_{21}&=x_{it}^{(2)}(Q_{is}-x_{is}^{(2)}R)-x_{its}^{(2)}P_i,\\U_{22}&=x_{it}^{(2)}(Q_{it}-x_{it}^{(2)}R)-x_{itt}^{(2)}P_i,\\d_1&=x_{is}^{(2)}P_i,\\d_2&=x_{it}^{(2)}P_i,\\P_i&=\frac{x_i^{(2)}-x_i^{(1)}}{v_1l_1}+\frac{x_i^{(2)}-x_i^{(3)}}{v_2l_2},\\Q_{is}&=\frac{x_i^{(2)}-x_i^{(1)}}{v_1l_1}S_s^{(1)}+\frac{x_i^{(2)}-x_i^{(3)}}{v_2l_2}S_s^{(3)},\\Q_{it}&=\frac{x_i^{(2)}-x_i^{(1)}}{v_1l_1}S_t^{(1)}+\frac{x_i^{(2)}-x_i^{(3)}}{v_2l_2}S_t^{(3)},\\R&=\frac{1}{v_1l_1}+\frac{1}{v_2l_2},\\S_s^{(1)}&=\frac{x_i^{(2)}-x_i^{(1)}}{l_1^2}x_{is}^{(2)},\\S_t^{(1)}&=\frac{x_i^{(2)}-x_i^{(1)}}{l_1^2}x_{it}^{(2)},\end{aligned}$$

$$S_s^{(3)} = \frac{x_i^{(2)} - x_i^{(3)}}{l_2^2} x_{is}^{(2)},$$

$$S_t^{(3)} = \frac{x_i^{(2)} - x_i^{(3)}}{l_2^2} x_{it}^{(2)},$$

$$x_{is}^{(2)} = \frac{\partial x_i^{(2)}}{\partial s}, \quad x_{it} = \frac{\partial x_i^{(2)}}{\partial t},$$

$$x_{iss}^{(2)} = \frac{\partial^2 x_i^{(2)}}{\partial s^2}, \quad x_{ist}^{(2)} = x_{its}^{(2)} = \frac{\partial^2 x_i^{(2)}}{\partial s \partial t}, \quad x_{itt}^{(2)} = \frac{\partial^2 x_i^{(2)}}{\partial t^2}. \tag{B-8}$$

The location of a midpoint has a corresponding value of s and t in the located triangle. From the modifying formula, a modifying Δs and Δt can be derived, and $(s + \Delta s, t + \Delta t)$ can be regarded as the new position.

REFERENCES

Aki, K., and P. G. Richards, 1980, Quantitative seismology: Theory and methods: W. H. Freeman & Co.

Chang, G. Z., 1995, The mathematics of surfaces (in Chinese): Hunan Education Press.

Delaunay, B. N., 1934, Sur la sphere vide: Biological Bulletin of the Academy of Sciences, Classical Science and Mathematics, **7**, 793–800.

Fischer, R., and J. Lees, 1993, Shortest path ray tracing with sparse graphs: Geophysics, **58**, 987–996.

Gao, E. G., G. M. Xu, and Y. Zhao, 1998, Segmentally iterative ray tracing method for any interface: Oil Geophysical Prospecting (in Chinese), **33**, no. 1, 54–60.

Gjøystdal, H., J. E. Reinhardsen, and K. Åstebol, 1985, Computer representation of complex 3-D geological structures using a new "solid modeling" technique: Geophysical Prospecting, **33**, 195–1211.

GuiZiou, J. L., J. L. Mallet, R. Madariaga, 1996, 3D seismic reflection tomography on top of the GOCAD depth modeler: Geophysics, **61**, 1499–1510.

Julian, B. R., and D. Gubbins, 1977, Three-dimensional seismic ray tracing: Journal of Geophysics, **43**, 95–113.

Langan, R. T., I. Lerche, and R. T. Cutler, 1985, Tracing of rays through heterogeneous media, An accurate and efficient procedure: Geophysics, **50**, 1456–1465.

Mallet, J. L., 1989, Discrete smooth interpolation: ACM Transactions on Graphics, **8**, 121–144.

——, 1992, Discrete smooth interpolation in geometric modeling: Computer-Aided Design, **24**, 178–193.

Mao, W. J., and G. W. Stuart, 1997, Rapid multiwave-type ray tracing in complex 2D and 3D isotropic media: Geophysics, **62**, 298–308.

Moser, T. J., 1991, Shortest path calculation of seismic rays: Geophysics, **56**, 59–67.

Pereyra, V., 1992, Two-point ray tracing in general 3-D media: Geophysical Prospecting, **40**, 267–287.

——, 1996, Modeling, ray tracing, and block nonlinear travel-time inversion in 3-D: Pure and Applied Geophysics, **148**, 345–386.

Pereyra, V., W. H. K. Lee, and H. B. Keller, 1980, Solving two-point seismic ray tracing problems in heterogeneous medium — Part 1: A general adaptive finite difference method: Bulletin of the Seismological Society of America, **70**, 79–99.

Prothero, W., W. Taylor, and J. 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.

Rawlinson, N., G. A. Houseman, and C. D. N. Collins, 2001, Inversion of seismic refraction and wide-angle reflection traveltimes for three-dimensional layered crustal structure: Geophysical Journal International, **145**, 381–400.

Sambridge, M., J. Braun, and H. McQueen, 1995, Geophysical parameterization and interpolation of irregular data using natural neighbours, Geophysical Journal International, **122**, 837–857.

Sun, Y., 1993, Ray tracing in 3-D media by parametrized shooting: Geophysical Journal International, **114**, 145–155.

Thurber, C., and W. Ellsworth, 1980, Rapid solution of ray tracing problems in heterogeneous media: Bulletin of the Seismological Society of America, **70**, 1137–1148.

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.

Velis, D. R., and T. J. 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.

Vinje, V., K. Åstebol, E. Iversen, and H. Gjøystdal, 1999, 3D ray modeling by wavefront construction in open models: Geophysics, **64**, 1912–1919.

Vinje, V., E. Iverson, K. Åstebol, and H. Gjøystdal, 1996, Estimation of multivalued arrivals in 3D models using wavefront construction — Part 1: Geophysical Prospecting, **44**, 819–842.

Vinje, V., E. Iverson, and H. Gjøystdal, 1993, Traveltime and amplitude estimation using wavefront construction: Geophysics, **58**, 1157–1166.

Virieux, J., and V. Farra, 1991, Ray tracing in 3D complex isotropic media: An analysis of the problem: Geophysics, **56**, 2057–2069.

Wang, D. R., and Z. H. Yang, 1990, Introduction to numerical approximation theory (in Chinese): Higher Education Press.

Xu, G. M., S. Wei, E. G. Gao, Q. Z. Lin, X. Y. Jiang, and K. Y. Luo, 2001, Block model-building and ray-tracing in 2-D complicated medium: Oil Geophysical Prospecting (in Chinese), **36**, no. 2, 213–219.

Zelt, C. A., and R. B. Smith, 1992, Seimic traveltime inversion for 2-D crustal velocity structure: Geophysical Journal International, **108**, 16–34.

Zhang, Z., G. Wang, J. Teng, and S. Klemperer, 2000, CDP mapping to obtain the fine structure of the crust and upper mantle from seismic sounding data: An example for the southeastern China: Physical Earth Planetary International, **122**, 133–146.

Zhao, A., Z. Zhang, and G. Teng, 2004, Minimum travel time tree algorithm for seismic ray tracing: improvement in efficiency: Journal of Geophysics and Engineering, **1**, no. 4, 245–251.

Zhao, D., A. Hasegawa, and S. Horiuchi, 1992, Tomographic imaging of P and S wave velocity structure beneath northeastern Japan: Journal of Geophysical Research, **97**, 19909–19928.