Geophys. J. Int. (2001) **145,** 447–459

Simulated annealing ray tracing in complex three-dimensional media

Danilo R. Velis[1] and Tadeusz J. Ulrych[2]

[1] *Facultad de Ciencias Astronómicas y Geofísicas, Universidad Nacional de La Plata, Paseo del Bosque s/n, La Plata* 1900, *Argentina.*
E-mail: velis@fcaglp.unlp.edu.ar
[2] *Department of Earth and Ocean Sciences, University of British Columbia,* 2219 *Main Mall, Vancouver, BC,* V6T 1Z4, *Canada.*
E-mail: ulrych@geop.ubc.ca

Accepted 2000 November 30. Received 2000 November 17; in original form 1999 February 2

SUMMARY

Simulated annealing has been applied to seismic ray tracing to determine the minimum traveltime ray path connecting two points in complex 3-D media. In contrast to conventional ray tracing schemes such as shooting and bending, simulated annealing ray tracing (SART) overcomes some well-known difficulties regarding multipathing and take-off angle selection. These include local convergence (that is, failing to obtain the ray path with absolute minimum traveltime) and divergence of the take-off angle selection strategy. Under these circumstances, shooting and bending methods may not provide reliable results in highly variable 3-D media. A flexible model representation is used to accommodate a large class of velocity models.

Key words: block model, heterogeneous media, numerical techniques, ray tracing, simulated annealing.

1 INTRODUCTION

Ray tracing plays a key role in seismological studies. Significant attention has been devoted to the initial value problem, in which the ray is specified by an initial point and take-off angles. This is in general a well-resolved problem (Červený 1987). However, tomographic studies and earthquake location usually require precise traveltime and trajectory computations of seismic waves propagating between two points (boundary value problem), not a trivial task in 3-D laterally heterogeneous media. A range of methods for solving the two-point ray tracing problem in 3-D media have been developed in the literature. Traditional methods include shooting and bending. Other methods include wave-front techniques and methods based on graph theory.

In the shooting method, a source point is chosen and a fan of rays is propagated by specifying a set of take-off angles. After selecting those angles that generate rays that arrive close to the receiver, a search strategy is applied to update the angles. This procedure terminates when a ray is found that ends within a given distance of the receiver. Frequently the receiver location is an ill-behaved function of the take-off angles, thus the strategy for updating the take-off angles may become a difficult task, and divergence is a common issue unless the model or the results of the computations are conveniently smoothed (Langan *et al.* 1985). As a consequence, some ray paths can be missed. The problem is especially severe in complicated 3-D models (Langan *et al.* 1985; Sambridge & Kennett 1990; Virieux & Farra 1991; Sun 1993).

In the bending method both endpoints are linked by an initial guess path (usually a straight segment), which is then perturbed iteratively so as to satisfy the ray equations or Fermat's principle of stationary time. Unlike standard shooting, bending always produces a solution that is assumed to be a ray connecting source and receiver. In general, bending involves the solution of a highly non-linear optimization problem, which requires gradients of the cost function to update the ray path. In complicated velocity structures, bending can potentially overlook multipathing propagation because the solution depends on the initial guess (Julian & Gubbins 1977; Thurber & Ellsworth 1980; Um & Thurber 1987; Prothero *et al.* 1988; Pereyra 1992; Mao & Stuart 1997).

Wave-front techniques are very efficient and effective for solving the two-point ray tracing problem, especially in smooth models with a large number of receivers. Here rays are shot simultaneously so as to form a wave front. As rays diverge, new rays are interpolated in such a way that the wave front is always regularly sampled (Vinje *et al.* 1993, 1996). A high density of rays is required when the model exhibits strong heterogeneities (including interfaces), decreasing both the computational efficiency and the effectiveness of the interpolation schemes for detecting rapid velocity variations.

Methods based on graph theory are designed to locate the shortest time path through a pre-defined discrete set of grid nodes, which can in turn be used as an initial guess for a more accurate bending procedure (Moser 1989, 1991; Fisher & Lees 1993). However, it may not be clear whether the final ray path is a global or a local minimum (Moser *et al.* 1992). Graph theory ray tracing has recently been applied in simple 3-D media (Cheng & House 1996), but its implementation in complex 3-D structures is limited not only because of memory limitations, but also because the computation time increases dramatically when the node spacing is reduced for accuracy purposes.

Thus, non-linear methods that can find the global minimum are important. Simulated annealing (SA) is one such technique. Velis & Ulrych (1996) applied SA to ray tracing for the first time overcoming the previously described difficulties in 2-D models. Specifically, their method reduces significantly the convergence problems associated with the initial angle selection strategy. At the same time, the solution is independent of the initial guess. Simulated annealing ray tracing (SART) is an iterative shooting procedure that attempts to find the optimum take-off angles corresponding to the ray path with minimum traveltime. At each iteration, the ray is propagated from the source by solving a standard initial value problem (IVP). The last portion of the ray path is then forced to pass through the receiver. In this paper, SART is extended to 3-D models and further improvements concerning accuracy and efficiency are incorporated into the original scheme. Essentially, the two-point ray tracing problem is put into a non-linear optimization framework which is then solved by means of very fast simulated annealing (VFSA) (Ingber 1989, 1993).

At this point it is important to note that even though all of the aforementioned techniques have been devised to solve the two-point ray tracing problem, usually their ultimate goals are different. Some methods are intended to obtain a single solution, which may depend on the initial guess (e.g. bending, shooting) or may not depend on the initial guess (e.g. graph theory). Other methods are intended to obtain all the solutions (e.g. wave-front construction). SART's goal is to obtain a single solution independent of the initial guess. This solution corresponds to the absolute minimum traveltime ray path that can be obtained by ray-theory-based methods such as shooting.

For testing SART performance, we develop a versatile model parametrization scheme that is capable of accommodating a large class of velocity structures. These include any number of regions with constant and non-constant velocities separated by planar and curved interfaces. As will be shown in the numerical examples, because of the extreme complexity of the traveltime curves, some models will always pose a challenge for any two-point ray tracing system. We also conduct a number of numerical experiments to illustrate the behaviour of SART in comparison with standard ray tracing techniques (in terms of computational effort, effectiveness and accuracy) and to assess its viability for solving the two-point ray tracing problem in complex 3-D media.

2 BACKGROUND

2.1 Earth model

The velocity model, which is contained within the cube $(x_{\min}, x_{\max}) \times (y_{\min}, y_{\max}) \times (z_{\min}, z_{\max})$, is composed of any number of regions separated by curved interfaces representing geological horizons, fault planes, etc. We assume that all interfaces are explicit functions of the form $z = g(x, y)$. Vertical interfaces can be approximated by surfaces with very large slopes. For example, the vertical plane $x = c$ can be approximated by the plane $z = (x - c)d$, with $|d| \gg |z|$, since $x = c + z/d \simeq c$. The velocity within each region may be specified by any function $v = v(\mathbf{x})$, $\mathbf{x} = (x, y, z)$, and must be twice differentiable. Interfaces may or may not intersect each other, allowing for a greater flexibility in constructing a wide variety of models, including complex 3-D structures. Fig. 1 shows a typical example, where we construct two blocky models: a fault model and a salt-dome model. Some of the interfaces are not defined for all values of x and y, thus a fault plane or a salt intrusion can be easily simulated. Table 1 shows the velocity and interface definitions for each of the models depicted in Fig. 1. More elaborate interface descriptions in terms of triangular networks (Mallet *et al.* 1989; Vinje *et al.* 1993, 1996), parametric surface patches (Pereyra 1992), implicit B-splines (Virieux & Farra 1991) or natural neighbour interpolation (Sambridge *et al.* 1995) can be devised, but explicit surfaces are preferred because of their simplicity and because it is easier and more cost-effective to compute the required intersection points between surfaces and ray paths.

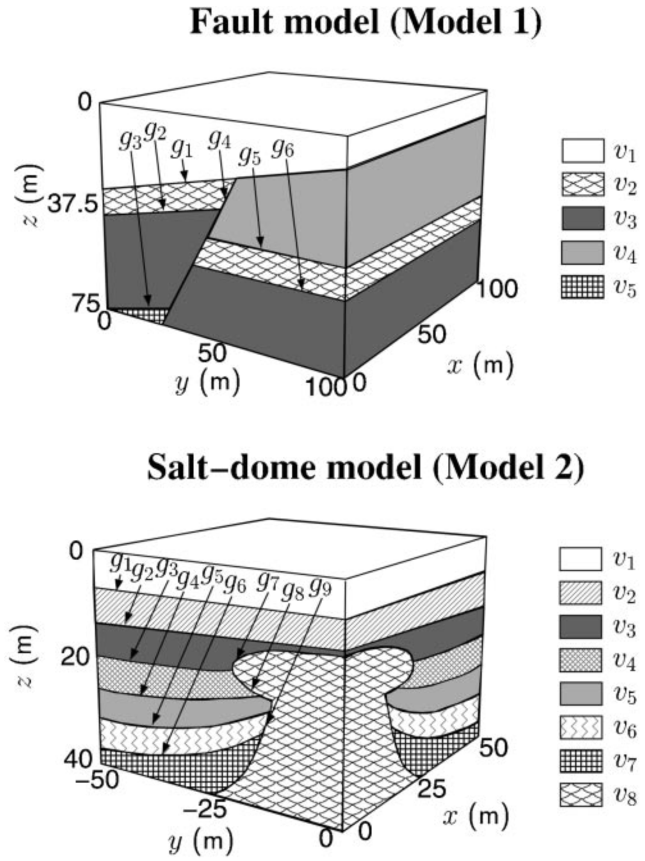


Figure 1. Velocity models used in the analysis of SART. Note that Model 2 is shown partially. Interfaces g_i and velocities v_i are summarized in Table 1.

2.2 Initial value problem

We write the ray equations (Elissevnin 1965; Červený 1987) in the form

$$\begin{cases} \partial_t x = v \sin\theta \cos\xi \\ \partial_t y = v \sin\theta \sin\xi \\ \partial_t z = v \cos\theta \\ \partial_t \theta = -\cos\theta \left(\dfrac{\partial v}{\partial x} \cos\xi + \dfrac{\partial v}{\partial y} \sin\xi \right) + \dfrac{\partial v}{\partial z} \sin\theta \\ \partial_t \xi = \dfrac{1}{\sin\theta} \left(\dfrac{\partial v}{\partial x} \sin\xi - \dfrac{\partial v}{\partial y} \cos\xi \right) \end{cases}, \qquad (1)$$

Table 1. Velocities and interfaces defining models in Fig. 1. Here $u=(1+r^2)^{-1}$ and $u'=(0.01+r^2)^{-1}$, where $r^2=(x^2+y^2)/400$.

	Fault model		Salt-dome model	
No.	$v(\mathbf{x})$ (km s^{-1})	$z=g(x, y)$ (m)	$v(\mathbf{x})$ (km s^{-1})	$z=g(x, y)$ (m)
1	1.5	$30-0.2y$	$2.0+0.007z$	$20/3-0.5u$
2	2.5	$40-0.2y$	$2.2+0.012z$	$40/3-2.5u$
3	3.6	$75-0.2y$	$2.25-\frac{y(1+0.2y)}{500}$	$60/3-5.0u$
4	3.0	$115-1.6y$	2.5	$80/3-10u$
5	5.0	40	3.1	$100/3-15u$
6	–	50	$4.0+\frac{z-25}{10}$	$120/3-20u$
7	–	–	4.8	$12.5+5r^4$
8	–	–	5.5	$22.5-5r^4$
9	–	–	–	$140/3-10u'$

where θ and ξ stand for declination and azimuth angles, which describe the direction of the ray at every point of its trajectory (Fig. 2), and $v=v(\mathbf{x})$ is the wave speed. By solving the system of differential equations (1), it is possible to describe the ray trajectory at every time t, the independent variable of integration, given the appropriate initial conditions. We solve eqs (1) using Euler and Runge–Kutta methods (e.g. Press *et al.* 1992) using the initial conditions ($t=0$)

$$\begin{cases} \mathbf{x}(0) = \mathbf{x}_s \\ \theta(0) = \theta_s \\ \xi(0) = \xi_s \end{cases}, \tag{2}$$

where subscript s stands for source.

Solving the two-point ray tracing problem in complex 3-D models is a very difficult task, and solving the IVP is not a trivial one. Only the main steps of the ray tracing system will be described here. The ray propagates until it finds an interface. At this stage it is necessary to obtain the intersection point between the ray path and the interface, and to apply Snell's law. The propagation then resumes with the new initial conditions until the ray endpoint arrives at a model boundary, where propagation stops. Also, the propagation stops when the ray arrives at a pre-defined target surface (usually a plane) passing through the receiver (for example, a vertical plane in a borehole experiment).

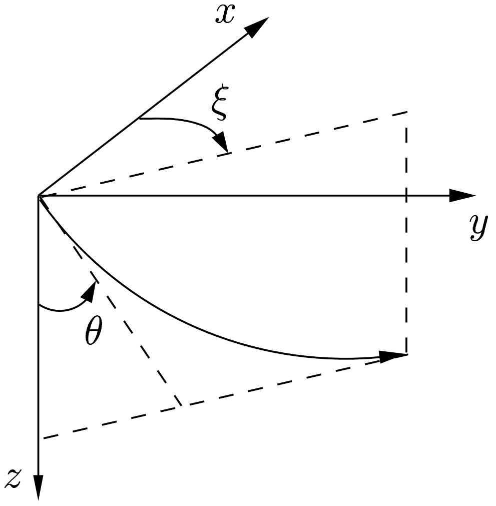


Figure 2. The ray direction is described by declination θ and azimuth ξ.

2.3 Intersection point

Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be the ray path coordinates at times t_1 and $t_2=t_1+\Delta t$ respectively, which are on opposite sides of the interface, described by $z=g(x, y)$. The intersection point, $\mathbf{x}'_2$, is then obtained by iteratively adjusting the time step until the ray coordinates coincide with the interface within a given tolerance. The problem thus reduces to solving for τ such that

$$z'_2(t_1+\tau) - g[x'_2(t_1+\tau), y'_2(t_1+\tau)] \simeq 0\,. \tag{3}$$

It is clear that the root of eq. (3) is bracketed in the interval [0, Δt]. The selection of an appropriate root-finder is very important for obtaining an accurate solution in a few iterations. We select Brent's method (see e.g. Press *et al.* 1992) because it provides quadratic convergence, requires no derivatives and is as robust as the bisection method. In practice we have found that one to three adjustments are enough to find the optimum τ with an absolute tolerance of 1×10^{-5} ms. Once the intersection point has been found, the ray path coordinates and traveltime are updated accordingly before applying the reflection/transmission laws and resetting the initial conditions (i.e. new take-off angles). If more than one interface is crossed at a given time step, the one with the smallest adjusted time step is selected. The same kind of iterative adjustment is performed when the ray crosses one of the model boundaries. In this particular case, the propagation is terminated.

2.4 Ray signature

During the propagation, an index indicating the current region the ray is traversing is saved and updated after each interface crossing. This index is used to determine which velocity function must be used in the integration of eqs (1). A flag for each interface is also provided that indicates the decision necessary in the case where the ray arrives at the interface. This allows one to model *P* and *S* waves, or a conversion between the two, or to force a reflection at any given interface to simulate a reflector. This is very important from the point of view of phase identification. The user must select *a priori* which type of ray is going to be traced.

3 BOUNDARY VALUE PROBLEM

3.1 Problem definition

In general, it is the goal of a ray tracing boundary value problem (BVP) solver to find the ray path connecting any two fixed points within the model boundaries (two-point BVP). The ray path must be consistent with the ray theory used to propagate the wave through the given velocity field. Usually this is carried out by finding the unknown take-off angles that give rise to a ray propagating from the source to the receiver. Although the solution is sometimes non-unique (several ray paths satisfying the ray equations may connect both endpoints), the purpose of SART is to pick the ray path that exhibits the absolute minimum traveltime. Here, an appropriate cost function is defined in such a way that its global minimum corresponds to the desired solution. We subsequently use VFSA (Ingber 1989, 1993) to solve the optimization problem. At this point it is worthwhile making a distinction among various BVP alternatives. In the standard two-point ray tracing problem, the wave travels from source to receiver without any additional constraint along its trajectory. We refer to this BVP as tracing direct waves. Other BVP variants introduce additional constraints along the ray trajectory. Such is the case of tracing reflections (including normal-incidence rays), head waves or point diffractions.

3.2 SART strategy

The two-point ray tracing problem used in SART is based on the straight-ray construction (Velis & Ulrych 1996; Velis 1998). Both source, $\mathbf{x}_s$, and receiver, $\mathbf{x}_r$, are fixed and the optimum take-off angles θ_s and ξ_s are to be found so that the total traveltime is a global minimum. The total traveltime (cost function) is written as

$$\Phi(\theta_s, \xi_s) = T_{se} + \rho T_{er} \,, \tag{4}$$

where T_{se} is the traveltime that is obtained after solving the IVP from the source to the point where the ray exits the model boundaries, $\mathbf{x}_e$ (emerging point), T_{er} is the traveltime associated with the straight-ray construction (refer to Fig. 3a) and ρ, $\rho \geq 1$, is a trade-off parameter. This construction is simply a straight segment connecting the emerging point with the receiver, and is used solely for obtaining a ray arriving at the receiver at any given iteration. The arbitrary construction just described in general leads to non-physical ray paths. However, when Φ is a minimum, Fermat's principle is satisfied and point $\mathbf{x}_e$ coincides with the receiver (the straight segment collapses to a point). Since two angles are required to determine uniquely the whole ray trajectory, Φ is a 2-D function and often multimodal and non-differentiable. VFSA is finally used to find the global minimum of Φ efficiently. The trade-off parameter ρ is used as a weighting factor that helps to reduce T_{er} to zero during the minimization process. In general, we have found that $\rho = 1$ is enough to produce the correct solution. In some difficult and unusual cases, however, and assuming the receiver does not lie in a shadow zone, $\rho > 1$ can be selected to avoid solutions with $T_{er} > 0$ (Velis 1998).

The above procedure can be extended to deal with reflected waves (Fig. 3b). Now we write the cost function

$$\Phi(\theta_s, \xi_s) = \begin{cases} T_{su} + T_{ue} + \rho T_{er} & \text{reflection = true} \\ T_{max} & \text{otherwise} \end{cases} \,, \tag{5}$$

where point $\mathbf{x}_u$ is the point where the ray intersects the reflector, $z = g(x, y)$, and T_{max} is the maximum guessed value $T_{su} + T_{ue} + \rho T_{er}$ may take for all possible take-off angles (in practice, a rough estimate only of T_{max} is needed). The ray is propagated from $\mathbf{x}_s$ to $\mathbf{x}_u$, where Snell's law of reflection is applied. The propagation continues to $\mathbf{x}_e$, and finally to $\mathbf{x}_r$ using a straight line. When the total traveltime Φ is a minimum, $\mathbf{x}_e \equiv \mathbf{x}_r$.

Normal-incidence rays are a particular case of reflections where source and receiver are coincident. These rays are used to build zero-offset sections. The problem can be cast as finding the optimum coordinates, (x_u, y_u), of the ray that travels towards the source–receiver point with an initial direction that is parallel to the normal of the reflector at that point. Consequently, the cost function becomes

$$\Phi(x_u, y_u) = 2(T_{ue} + \rho T_{es}) \,, \tag{6}$$

again a 2-D function.

For tracing head waves along a plane refractor, the number of unknowns is four: the coordinates of the points the ray enters and leaves the refractor, x_u, y_u, x_v and y_v [note that $z_u = g(x_u, y_u)$ and $z_v = g(x_v, y_v)$]. In fact, the ray is propagated starting at these two locations towards the source and receiver, respectively. The take-off angles, which are known, are easily computed in terms of the critical angles of refraction (Velis 1998). As a result the final ray path is composed of five portions

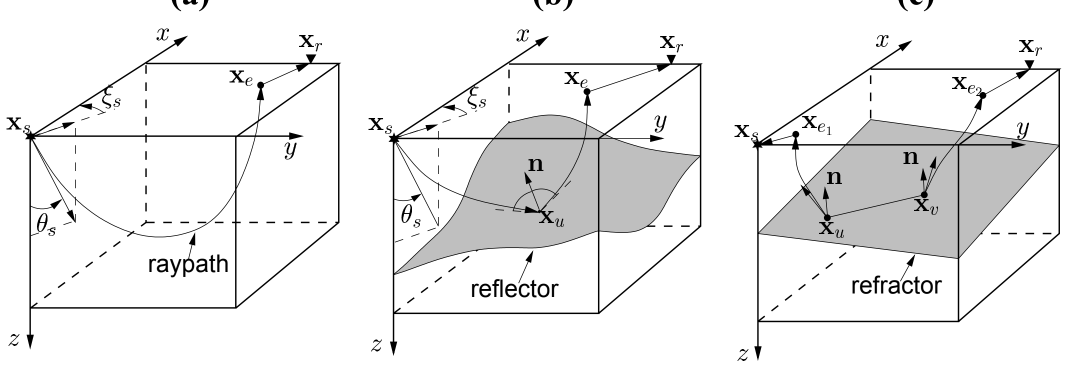


Figure 3. Ray tracing (a) direct waves, (b) reflections and (c) head waves using SART.

(see Fig. 3c), and the cost function is

$$\Phi(x_\mathrm{u}, y_\mathrm{u}; x_\mathrm{v}, y_\mathrm{v}) = \rho T_{\mathrm{se}_1} + T_{\mathrm{e}_1\mathrm{u}} + T_{\mathrm{uv}} + T_{\mathrm{ve}_2} + \rho T_{\mathrm{e}_2\mathrm{r}} \,. \tag{7}$$

In this equation, the first and last contributions to the sum correspond to the straight-ray constructions, which at convergence go to zero. Term T_uv is the traveltime along the refractor from $\mathbf{x}_\mathrm{u}$ to $\mathbf{x}_\mathrm{v}$. The remaining two terms are obtained by solving the IVP starting at points $\mathbf{x}_\mathrm{u}$ and $\mathbf{x}_\mathrm{v}$.

3.3 Simulated annealing optimization

A Monte Carlo sampling technique for modelling the evolution of a solid at a given temperature was proposed by Metropolis *et al.* (1953). Kirkpatrick *et al.* (1983) generalized the concept and applied it to non-linear optimization problems. Here the unknown parameters (the model) play the role of the particles in the solid, and the cost function represents the energy of the system. In SA optimization, parameter values are drawn from a distribution that depends on a control temperature, which is gradually decreased. At high temperatures, the model space is sampled more or less uniformly. However, at low temperatures, models with the lowest energy are preferentially sampled. The model is updated using the Metropolis criterion (Metropolis *et al.* 1953) by evaluating whether to accept or reject the proposed model according to the change in the cost function value. Finally, convergence is achieved at low temperatures when no further improvement in the cost function is observed. The key is that when the cost function decreases, the proposed model is accepted unconditionally. On the other hand, when the cost function increases, it is accepted with a probability greater than zero. This strategy allows SA to escape local minima.

The advantage of VFSA over traditional SA techniques relies on the choice of the distribution generator and the cooling schedule. For selecting new parameter values, VFSA uses a long-tailed Cauchy-like distribution that permits the exploration of the model space more effectively than using Gaussian or uniform distributions (Ingber 1993; Sen & Stoffa 1995), as in various SA algorithms. Also, a faster cooling rate is allowed to accelerate convergence without limiting its ability to avoid local minima. At iteration k, the temperature associated with each parameter is given by

$$T_k = T_0\, \mathrm{e}^{-ck^{1/M}} \,, \tag{8}$$

where T_0 is the initial temperature, M is the number of unknowns and c is a user-defined constant that can be used to tune the algorithm. The cost temperature is defined similarly to the parameter temperature. A further feature of VFSA is that parameter temperatures can be adapted dynamically ('re-annealing') according to the sensitivity of the cost function to each dimension in the model space, thus allowing for a different annealing schedule for each parameter. The reader is referred to the work by (Ingber 1989, 1993) for a detailed description of VFSA.

3.4 Refinement of the solution

Despite the fact that VFSA (the method used in SART) converges significantly faster than conventional SA methods, when Φ is close to the global minimum, reducing T_er to zero may take several iterations. At these low-temperature stages, SART switches to a local optimization algorithm to find the optimum θ_s and ξ_s such that

$$d^2_\mathrm{er}(\theta_\mathrm{s}, \xi_\mathrm{s}) = \sum_{i=1}^{3} (x_{\mathrm{r}_i} - x_{\mathrm{e}_i})^2 \le \varepsilon^2 \,, \tag{9}$$

where x_{r_i} and x_{e_i} are the coordinates of the receiver and the emerging points, $\mathbf{x}_\mathrm{r}$ and $\mathbf{x}_\mathrm{e}$, respectively, and ε is a small tolerance distance. Since eq. (9) is non-linear, the minimization is performed iteratively. Here the best SA solution obtained so far is used as the initial guess for the linearizing stage. The switch is made after the maximum number of SA iterations, k_max, has been reached. This hybrid strategy allows SART to compute accurately the global minimum traveltime in an efficient manner. Since the local optimization algorithm is applied only after SA has converged close enough to the global minimum, problems regarding instability and divergence associated with these methods are of no concern.

Any linearizing method can be used to solve eq. (9). At the beginning of the jth iteration, the current take-off angle estimates are $(\theta^j_\mathrm{s}, \xi^j_\mathrm{s})$. Essentially, the jth iteration then consists of the computation of a search vector $(\Delta\theta^j_\mathrm{s}, \Delta\xi^j_\mathrm{s})$ from which one obtains the new estimate $(\theta^{j+1}_\mathrm{s}, \xi^{j+1}_\mathrm{s})$ according to $\theta^{j+1}_\mathrm{s} = \theta^j_\mathrm{s} + \alpha^j \Delta\theta^j_\mathrm{s}$ and $\xi^{j+1}_\mathrm{s} = \xi^j_\mathrm{s} + \alpha^j \Delta\xi^j_\mathrm{s}$, where the step size α^j is obtained by linear search or other strategy. The selection of the search vector is largely what distinguishes one method from another. In the method of steepest descent (the approach used in this work) the greatest reduction in the cost function value is obtained in the direction of the negative gradient. Thus, dropping the superscript j for the sake of clarity,

$$(\Delta\theta_\mathrm{s}, \Delta\xi_\mathrm{s}) = -\nabla d^2_\mathrm{er} = \left(2 \sum_{i=1}^{3} \delta_i \frac{\partial x_{\mathrm{e}_i}}{\partial \theta_\mathrm{s}},\, 2 \sum_{i=1}^{3} \delta_i \frac{\partial x_{\mathrm{e}_i}}{\partial \xi_\mathrm{s}} \right), \tag{10}$$

where $\delta_i = x_{\mathrm{r}_i} - x_{\mathrm{e}_i}$ and the derivatives are estimated using finite differences. To obtain the step size α we write the Taylor expansion to first-order approximation,

$$x_{\mathrm{e}_i}(\theta_\mathrm{s} + \Delta\theta_\mathrm{s}, \xi_\mathrm{s} + \Delta\xi_\mathrm{s}) \simeq x_{\mathrm{e}_i}(\theta_\mathrm{s}, \xi_\mathrm{s}) + \alpha a_i \,, \tag{11}$$

where $a_i = \Delta\theta_\mathrm{s}(\partial x_{\mathrm{e}_i}/\partial\theta_\mathrm{s}) + \Delta\xi_\mathrm{s}(\partial x_{\mathrm{e}_i}/\partial\xi_\mathrm{s})$, $i = 1, 2, 3$. Finally, substituting eq. (11) into eq. (9), differentiating with respect to α and setting $\partial d^2_\mathrm{er}/\partial\alpha = 0$ yields

$$\alpha = \frac{\sum_i (x_{\mathrm{r}_i} - x_{\mathrm{e}_i}) a_i}{\sum_i a_i^2} \,. \tag{12}$$

Locally, d_er is a well-behaved function, and the convergence to the global minimum is guaranteed. In practice, 2–8 iterations are enough to reduce d_er virtually to zero within machine precision.

4 SART PERFORMANCE: ANALYSIS AND DISCUSSION

4.1 Numerical examples

In this section we test SART using two models representing 3-D geological complex structures (Fig. 1). Model 1 represents a fractured layered subsurface with planar interfaces and constant velocities within each region that we call the 'fault model'. Model 2 is called the 'salt model' and represents a salt dome that bursts into a layered medium. Here interfaces are curved and velocities are not constant within regions. Although Model 1 is much simpler than Model 2 in terms of ray path and traveltime behaviour, both models are intended to illustrate the difficulties that may arise in solving the two-point ray tracing problem in complex 3-D media.

For simplicity, all receivers are located on a model boundary, so that the target interface coincides with some of the planes defining the model boundaries. In all cases, the SA iteration stops after a maximum number of iterations, $k_{\max}$, or when no further improvement in the cost function is observed after several iterations. We set $c=1.0$ for the cooling schedule (eq. 8) and we choose the initial temperature so that almost all proposed take-off angles are accepted initially. This corresponds to the thermal analogue being completely melted. For this purpose it is enough to set this value numerically to be equal to the average cost function for a set of arbitrary points selected at random from the model space. All computations are performed on a PC Pentium III 500 MHz.

4.1.1 Fault model

Model 1 is comprised of seven regions with constant velocities delimited by planar interfaces and a fault plane, as shown in Fig. 1. Table 1 summarizes the interfaces and velocities that were used to generate this model. For simplicity, we first take into account a 2-D slice of the model: the plane $x=50$ m. We placed a source at (50, 0, 70) m and produced a fan of rays with take-off angles θ_s varying from 60° to 140° (ξ_s is fixed at 90° so that all ray trajectories lie on the plane $x=50$ m). Fig. 4(a) shows the resulting ray paths. The presence of shadow zones and multiple arrivals to any receiver located either on the surface or on the right-hand model boundary are clear from the figure. Due to incidence angles beyond the critical angle, total reflected waves are also generated at some discontinuities.

For a receiver at (50, 100, 16.5) m, for example, there are four solutions to the BVP. Their trajectories are shown in Fig. 4(b) and their traveltimes are summarized in Table 2. Cost function Φ as a function of θ_s and ξ_s is illustrated in Fig. 5. What makes it difficult to minimize this function globally is not only the presence of local minima, but also the great number of discontinuities generated by the model. SA appears to be a natural tool for solving this kind of non-linear optimization problem. The global minimum, obtained in about 250 iterations, is $\theta_s=125.90°$, $\xi_s=90.00°$, $\Phi_{\rm opt}=34.714$ ms.

Table 2. Optimum θ_s and $\Phi(\theta_s, \xi_s)$ in the fault and salt-dome models. Optimum azimuth $\xi_s=90.00$ in all cases. Numbers in italics indicate global minimum (SART solution).

	Fault model		Salt-dome model	
No.	θ_s (°)	Φ (ms)	θ_s (°)	Φ (ms)
1	70.16	35.986	43.11	43.558
2	87.44	36.410	43.63	43.580
3	108.21	35.915	46.44	43.596
4	*125.90*	*34.714*	*54.55*	*39.733*
5	–	–	58.97	47.011
6	–	–	59.62	46.653
7	–	–	78.19	49.782
8	–	–	80.27	49.748

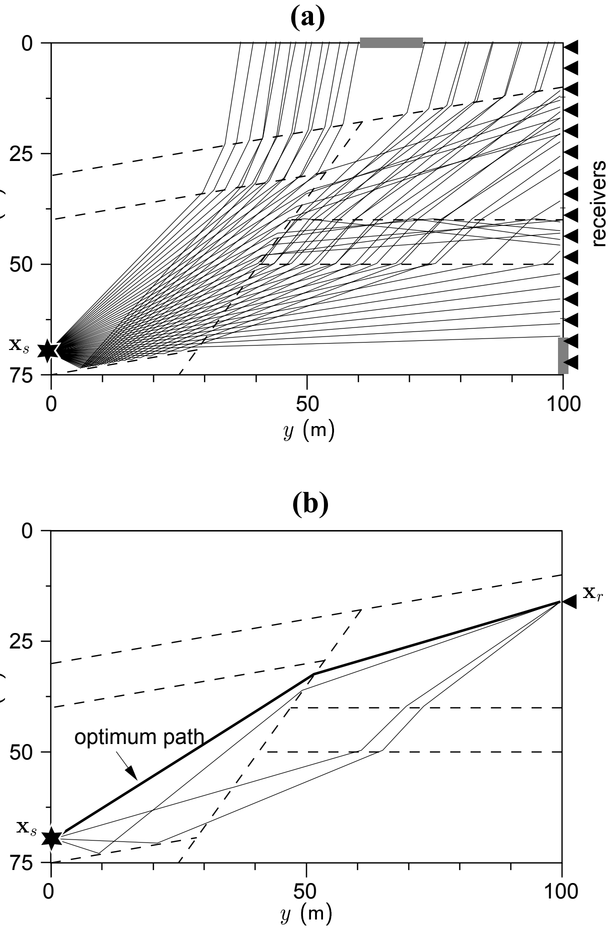


Figure 4. Model 1. (a) Fan of rays with equally spaced take-off angles in the range (60°, 140°). Short grey segments indicate shadow zones. (b) The four plotted ray paths satisfy the ray equations but exhibit different traveltimes. Dashed lines show model interfaces.

Finally, we used SART to find each global minimum for a set of 75 receivers uniformly distributed at locations $\mathbf{x}_{r_i}=(50, 100, i\times h+0.5)$ m, $i=0, 1, \ldots, 74$, where $h=1$ m is the geophone vertical spacing. In all cases the source was fixed

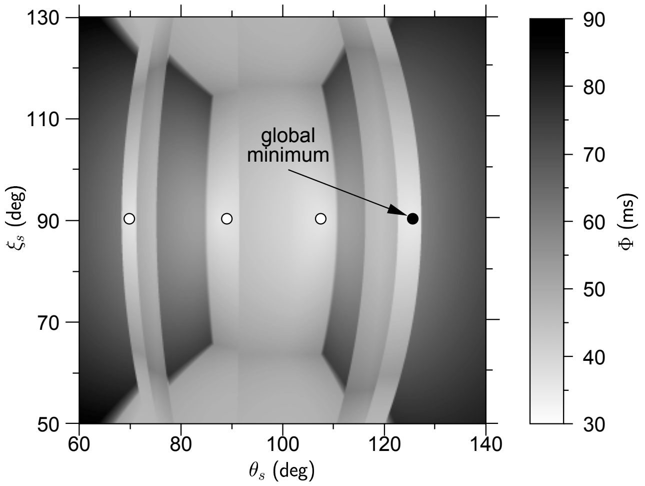
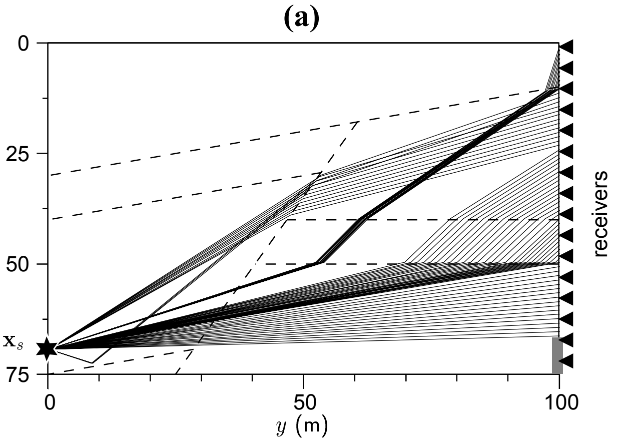


Figure 5. Model 1. Cost function versus take-off angles (θ_s and ξ_s). White circles show local minima that are solutions to the BVP (Table 2). Figs 5 and 7 may be viewed in colour in the online version of the journal (www.blackwell-synergy.com).



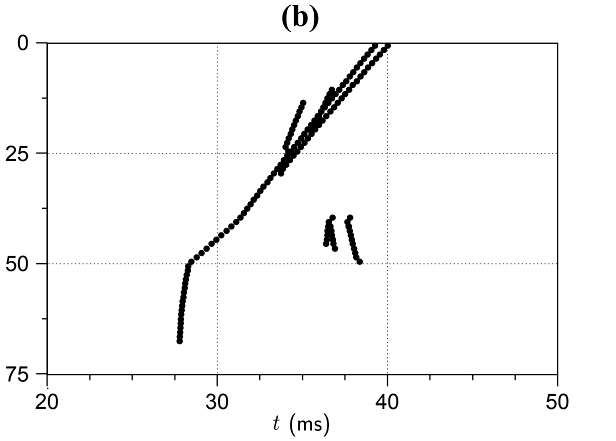


Figure 6. Model 1. (a) These 68 ray paths share the same source. Individually, they all correspond to global minima. Dashed lines show model interfaces. Short grey segment indicates shadow zone. (b) Traveltime versus receiver depth.

at (50, 0, 70) m. Fig. 6(a) shows the resulting ray paths and Fig. 6(b) the corresponding traveltimes. Here we also show all the solutions (first and later arrivals) to each BVP problem. Note that there are various nearly identical arrivals. Later arrivals at depths between 40 and 50 m correspond to multiply reflected waves (total reflection) generated in the low-velocity region. Geophones at depths below 67.5 m do not record any arrival and lie in a shadow zone. In this test we set $k_{\max}=750$, but often the solutions were obtained long before this number. The CPU time for obtaining all global minima was 4.8 s on a PC Pentium III 500 MHz. Table 3 shows the take-off angle search ranges.

4.1.2 *Salt-dome model*

Model 2 represents a salt dome with several layers and laterally varying velocities, as illustrated in Fig. 1. A total of eight regions are delimited by non-planar interfaces with cylindrical symmetry (see Table 1). The same symmetry, however, is not observed for all velocities. The model is bounded by planes $x=-50$ m, $x=50$ m, $y=-50$ m, $y=50$ m, $z=0$ m and $z=40$ m. Fig. 7 shows Φ as a function of both take-off angles for a source at (0, -50, 0) m and a receiver at (0, 50, 0) m. The topography of this surface is very complicated. It exhibits a large number of discontinuities and local minima, some of which correspond to solutions of the BVP. By inspecting this figure, the difficulties that a local minimization scheme would have in finding the

Table 3. Source location and take-off angle search ranges used in the numerical tests.

Model	$\mathbf{x}_s$ (m)	θ_s (°)	ξ_s (°)
1	(50, 0, 70)	[60, 140]	[50, 130]
2	(0, -50, 0)	[30, 90]	[60, 120]

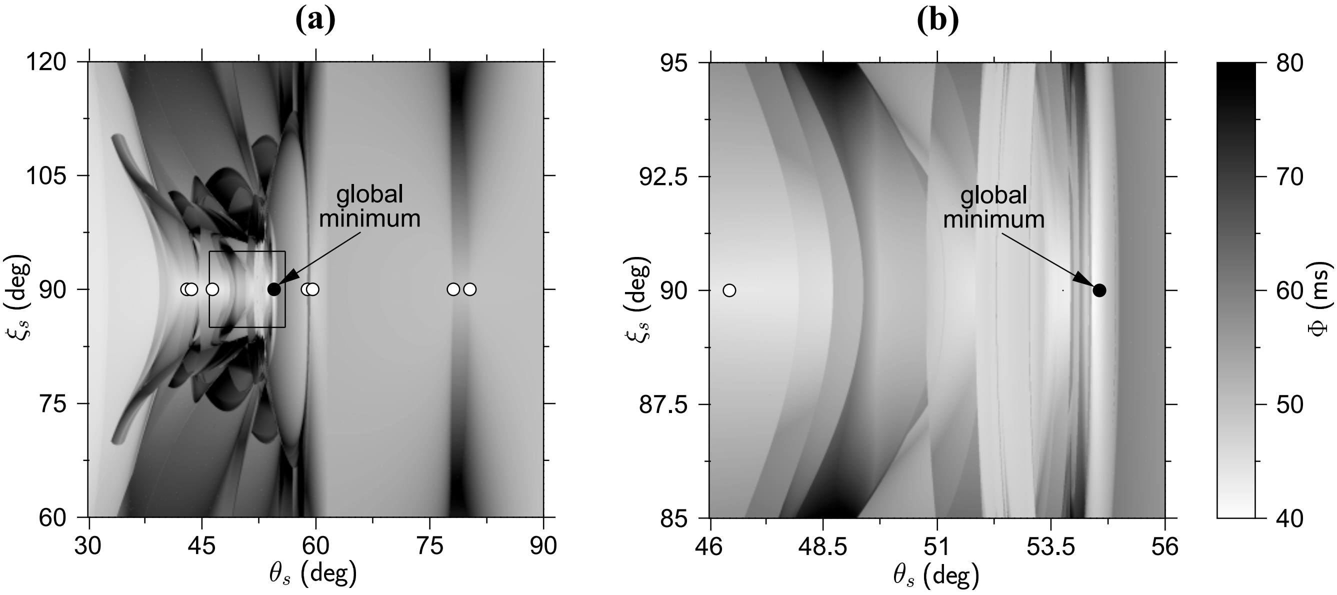


Figure 7. Model 2. (a) Cost function versus take-off angles. (b) Detail around the global minimum. White circles show local minima that are solutions to the BVP (Table 2).

global minimum are very clear. A detailed inspection of $\Phi(\theta_s, \xi_s)$ reveals that the global minimum lies in a narrow valley about 0.1° wide (see Fig. 7b). This makes the optimization problem very difficult. Despite the difficult nature of this optimization problem, SART found the global minimum in about 500 iterations:

$$\theta_s = 54.55^\circ\,, \quad \xi_s = 90.00^\circ\,, \quad \Phi_{\mathrm{opt}} = 39.733\ \mathrm{ms}\,.$$

Fig. 8 and Table 2 show all solutions to the BVP together with the optimum ray path obtained by SART.

As in the fault model case, we then used SART to find each global minimum for a set of 41 receivers uniformly distributed along a vertical line, $\mathbf{x}_{r_i} = (0,\ 50,\ i \times h)$ m, $i = 0, 1, \ldots, 40$, where $h = 1$ m is the geophone vertical spacing. In all cases the source was fixed at (0, −50, 0) m. Fig. 9(a) shows the resulting ray paths and Fig. 9(b) the corresponding traveltimes. Here we also show all the solutions (first and later arrivals) to each BVP problem. It can be seen that there are multiple arrivals at all geophone depths. All arrivals follow very different trajectories, but those whose traveltimes are a global minimum share basically the same path until they traverse the salt dome. Note that most of the solutions are within one of two very narrow beams. Ray paths connecting geophones at depths between 0 and 12 m have take-off angles in the range (54.522°, 54.550°). Ray paths connecting geophones at depths between 13 and 36 m have take-off angles in the range (53.499°, 53.564°). All these solutions were obtained by SART with $k_{\max} = 2000$ in about *35* s on a PC Pentium III 500 MHz. In all the examples we used the take-off angle search ranges shown in Table 3. It is worth mentioning that wave amplitudes are not calculated during the take-off angle search. Here, no conclusion is drawn about the observability of the waves propagating in these narrow beams.

4.2 Maximum number of iterations

Perhaps the most important VFSA parameter the user must select is the maximum number of iterations, $k_{\max}$. This value represents one of the stopping conditions of the algorithm. The cooling schedule is critical too. In this work we used the same annealing schedule that we described in previous sections for all the examples and obtained good results. Alternative schedules can be used to accelerate convergence, but care must be taken in order to avoid a premature freezing that may lead to local minima. An interesting discussion on this and other details of the SA method can be found in Ingber (1996).

Clearly, the lower the number of local minima and the smoother the cost function, the lower the number of iterations required for convergence to the global minimum. In the fault model, convergence is achieved very quickly since Model 1 is relatively simple. The complexity associated with Model 2 is enormous and the global minimum is very difficult to locate, yet convergence was obtained quite quickly (about 500 iterations). However, the multimodality of $\Phi(\theta_s, \xi_s)$ sometimes delays convergence, especially when local and global minimum traveltimes are very similar (this is the case of Model 1), or when the global minimum lies in a very small region of the model space (this is the case of Model 2). This is illustrated in Fig. 10, where we have plotted the convergence curves for 20 independent realizations (each realization corresponds to an SA run using different seeds for the random number generator utilized in the model perturbation stage of the algorithm), along with the cooling schedule. Note the various ‘plateaus’ in the convergence curves corresponding to local minima. These plateaus map into ‘clusters’ in the model space sampling process, as seen in Fig. 11. This figure shows scatter plots for 500 annealing iterations in the fault and salt-dome models for a single realization.

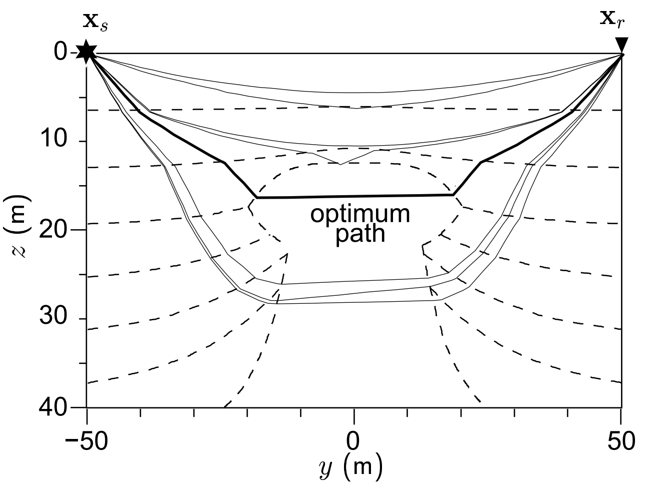


Figure 8. Model 2. The eight plotted ray paths satisfy the ray equations but exhibit different traveltimes. Dashed lines indicate model interfaces.

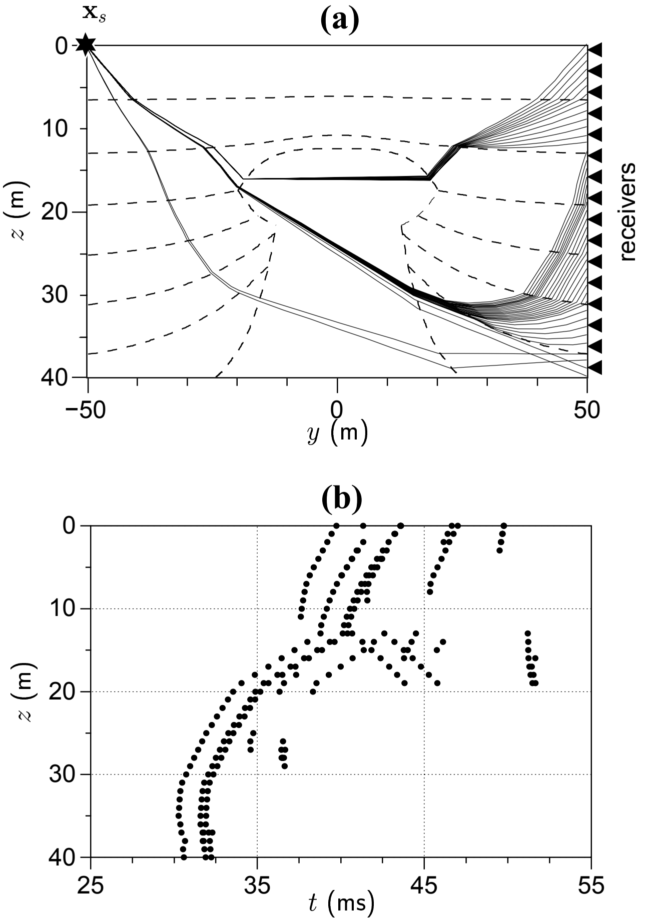


Figure 9. Model 2. (a) These 41 ray paths share the same source. Individually, they all correspond to global minima. Dashed lines indicate model interfaces. (b) Traveltime versus receiver depth.

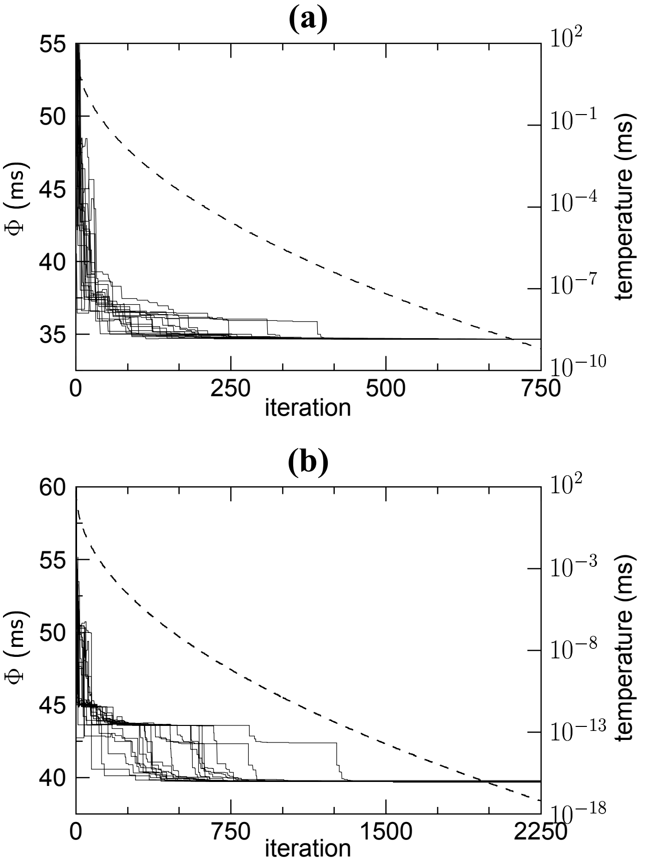


Figure 10. SART convergence for 20 independent realizations. (a) Fault model; (b) salt-dome model. Dashed line indicates cooling schedule

To provide better insight into the number of iterations required to reach the global minimum, the following experiment was conducted. SART was run 200 times using different seeds, for both Model 1 and Model 2, using the same single source–receiver pair in each run. Since the global minimum in each example was known, it was possible to obtain the mean number of iterations, $\bar{k}$, required to reach the global minimum within a certain accuracy, a. In this experiment the linearizing stage was not applied because we wanted to test how many SA iteration it would take to reach a specified cost. The results of the computations are shown in Table 4. The target cost Φ_a shown in column 3 is defined as

$$\Phi_a = (1 + a)\Phi_{\rm opt}\,, \qquad (13)$$

where $\Phi_{\rm opt}$ is the global minimum traveltime, and a, a small positive scalar, is tabulated in column 2. Column 6 shows the percentage of realizations, B, that fall within one standard deviation of the mean number of iterations. In all cases, $B \geq 90$ per cent. As expected, $\bar{k}$ increases with the desired accuracy. In the salt-dome case (Model 2), this number is much larger than in the fault model case (Model 1) because the associated cost function is much more complicated. Note that the use of the linearizing stage would have reduced $T_{\rm er}$ virtually to zero in a few iterations for all three as. This means that $a = 0.02$ (and consequently fewer iterations) would have been enough to obtain the global minimum very accurately. Finally, Fig. 12 shows the dispersion plots of the number of iterations for each example.

Table 4. Mean and standard deviation of the number of SA iterations required to achieve a given target cost after 200 realizations. Note that Φ_a is defined in eq. (13), with $\Phi_{\rm opt} = 34.714$ and 39.733 ms for models 1 and 2, respectively.

Model	a	Φ_a (ms)	$\bar{k}$	σ_k	B (%)	$\bar{T}_{\rm er}$ (ms)
1	0.020	35.408	168	145	93	0.536
	0.010	35.061	208	112	92	0.287
	0.001	34.749	437	136	90	0.030
2	0.020	40.529	770	552	95	0.978
	0.010	40.132	887	606	91	0.268
	0.001	39.774	1236	395	90	0.014

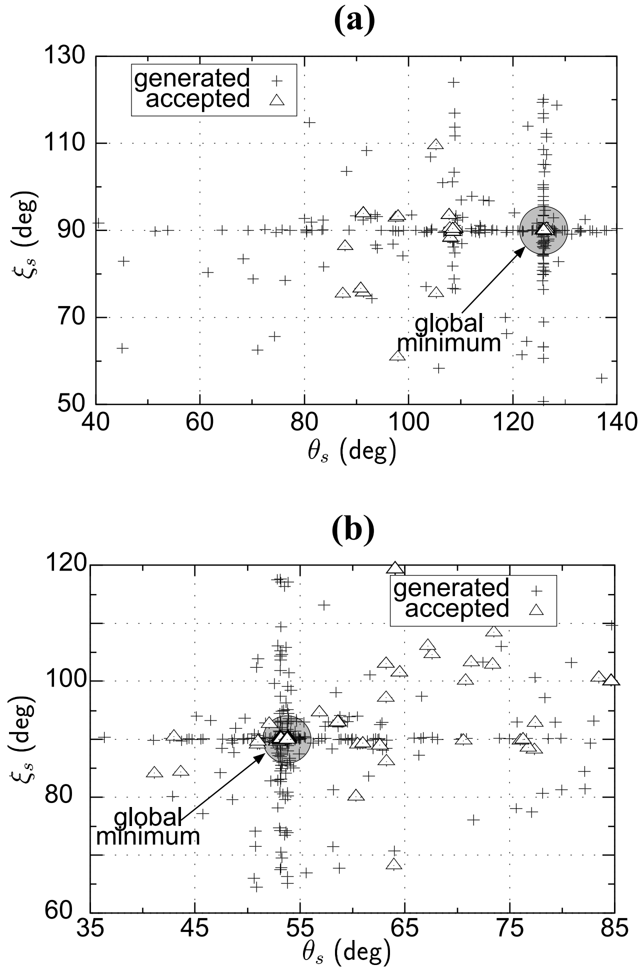


Figure 11. Scatter plot for 500 annealing iterations. (a) Fault model; (b) salt-dome model.

4.3 SART efficiency

In SART, each iteration involves the solution of the ray equations for a given set of initial conditions. Traveltime, $T_{\rm er}$, along the straight-ray construction also needs to be computed. At each time step, the algorithm also checks whether the ray has crossed any model interface. Usually this procedure, along with the calculation of the corresponding intersection points, is a very time-consuming stage, since all or some interfaces must be evaluated at every point of the ray trajectory. When Δt is

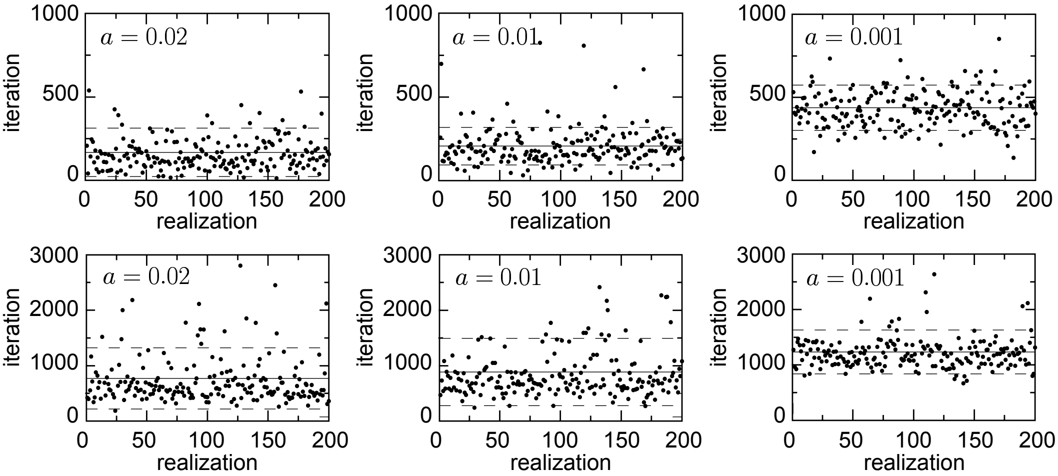


Figure 12. Dispersion plots of the number of iterations required to reach the global minimum with a given accuracy. Top row: fault model; bottom row: salt-dome model.

small and/or the number of interfaces is large, this process may account for the major part of the total computational time spent on each iteration. In turn, all these procedures are involved in the evaluation of the cost function at each iteration. The cost function evaluation is by far the most time-consuming process of SART, accounting for 95–99 per cent of the total computational effort.

4.3.1 Improvement of SART efficiency

Instead of using a fixed method for integrating the ray equations, in SART the integration is performed dynamically, allowing the user to specify which method to use in each model region. Every time the ray enters a new region, the appropriate method is automatically selected for optimum performance. Lower-order integration (e.g. Euler method) can be selected for very smooth regions. On the other hand, higher-order integration (e.g. fourth-order Runge–Kutta with adaptive step size) can be chosen for highly varying regions. This is the methodology we adopted throughout this work.

A significant computational cost can be saved whenever more than one BVP is to be solved for a given experiment. It seems reasonable to take advantage of previously computed cost functions to improve future SA runs. This is particularly useful for source–receiver pairs sharing the same source point. At each iteration, T_{er} in eq. (4) can be evaluated for all the remaining receivers sharing the same T_{se} value. This allows one to update iteratively the optimum solution associated with every source–receiver pair at no significant extra cost (in practice it is enough to compute T_{er} using large time steps and low-order integration). As an example, let $(\mathbf{x}_{\mathrm{s}}, \mathbf{x}_{\mathrm{r}_1})$ and $(\mathbf{x}_{\mathrm{s}}, \mathbf{x}_{\mathrm{r}_2})$ be two source–receiver pairs. After finding the solution to the first BVP, the cost function associated with the second BVP has already been evaluated a number of times. At this stage, despite the fact that the search has focused around the first solution, SART has taken advantage of the previously computed T_{se} values to find the solution of the second BVP. This procedure requires fewer iterations to reach the global minimum of the second BVP in comparison to running SA for each BVP independently (often the second BVP has already been solved after solving the first one). In the examples that follow, the maximum number of iterations for the last source–receiver pair is set, by default, equal to a quarter of the maximum number of iterations for the first source–receiver pair. For the other receivers, $k_{\max}$ varies linearly between these two values. Since the most time-consuming process is associated with the first portion of the ray path (from the source, $\mathbf{x}_{\mathrm{s}}$, to the emerging point, $\mathbf{x}_{\mathrm{e}}$), for a moderate number of receivers with a common shot, in general this strategy leads to a significant overall speed-up.

4.4 Comparison with other ray tracing methods

It is very difficult to make a fair comparison between various methods for calculating traveltimes because methods often use different model specifications and are devised for obtaining different kinds of traveltimes. In particular, one should make a strict distinction between first-arrival traveltimes and ray-theory traveltimes. Besides, one should keep in mind what kind of accuracy is needed for a certain application.

In this section we will make a comparison between SART, RKP (Runge–Kutta perturbation) and ART-PB (approximate ray tracing–pseudo-bending). RKP is an accurate shooting method based on Virieux (1991), whereas ART-PB is a combination of an approximate ray tracing method and a pseudo-bending method based on Thurber (1983) and Um & Thurber (1987). We used both RKP and ART-PB as implemented in the SIMULPS code (Eberhart-Phillips 1993; Haslinger 1998). A description of these two ray tracing methods can be found in Haslinger (1998).

In SIMULPS, both RKP and ART-PB use a model where velocities are defined on nodes of a grid that is interpolated using either 3-D cubic B-splines (RKP) or 3-D linear interpolation (ART-PB). For the comparison, in SART we generated a gridded model from the block model used in previous sections. This process was carried out by sampling the block model onto an evenly spaced 3-D grid. Velocity values at any point within the model boundaries are then obtained via 3-D cubic B-spline interpolation, bearing in mind that there is only one region and no interfaces. It is clear that one cannot expect that rays with the same source–receiver endpoints have the same paths and traveltimes in different velocity representations. However, we selected a fine grid in the three methods to ensure that rays are computed on equivalent velocity fields and to enable, to within a certain extent, a meaningful comparison. Clearly, this is a rather limited investigation, but we are primarily interested in some broad rather than detailed conclusions.

In the experiments we started with the salt-dome model shown in Fig. 1 and generated a gridded model with a spacing equal to 1 m. This resulted in a grid of $101 \times 101 \times 41$ equally spaced nodes. Also, we used a 3-D spatial Hanning window of radius R_h to smooth the model and to reduce possible instabilities of the ray tracing schemes. This process was necessary due to the extremely high velocity contrasts generated near the interface regions (note that in some cases this contrast was greater than 200 per cent). Using the same acquisition geometry as the previous examples (Fig. 9a), the two-point ray tracing problems were solved setting a tolerance receiver distance $\varepsilon = 0.01$ m (see eq. 9). Since RKP uses a fourth-order Runge–Kutta solver at the numerical integration stage, we selected the same solver for SART. The integration step in SART was chosen in such a way that the resulting ray paths had approximately the same number of segments as in RKP. In particular, we set $\Delta t = 0.125$ ms. Fig. 13 illustrates the velocity at every ray path time step in the block model and in two smoothed gridded models. The selected ray path corresponds to the global minimum solution for the receiver at (0, 50, 0) m.

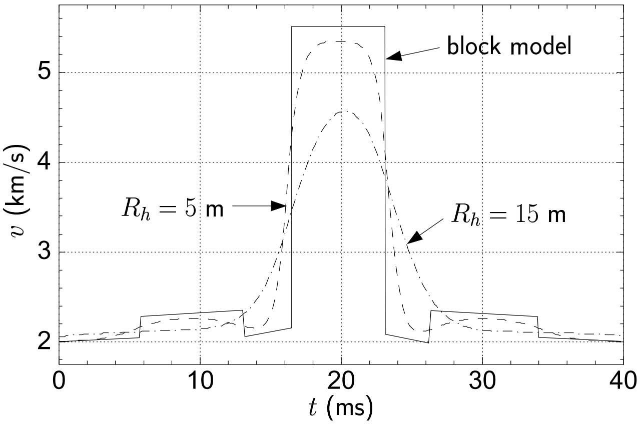


Figure 13. Model 2. Velocity along the global minimum ray path connecting (0, −50, 0) m with (0, 50, 0) m in the block model and in various smoothed gridded models.

We ran SART ($k_{\max} = 250$, 1000 and 2500), RKP and ART-PB and measured the total CPU time required to find all solutions in the gridded models with $R_h = 5$ m and $R_h = 15$ m (all calculations were performed on a PC Pentium III 500 MHz). For statistical purposes, each SART run was repeated five times using different seeds. The results are summarized in Table 5. The table shows the number of rays per second (that is, the number of source–receiver pairs divided by the total CPU time) and the mean traveltime relative error, which is calculated by averaging the quantity $100 \times (\Phi - \Phi_{\mathrm{opt}})/\Phi_{\mathrm{opt}}$. Here, global minima Φ_{opt} were assumed to be known after running SART with $k_{\max} = 5000$. Fig. 14 displays the relative errors in the two gridded models for each source–receiver pair using

Table 5. Comparison between SART ($k_{\max} = 250$, 1000 and 2500, five runs each), RKP and ART-PB in three different representations of Model 2. 'Ray s^{-1}' denotes the average number of rays traced per second and 'error' the mean relative error.

Method	$R_h = 5$ m		$R_h = 15$ m		Block model	
	ray s^{-1}	error (%)	ray s^{-1}	error (%)	ray s^{-1}	error (%)
SART (250)	0.48	3.977	0.51	1.109	5.88	4.397
SART (1000)	0.15	1.106	0.18	0.024	2.44	0.298
SART (2500)	0.07	0.010	0.07	0.005	0.98	0.121
RKP	0.68	3.246	1.47	0.093	–	–
ART-PB	11.36	1.350	13.51	0.119	–	–

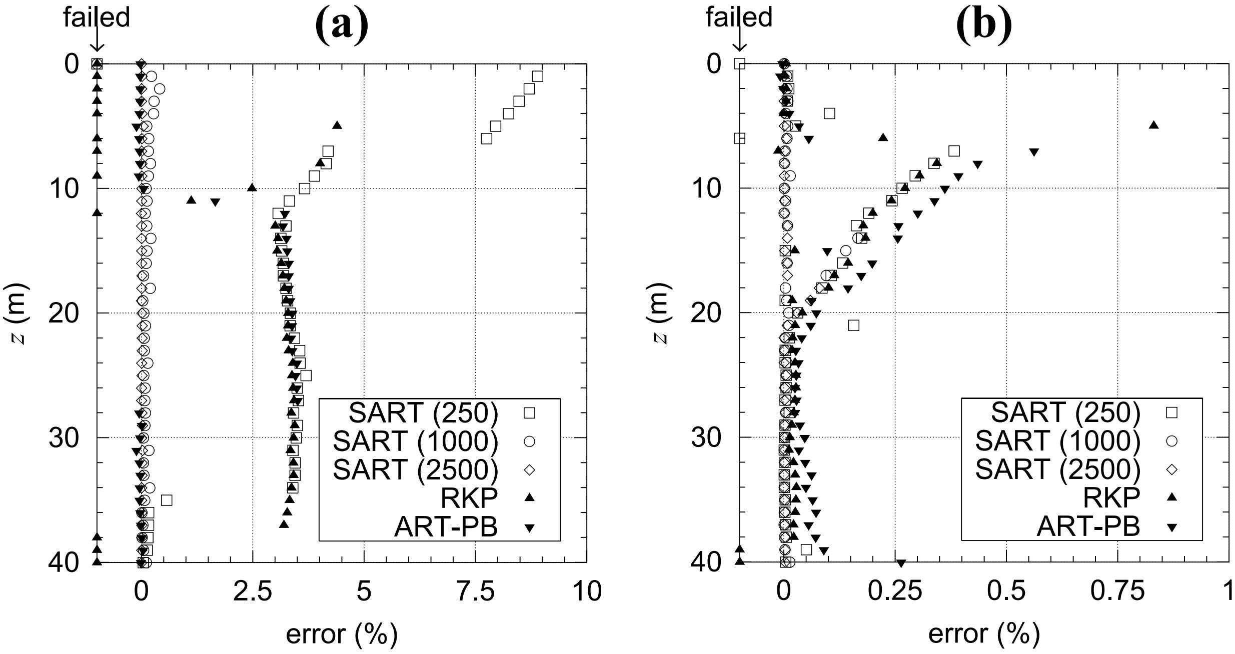


Figure 14. Model 2. Traveltime relative error using SART, RKP and ART-PB in the gridded salt-dome model with (a) $R_h = 5$ m and (b) $R_h = 15$ m. Note that RKP and SART (with $k_{\max} = 250$) were not able to connect some source–receiver pairs.

SART, RKP and ART-PB (in the case of SART, the figure shows median values).

By inspecting Table 5, it can be seen that ART-PB is much more efficient than both RKP and SART, SART being the slowest method in most cases. On the other hand, for $R_{\rm h}=5$ m, the accuracy of both ART-PB and RKP is fair to very poor because a large number of global minima (about 50 and 80 per cent, respectively) were missed (see Fig. 14a). In SART, the accuracy can be made very high by simply increasing $k_{\max}$. For $R_{\rm h}=15$ m, errors decrease by a factor of 10. Despite the fact that the smoothing introduced by the Hanning window with $R_{\rm h}=15$ m is quite significant, RKP and ART-PB still miss an important number of global minima (Fig. 14b). At this point it is important to note that some fraction of the errors of ART-PB might be due to differences in interpolation. However, the high correlation between ART-PB and SART ($k_{\max}=250$) errors observed in Fig. 14 indicates that ART-PB errors are mainly due to local minimum convergence.

For illustrative purposes only, Table 5 also shows the results for SART in the block salt-dome model (RKP and ART-PB are not suitable for these kinds of models). Here it was enough to set $\Delta t=1.0$ ms to obtain accurate traveltimes, since high velocity contrasts are taken into account by the interfaces. Note that the efficiency increased by one order of magnitude with respect to the gridded models for the same number of iterations. This is because the 3-D cubic B-spline interpolation scheme is very expensive.

5 CONCLUSIONS

SART is a versatile computational algorithm for solving the boundary value ray tracing problem in a general 2-D/3-D model. SART aims to find the ray path connecting any source–receiver pair so that the associated traveltime is a global minimum. The solution is found after iteratively solving a highly non-linear optimization problem by means of VFSA. At each iteration, SART numerically solves the equations derived from the high-frequency ray approximation theory. This scheme is coupled with a versatile model parametrization system that allows one to represent a wide variety of geological 3-D (or 2-D) structures. Any number of regions delimited by arbitrary interfaces can be defined. The velocity within each region is specified separately, which allows one to model any type of wave, including converted waves.

SART exhibits some improvements over existing ray tracing techniques such as bending and shooting. The problem of local minimum paths in complex structures is reduced significantly. Despite the fact that the traveltime is usually an ill-behaved function of the take-off angles in shooting-like methods, the selection of the appropriate initial angles does not represent a serious difficulty for SART.

It is emphasized that SART's goal is to find the solution that corresponds to the global minimum traveltime, and that the physical nature of the resulting ray is known *a priori* and *a posteriori*, a point that is very important in phase identification. Later arrivals, such as reflections and head waves, can be obtained by specifying a ray signature. This is done by penalizing those rays that do not follow the desired signature and deriving the appropriate cost function. The methodology can be extended to consider more complex ray paths such as multiples, higher-order head waves and diffractions and to include ray conversions of any type.

As opposed to bending and/or graph theory methods that require a large number of nodes to attain high-order accuracies at the expense of larger computing needs, in SART the accuracy can be made very high at no significant extra cost (by means of fourth-order Runge–Kutta integration, for example). The source of errors is limited then to the accuracy used in the calculation of the intersections between ray paths and interfaces, and the distance between the ray endpoint and the receiver. As explained above, these errors can be made very small by increasing the accuracy of the iterative root-finder and by polishing the final ray path via the steepest descent method.

In terms of computational cost, SART is expensive compared to conventional methods in gridded models. This is because several iterations are required to ensure global convergence. In contrast, although conventional shooting and bending methods usually converge faster, their convergence may only be local. SART is especially suited to complex 3-D block models, where the computational effort in solving the IVP is not as high as in gridded models with cubic B-spline interpolation, and where highly accurate traveltimes and paths can be obtained very efficiently. This trade-off between effectiveness and efficiency is a key issue the user should take into account when selecting a ray tracing algorithm. This is closely related to the kind of model at hand, the type of traveltimes required, and the accuracy to which the results are to be calculated.

ACKNOWLEDGMENTS

We are thankful to Martyn Unsworth, Cliff Thurber and Jean Virieux for critical comments and suggestions that helped to improve the quality of this work. We are also grateful to Florian Haslinger for his permission and help in using the ray tracing routines implemented in the SIMULPS code. This research was partially funded by the Consortium for the Development of Specialized Seismic Techniques (CDSST), University of British Columbia, Canada. DRV acknowledges support from Facultad de Ciencias Astronómicas y Geofísica (Universidad Nacional de La Plata), Agencia Nacional de Promoción Científica y Tecnológica (BID 802/OC–AR) and Consejo Nacional de Investigaciones Científicas y Técnicas (PIP 0363/98), Argentina.

REFERENCES

Červený, V., 1987. Ray-tracing algorithms in three-dimensional laterally varying layered structures, in *Seismic Tomography with Applications in Global Seismology and Exploration Geophysics,* ed. Nolat, G., pp. 99–133, E. Reidel, Dordretcht.

Cheng, N. & House, L., 1996. Minimum traveltime calculation in 3-D graph theory, *Geophysics,* **61,** 1895–1898.

Eberhart-Phillips, D., 1993. Local earthquake tomography: earthquake source regions, *Seismic Tomography: Theory and Practice*, pp. 613–643, eds Iyer, H.M. & Hiahara, K., Chapman & Hall, London.

Elissevnin, V., 1965. Analysis of rays propagating in an inhomogeneous medium, *Sov. Phys. Acoust.,* **10,** 242–245.

Fisher, R. & Lees, J., 1993. Shortest path ray tracing with sparse graphs, *Geophysics,* **58,** 987–996.

Haslinger, F., 1998. Velocity structure, seismicity and seismotectonics of Northwestern Greece between the Gulf of Arta and Zakynthos, *PhD thesis,* Federal Institute of Technology, Zürich.

Ingber, L., 1989. Very fast simulated re-annealing, *Math. Comput. Model.,* **12,** 967–973.

Ingber, L., 1993. Simulated annealing: practice versus theory, *Math. Comput. Model.*, **18,** 29–57.

Ingber, L., 1996. Adaptive simulated annealing (ASA): lessons learned, *Control Cybernetics,* **25,** 33–54.

Julian, B. & Gubbins, D., 1977. Three-dimensional seismic ray tracing, *J. Geophys.*, **43,** 95–113.

Kirkpatrick, S., Gellat, C.J. & Vecchi, M., 1983. Optimization by simulated annealing, *Science,* **220,** 671–680.

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

Mallet, J., Jacquemin, P. & Cheimanoff, N., 1989. GOCAD project: geometric modeling of complex geological surfaces, in *Expanded Abstracts,* vol. 89, p. 126, SEG, Dallas, TX.

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

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E., 1953. Equation of state calculations by fast computing machines, *J. Chem. Phys.*, **21,** 1087–1092.

Moser, T., 1989. Efficient seismic ray tracing using graph theory, in *Expanded Abstracts,* pp. 1106–1108, SEG, Dallas, TX.

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

Moser, T., Nolet, G. & Snieder, R., 1992. Ray bending revisited, *Bull. seism. Soc. Am.,* **82,** 259–288.

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

Press, W.H., Teukolsky, S., Vetterling, W. & Flannery, B., 1992. *Numerical Recipes in FORTRAN: the Art of Scientific Computing,* 2nd edn, Cambridge University Press, Cambridge.

Prothero, W., Taylor, W. & Eickemeyer, J., 1988. A fast, two-point, three-dimensional ray-tracing algorithm using a simple step search method, *Bull. seism. Soc. Am.,* **78,** 1190–1198.

Sambridge, M. & Kennett, B., 1990. Boundary-value ray tracing in a heterogeneous medium: a simple and versatile algorithm, *Geophys. J. Int.,* **101,** 157–168..

Sambridge, M., Braun, J. & McQueen, H., 1995. Geophysical parametrization and interpolation of irregular data using natural neighbours, *Geophys. J. Int.,* **122,** 837–857.

Sen, M. & Stoffa, P., 1995. *Global Optimization Methods in Geophysical Inversion,* Elsevier, Amsterdam.

Sun, Y., 1993. Ray tracing in 3-D media by parametrized shooting, *Geophys. J. Int.,* **114,** 145–155.

Thurber, C. & Ellsworth, W., 1980. Rapid solution of ray tracing problems in heterogeneous media, *Bull. seism. Soc. Am.,* **70,** 1137–1148.

Thurber, C.H., 1983. Earthquake locations and three dimensional crustal velocity structure in the Coyote lake area, central California, *J. geophys. Res.,* **88,** 8226–8236.

Um, J. & Thurber, C., 1987. A fast algorithm for two-point seismic ray tracing, *Bull. seism. Soc. Am.,* **77,** 972–986.

Velis, D.R., 1998. Application of simulated annealing to some seismic problems, *PhD thesis,* University of British Columbia, Vancouver, Canada.

Velis, D.R. & Ulrych, T.J., 1996. Simulated annealing two-point ray tracing, *Geophys. Res. Lett.,* **23,** 201–204.

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

Vinje, V., Iverson, E., Åstebøl, K. & Gjøystdal, H., 1996. Estimation of multivalued arrivals in 3D models using wavefront construction—part 1, *Geophys. Prospect.,* **44,** 819–842.

Virieux, J., 1991. Fast and accurate ray tracing by Hamiltonian perturbation, *J. geophys. Res.,* **96,** 579–594.

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