Geophys. J. Int. (1990) **101**, 157–168

Boundary value ray tracing in a heterogeneous medium: a simple and versatile algorithm

M. S. Sambridge* and B. L. N. Kennett
Research School of Earth Sciences, Australian National University, GPO Box 4, Canberra, ACT 1601, Australia

Accepted 1989 October 16. Received 1989 October 13; in original form 1989 May 1

SUMMARY

Traveltime calculations in 3-D velocity models have become more commonplace during the past decade or so. Many schemes have been developed to deal with the initial value problem, which consists of tracing rays from a known source position and trajectory usually towards some distant surface. Less attention has been given to the more difficult problem of boundary value ray tracing in 3-D. In this case, source and receiver positions are known and one, or more, minimum time paths are sought between fixed endpoints.

A new technique for boundary value ray tracing is proposed. The scheme uses a common numerical integration technique for solving the initial value problem and iteratively updates the take-off angles until the ray passes through the receiver. This type of 'shooting' technique is made efficient by using expressions describing the geometrical spreading of the wavefront to determine the relationship between the ray position at any time and the take-off angles from the source. The use of numerical integration allows the method to be compatible with a wide variety of structures. These include models with velocity varying smoothly as a function of position and those with arbitrarily orientated surfaces of discontinuity. An examination of traveltime accuracy is given as well as a discussion of efficiency for a few classes of velocity model.

To improve upon the first guess pair of take-off angles, a small-scale non-linear inverse problem must be solved. The difference between the receiver position and the arrival point of a ray, on a plane through the receiver, describe a mis-match surface as a function of the two take-off angles of the ray. The shape of this surface can possess local minima and multiple 'global' minima even for relatively simple 1-D velocity models. Its study provides some insight into the non-linearities of a small-scale geophysical inverse problem.

Key words: boundary value ray tracing, heterogeneous media.

1 INTRODUCTION

In recent years much work has been devoted to the production of efficient and accurate ray tracing schemes. Many authors have presented algorithms which trace rays through complicated heterogeneous structures. [For a review of techniques useful for laterally varying layered models see Červený (1987).] Usually the schemes are designed with a particular type of seismic velocity model in mind and perform well in that case. In fact many schemes are only applicable to a limited class of velocity models, and so the usefulness of any particular method will depend on whether the corresponding type of velocity model is appropriate for the problem under consideration. In this paper we present and discuss a relatively simple algorithm which performs two-point ray tracing in heterogeneous media and may be applied to a large class of velocity structures. It was developed as part of a 3-D traveltime inversion study and has been tested using various heterogeneous velocity fields, examples of which are presented to demonstrate its accuracy and efficiency.

1.1 Techniques for boundary value ray tracing

In a two-point, or boundary value ray tracing problem one must determine a ray path between fixed source and

* Now at Department of Earth Sciences, University of Cambridge, Downing Street, Cambridge CB2 3EQ, UK.

receiver positions through a known velocity structure; however, in the initial value case, one specifies the starting position and take-off angles and attempts to trace a ray, usually towards some target surface. The former problem is considerably more difficult to solve than the latter, especially when a complex velocity structure is involved. The initial value problem has simple analytical solutions only in a few special cases e.g. when the velocity $V(x, y, z)$, V^{-N} or $\ln(V)$ take a few special forms (see Červený 1987). Even in these cases the analytical expression for the ray path may become exceedingly complex and there usually exists no general solution to the two-point problem. In heterogeneous media one normally resorts to some numerical method to find a solution. Two types are in common use, namely 'bending' and 'shooting'.

The bending technique involves adjusting an initial path between source and receiver until Fermat's principle of stationary time is satisfied. Several algorithms have been put forward for 3-D structures; examples are those by Julian & Gubbins (1977), and Pereyra, Lee & Keller (1980) who both use a finite difference technique to solve a non-linear system of first-order differential equations. The method has been shown to work well in cases where the velocity field has no discontinuities in wavespeed. However when discontinuities are introduced, the formation of the problem becomes more complex. Internal boundary conditions have to be perturbed and linearized and also the order in which the ray encounters different surfaces of discontinuity has to be known *a priori*. These complications make the bending method less attractive for complex structures. The technique is better suited to simpler velocity models and is usually more efficient than the shooting method in these cases.

The shooting method is intrinsically less complicated than the bending method. In this scheme one solves the initial value problem using some first guess take-off angles for the ray and attempts to iteratively improve upon these until the ray hits the target location. The way in which the initial value problem is solved, and the improvement to the take-off direction obtained, is at the discretion of the user. (A fundamental difference between this and the bending method is that at each iteration of the shooting method the raypath is always a 'true' ray i.e. it satisfies Fermat's principle. In the bending method the ray is non-physical until convergence. At all intermediate stages the current 'ray' is merely a path connecting source and receiver.) In general the usefulness and accuracy of the shooting method in different classes of velocity structure will depend on the type of technique that is used to solve the initial value problem, whereas its efficiency will largely depend on the way the ray is improved at each iteration.

As mentioned above for special forms of velocity structure the initial value problem may be solved analytically. To extend this technique to more complicated structures one usually divides the velocity model into a series of cells within which the velocity field takes one of the special forms and so an analytical solution is possible. The complete ray path is found by combining the individual segments from each cell. For a 2-D model the cells are usually rectangular or triangular, and for a 3-D model either cubic or tetrahedral. This type of technique can be very useful for 2-D ray tracing when one wishes to shoot a whole family of rays from a shot point at depth to a line of geophones on the surface of the Earth, or even to solve the two-point problem in 2-D by performing a 1-D interpolation between traced rays. It has been applied to reflected rays in 2-D (Langan, Lerche & Cutler 1985; Williamson 1986) and teleseismic rays in 3-D, where rays enter the base of a model, made up of cubic cells, and are traced to the surface (Koch 1985). Analytical ray tracing is usually the most efficient way to shoot rays through a 3-D structure which may be conveniently represented by one of the special forms for V, or subdivided into cells within which V takes one of those forms. However it is not the most versatile of methods since a different set of analytical expressions are required for different forms of V and usually a structural rearrangement of the computational code is required for different cell shapes in 2- and 3-D. In cases where such velocity model constraints are unacceptable one must turn to other methods for solving the initial value problem. Indeed even in cases where such constraints are acceptable, a suitable procedure for improving the current guess take-off angles must be found which is compatible with the initial value solver.

The two-point method to be presented here places fewer constraints on the form of velocity model and is therefore more versatile than the class of methods discussed above. This is achieved by using numerical integration to solve the initial value problem, together with an efficient procedure to improve the take-off angles of the ray at each iteration and a modification to allow surfaces of velocity discontinuity in the model. Numerical integration has been used to solve the two-point problem in complex 2-D structures (Červený & Pšenčík 1981). The formulation of this approach is particularly simple since the equations governing the path of a ray may be written as a first-order system of ordinary differential equations (ODE's), and so the initial value problem may be solved by integrating the system numerically using either a Runge–Kutta or a Predictor–Corrector method. Usually the traveltime of the ray is chosen as the independent variable and the parameters describing the ray position and direction are solved for at successive time steps until some target plane has been reached. In order to integrate the ray equations, the velocity field must be differentiable. This constraint is not too serious since, in theory, it is possible to introduce surfaces of discontinuity in those parts of the model where the constraint does not hold, or it is not convenient to assume a continuous velocity field. When tracing a ray numerically between regions with continuous velocity fields the point at which the ray hits the interface must be found and the appropriate boundary conditions solved.

The procedure used to improve the ray's take-off angles at each iteration is usually considered the most problematic area of any shooting technique. The method employed here is similar to the paraxial boundary value ray tracing of Červený, Klimeš & Pšenčík (1984) (see also Červený 1987), which may be applied to two-point ray tracing in laterally varying layered models. The current algorithm differs in that one only calculates information about neighbouring rays that have a common endpoint, since this is all that is required to update the take-off angles at each iteration. The algorithm, its various extensions, a discussion of accuracy versus efficiency and some example applications appear below.

2 A TWO-POINT RAY TRACING ALGORITHM

2.1 The initial value ray tracer

The initial value formulation of the ray equations was first presented by Eliseevnin (1965). These may be written

$$\begin{aligned}
\partial_t x &= v\cos\sigma,\\
\partial_t y &= v\cos\beta,\\
\partial_t z &= v\cos\gamma,\\
\partial_t \alpha &= \frac{\partial v}{\partial x}\sin\alpha - \frac{\partial v}{\partial y}\cot\alpha\cos\beta - \frac{\partial v}{\partial z}\cot\alpha\cos\gamma,\\
\partial_t \beta &= \frac{\partial v}{\partial x}\cot\alpha\cot\beta - \frac{\partial v}{\partial y}\sin\beta - \frac{\partial v}{\partial z}\cot\beta\cos\gamma,\\
\partial_t \gamma &= \frac{\partial v}{\partial x}\cot\alpha\cot\gamma - \frac{\partial v}{\partial y}\cos\beta\cot\gamma - \frac{\partial v}{\partial z}\sin\gamma,
\end{aligned} \tag{1}$$

where ∂_t denotes differentiation with respect to time; x, y and z describe the endpoint position of the ray at a particular time, $v(x, y, z)$ is the wavespeed; and, $\cos\alpha$, $\cos\beta$ and $\cos\gamma$ are the local direction cosines related by the expression,

$$\cos\alpha^2 + \cos\beta^2 + \cos\gamma^2 = 1.$$

For convenience we use Cartesian coordinates throughout although the spherical form of these equations, and many others in this section, may be found in Julian (1970). Because of the relationship between direction cosines, only five of the equations in (1) are independent and therefore only five variables are required to describe the ray at any point of its trajectory. We introduce two new variables i and j, where i represents the angle the ray makes with the downward vertical, and j the angle between the vertical plane tangent to the ray and the x direction (see Fig. 1). These are related to the direction cosines by

$$\begin{aligned}
\cos\alpha &= \sin i\cos j,\\
\cos\beta &= \sin i\sin j,\\
\gamma &\equiv i.
\end{aligned} \tag{2}$$

Using (2), the ray equations simplify to

$$\begin{aligned}
\partial_t x &= v\sin i\cos j,\\
\partial_t y &= v\sin i\sin j,\\
\partial_t z &= v\cos i,\\
\partial_t i &= -\cos i\left(\frac{\partial v}{\partial x}\cos j + \frac{\partial v}{\partial y}\sin j\right) + \frac{\partial v}{\partial z}\sin i,\\
\partial_t j &= \frac{1}{\sin i}\left(\frac{\partial v}{\partial x}\sin j - \frac{\partial v}{\partial y}\cos j\right).
\end{aligned} \tag{3}$$

Since this is a system of first-order differential equations it may be solved with the use of most standard numerical integration techniques. Here we use a fourth-order Runge–Kutta algorithm throughout [details of which can be found in most books on numerical methods e.g. Conte & de Boor (1980)]. The method works by integrating the system (3) in a series of steps for the independent variable time. As we mentioned above, for a numerical integration to be possible, the right-hand side of (3) must be continuous along the ray. Both $v(x, y, z)$ and its gradients must therefore be continuous, and hence v must be twice differentiable. To handle discontinuities, rays must be traced to the interface and the appropriate jump conditions applied to the five independent variables x, y, z, i and j. Since the ray path is continuous, the spatial coordinates are continuous and jump conditions are required only for the angles i and j. These conditions are found by equating the component of the ray's slowness vector parallel to the interface on either side of the interface. The slowness vector at any point of the ray is defined simply as $\mathbf{S}(x, y, z, i, j)$ where

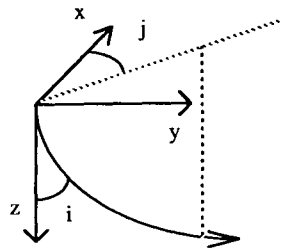

Figure 1. Two independent variables azimuth j, and declination i describing the direction of a ray.

$$\mathbf{S} = \frac{1}{v(x, y, z)}(\cos\alpha\,\mathbf{i} + \cos\beta\,\mathbf{j} + \cos\gamma\,\mathbf{k}). \tag{4}$$

If the normal to the interface is given by $\mathbf{n}$ then the resulting reflection/transmission law may be written,

$$\mathbf{S}_2 = \mathbf{S}_1 + \left\{\varepsilon\left[\frac{1}{v_2^2} - \frac{1}{v_1^2} + (\mathbf{S}_1\cdot\mathbf{n})^2\right]^{1/2} - (\mathbf{S}_1\cdot\mathbf{n})\right\}\mathbf{n} \tag{5}$$

where the indices refer to the media either side of the discontinuity and the ray is assumed to be passing from medium 1 to 2. The term ε equals sign $(\mathbf{S}_1\cdot\mathbf{n})$ and is known as the orientation index; ε equals $+1$ if $\mathbf{S}_1$ makes an acute angle with $\mathbf{n}$, and negative otherwise. All quantities on the right-hand side of (5) are known at the boundary and so the slowness vector in the new medium is easily determined. Once $\mathbf{S}_2$ is known the new direction angles may be obtained using (2) and (4). As it stands equation (5) represents the transmission case; for a reflected ray we set v_2 equal to v_1 and take the negative root in (5). We obtain

$$\mathbf{S}_2 = \mathbf{S}_1 - 2(\mathbf{S}_1\cdot\mathbf{n})\mathbf{n}. \tag{6}$$

Although, strictly speaking, we have only considered a plane wave at a planar interface, the above equations are locally valid even for curved rays at a curved interface (Červený 1987). By using (5) or (6) we may model either reflected or transmitted P- and S-waves, or even a conversion between the two. These expressions are sufficient to trace rays through a structure containing an arbitrary number of regions with different velocity fields separated by planar surfaces, as long as we can calculate the velocity and its first derivatives at any point of the model, and determine the unit normals to any interface at any point.

In order to apply the reflection/transmission (R/T) laws (5) and (6) the ray must be traced to hit the surface. Since the numerical integration of the system (3) is a time stepping method it is likely that the interface will be crossed during one particular step and not hit directly. To force the 'ray

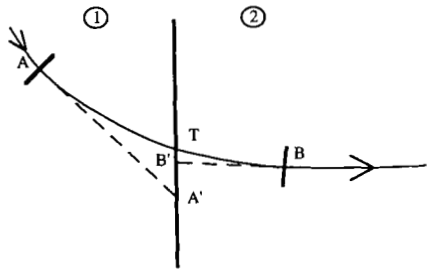

Figure 2. Diagram showing the plane of a ray when crossing the interface between medium 1 and 2. In one time step Δt the ray endpoint moves from A to B. To force the ray to hit the interface the step is repeated with $\Delta t_1 = \Delta t\,|AA'|/|AB|$ until B moves to T. Note all ray-path calculations are performed using the velocity field in medium 1 or its overlap into 2 so that the shape of the ray path remains constant and the correct intersection point T is found.

step' to hit the surface the step must be repeated with a reduced step length in order to bring the endpoint of the ray closer to the interface (see Fig. 2). This process is continued until the distance between the endpoint of the ray and the interface satisfies some tolerance condition. In practice a linear adjustment is usually sufficient to force the ray to hit the interface in two or three repeated steps. In practice it makes good sense to assume that medium 1 (in Fig. 2) overlaps the interface until the endpoint has converged on the interface. In this way all calculations of the ray step are performed without being influenced by medium 2 and so the correct intersection between the ray and the surface of discontinuity is found.

To complete the initial value part of the ray tracing algorithm we must decide when to halt the integration. Unless we are exremely fortunate the ray shot from our 'first guess' starting direction will not hit the desired target, or even come close. One possibility is to halt the ray when it reaches the horizontal plane passing through the target. This was done in all rays traced with the present algorithm. The problem of updating the initial direction then becomes one of finding the values of i_0 and j_0 such that the ray emerges at the correct x and y location on the target plane. (We could equally well stop the ray when it hits some vertical plane through the receiver, which may be more appropriate for a borehole to borehole experiment.)

2.2 Solving the two-point boundary value problem

The most common way of dealing with this problem is by the method of False position described by Julian & Gubbins (1977). If we let x_T and y_T be the coordinates of the desired endpoint of the ray on the horizontal target plane, then we have a set of two non-linear simultaneous equations relating x_T and y_T to the ray's take-off angles i_0 and j_0 of the form

$$\begin{aligned} x_c(i_0, j_0) &= x_T, \\ y_c(i_0, j_0) &= y_T, \end{aligned} \tag{7}$$

where x_c and y_c are the endpoint coordinates of the calculated ray. Usually one attempts to find the pair of take-off angles which satisfy (7) by linearizing the expression and iteratively solving the system,

$$\begin{pmatrix} \dfrac{\partial x}{\partial i_0} & \dfrac{\partial x}{\partial j_0} \\ \dfrac{\partial y}{\partial i_0} & \dfrac{\partial x}{\partial j_0} \end{pmatrix} \begin{pmatrix} i_0^{(n+1)} - i_0^{(n)} \\ j_0^{(n+1)} - j_0^{(n)} \end{pmatrix} = \begin{pmatrix} x_T - x_c^{(n)} \\ y_T - y_c^{(n)} \end{pmatrix} \tag{8}$$

where $\partial x/\partial i_0$, $\partial x/\partial j_0$, $\partial y/\partial i_0$ and $\partial y/\partial j_0$ are the partial derivatives of the calculated endpoint with respect to the take-off angles, and the superscripts refer to the iteration number. Hence $i_0^{(n+1)}$ and $j_0^{(n+1)}$ may be determined from $i_0^{(n)}$ and $j_0^{(n)}$ by inverting a simple (2×2) matrix of partial derivatives. Estimates of the partial derivatives are usually obtained by tracing three trial rays with slightly varying take-off angles, and fitting an imaginary plane through the three pairs of calculated x_c and y_c values. At all subsequent iterations the three previous rays are used to calculate the new derivatives in a similar way. The accuracy of the partial derivative estimation will depend upon the size of increment in i_0 and j_0 used. Although this may be fixed for the initial three trial rays, afterwards it is essentially determined by the linearized system (8). A poor estimate of the derivative matrix in (8) will at best result in slow convergence of the ray (increasing computation) and, at worst, complete failure, or divergence. The validity of the linearization approximation in (8) and the difficulty in calculating an accurate set of derivatives is primarily responsible for the notoriously slow nature of this type of procedure. Clearly to make the method more attractive a better estimate of the partial derivatives is required and may be obtained with a little extra work.

The basic problem is to determine by how much the endpoint position on the target plane varies when the take-off angles at the source are varied. Since the relationship between the position of a ray at any point along its trajectory and the direction is described by the curvature of the local wavefront, then the required information is contained in the curvature of the wavefront at the endpoint of the ray. Therefore the partial derivatives in (4) may be determined by solving the equations describing the geometrical spreading of the wavefront. These are given by Julian (1970) in spherical coordinates. The Cartesian equivalents are obtained from (3) by differentiating both sides with respect to the intitial take-off angles i_0 & j_0 and reversing the order of differentiation. This produces two more systems of first-order differential equations coupled to the original system via the ray-path parameters. The first of these has as its dependent variables the derivatives of x, y, z, i and j with respect to i_0, and the second the same variables with respect to j_0. We write the two sets of equations as

$$\left(\frac{\partial x}{\partial q}\right)' = \frac{Dv}{Dq} \sin i \cos j + v \cos i \cos j \frac{\partial i}{\partial q} - v \sin i \sin j \frac{\partial j}{\partial q},$$

$$\left(\frac{\partial y}{\partial q}\right)' = \frac{Dv}{Dq} \sin i \sin j + v \cos i \sin j \frac{\partial i}{\partial q} + v \sin i \cos j \frac{\partial j}{\partial q},$$

$$\left(\frac{\partial z}{\partial q}\right)' = \frac{Dv}{Dq}\cos i - v \sin i \frac{\partial i}{\partial q},$$

$$\left(\frac{\partial i}{\partial q}\right)' = \sin i \frac{\partial i}{\partial q}\left(\frac{\partial v}{\partial x}\cos j + \sin j \frac{\partial v}{\partial y}\right) + \frac{D}{Dq}\left(\frac{\partial v}{\partial z}\right)\sin i + \cos i \frac{\partial i}{\partial q}\frac{\partial v}{\partial z} - \cos i\left[\frac{D}{Dq}\left(\frac{\partial v}{\partial x}\right)\cos j - \sin i \frac{\partial j}{\partial q}\frac{\partial v}{\partial x} + \frac{D}{Dq}\left(\frac{\partial v}{\partial y}\right)\sin j + \cos j \frac{\partial j}{\partial q}\frac{\partial v}{\partial y}\right], \quad (9)$$

$$\left(\frac{\partial j}{\partial q}\right)' = -\frac{\cos i}{\sin^2 i}\frac{\partial i}{\partial q}\left(\frac{\partial v}{\partial x}\sin j - \cos j \frac{\partial v}{\partial y}\right) + \frac{1}{\sin i}\left[\frac{D}{Dq}\left(\frac{\partial v}{\partial x}\right)\sin j - \cos i \frac{\partial j}{\partial q}\frac{\partial v}{\partial x} + \frac{D}{Dq}\left(\frac{\partial v}{\partial y}\right)\cos j + \sin j \frac{\partial j}{\partial q}\frac{\partial v}{\partial y}\right],$$

where q represents either i_0 or j_0, $'$ denotes differentiation with respect to time, and

$$\frac{D}{Dq} \equiv \frac{\partial x}{\partial q}\frac{\partial}{\partial x} + \frac{\partial y}{\partial q}\frac{\partial}{\partial y} + \frac{\partial z}{\partial q}\frac{\partial}{\partial z}.$$

The terms on the right-hand side of (9) contain second-order derivatives of velocity and hence second-order smoothness is now required in the velocity field to prevent the introduction of an internal surface. By solving the two new sets of equations together with the original ray equations we have the variables on the left-hand side of (9) at any point along the ray path. These are closely related to the derivatives required in (8), however they apply over a constant time surface, i.e. along a wavefront, and not on a constant z plane. The appropriate adjustments are made by using the chain rule of partial derivatives in the form

$$\left.\frac{\partial}{\partial q}\right|_z = \left.\frac{\partial}{\partial q}\right|_t - \left.\frac{\partial z}{\partial q}\right|_t \left(\left.\frac{\partial z}{\partial t}\right|_q\right)^{-1} \left.\frac{\partial}{\partial t}\right|_q$$

where the time derivatives at constant q are given by the left-hand side of (3).

By introducing the two extra systems of equations (7) we are able to calculate the partial derivative terms in (8) directly, without making any crude approximation, and so the linerized system may be solved exactly. There is also no need to trace any trial rays, although it must be remembered that a system three times the size of the original is now being solved. The larger system of 15 equations is still of the same form as the original set in (3) and so exactly the same numerical integration technique may be used to solve it.

To deal with internal surfaces of discontinuity we must derive two new sets of boundary conditions for the 10 equations in (9). (Note we now treat an interface as a discontinuity in either the wavespeed or its first or second spatial derivatives.) These are obtained by considering the jump conditions of the ray equations at the interface and are given in the Appendix. It turns out that, in general, all of the terms involving $\partial i/\partial q$, $\partial j/\partial q$ etc. are discontinuous across an interface, for $q = i_0$ or j_0. These extra jump conditions introduced by (9) can easily be solved.

3 IMPLEMENTATION OF THE RAY TRACING SCHEME

3.1 Accuracy and efficiency

To try and describe how accurate traveltimes or ray paths will be in all situations would be futile. The size of errors introduced will depend almost entirely on the type of velocity model that rays are being traced through. However it is possible to give some indication of how error introduction may be monitored and minimized in any given situation. The shooting part of the procedure (the initial value solver) is the most important in terms of error introduction, since it is here that all ray tracing is performed. The numerical solution of (3) and (9) introduces an error into each of the dependent variables which is a function of the type of integration formula used and the size of the step taken. The error characteristics of the Runge–Kutta technique as a function of step size are well known and may be found in any numerical textbook dealing with the method. Usually the greatest concern is with the traveltime calculation, and how the error in the total traveltime varies with the step size in the numerical integration. Once this is understood then the largest step length (most efficient) numerical integration could be used that ensured a traveltime to within a specified accuracy. In the system (3) we have cast time as the independent variable and so a direct application of the Runge–Kutta error characteristics is not possible, since errors are estimated in the dependent variables for a given value of the independent variable. Obviously it is possible to recast the system with path length as the independent variable and introduce an extra equation in (3) to represent the traveltime (which would allow the cumulative traveltime errors to be monitored); however, in practice this problem can usually be solved with a few simple tests with the velocity model.

The accuracy with which a ray may be traced depends on the time step used in the numerical integration. Each time step corresponds to a movement of the ray endpoint (wavefront) and so the ratio of this distance to the scale length of velocity variation will determine the accuracy with which the ray path and traveltime may be calculated. Since vertical velocity gradients are likely to be much larger than lateral ones in most regional scale earth models then 1-D velocity models will provide a useful testing ground for the accuracy of the initial value solver. Tests were carried out using 1-D models, with constant vertical velocity gradients, and 3-D models with constant horizontal and vertical linear gradients. Source/receiver separations of between ~50 and 140 km were used which corresponded to travel times of between ~10 and 30 s. These indicated that for horizontal velocity gradients ~0.02 s^{-1} and vertical gradients ~0.05 s^{-1} a time step of 2.0 s resulted in a maximum error in traveltime of less than 0.02 s. The traveltime errors were calculated exactly by using analytical solutions. Again the results are dependent on the size of velocity gradients present in the model; however, they show that the time step can be much larger than the required error bound on traveltime.

The second factor which affects the accuracy of the traveltimes and the efficiency of the algorithm is the convergence criteria employed. (Note: since two-point ray tracing necessarily involves initial value ray tracing, then all error characteristics for the two-point case will include the effect of the initial value solver.) A prerequisite for the initial value solver is a tolerance allowed in the target plane position with respect to the target plane. The ray is defined to have hit the target plane when

$$|(z_c^{(n)} - z_T)| \leq Z_{TOL}. \tag{10}$$

For the complete two-point ray tracing, a tolerance in the position of the target on that plane must be decided, i.e.

$$\begin{aligned} |(x_c^{(n)} - x_T)| &\leq X_{TOL}, \\ |(y_c^{(n)} - y_T| &\leq Y_{TOL}, \end{aligned} \tag{11}$$

where $x_c^{(n)}$ and $y_c^{(n)}$ are the endpoints of the calculated ray, on the target plane, after the nth iteration of the algorithm. For each new iteration of the two-point ray tracer a new ray must be shot through the structure which requires more computation. If the area on the target plane defined by X_{TOL} and Y_{TOL} is large then the ray is more likely to converge on this area more quickly. However if the target area is too large then the difference in traveltime between the desired ray, which hits the receiver, and one which merely hits the target area may be larger than the required traveltime accuracy. Again the relationship between the traveltime error and the size of X_{TOL} and Y_{TOL} will depend on the velocity model under consideration. Some test results are displayed for three 3-D linear gradient models (Table 1a), a complex 1-D model shown in Fig. 3 (Table 1b) and a smoothly varying 3-D heterogeneous model represented by a set of 3-D splines on a regular cubic mesh of knots (Table 1c).

The 3-D linear gradient models used in Table 1(a) have horizontal gradients in the range 0.03–0.045 s^{-1}, and vertical gradients in the range 0.06–0.09 s^{-1}. The ray traced in Table 1(a) has an epicentral range of 100 km and therefore experiences lateral velocity increases of between 3.0 and 4.5 km s^{-1}. Although these are abnormally high values, the model is only used to test the performance of the two-point ray tracer in a linear gradient velocity field. The laterally heterogeneous velocity field was obtained by randomly perturbing the model in Fig. 3 at each node of a cubic mesh. The node spacing in the mesh was 30 km in both x and y directions and 10 km in z, and the maximum velocity perturbation at each node was fixed at 3 per cent. In all trials, rays were traced with source/receiver separations of about 170 km (simply because this scale length was appropriate for the application to which the ray tracing was designed). In the 3-D model the ray spans at least five mesh points, and therefore samples a reasonably heterogeneous velocity field in between. Both the 3-D linear gradient and laterally heterogeneous models contain a velocity jump across a horizontal surface at depth. The 3-D splines used in the latter impose second-order smoothness in the velocity field on either side of the interface. In this case no analytical traveltime calculations are possible, and so the traveltime 'error' here is merely a comparison to the calculated ray with a time step of 0.5 s. The considerable difference in traveltime error between the rays shown in Table 1(a) and those in Tables 1(b) and (c) is entirely due to the differences in tolerance levels for the target convergence X_{TOL} and Y_{TOL}. Even in the results for the largest time step it is clear that the errors due to the initial value solver are swamped by those due to the error in the target position. Tables 1(b) & (c) show much smaller traveltime errors because of the demand for an arrival closer to the target. It therefore seems reasonable to expect that, in most applications, the tolerance on target position will control the overall size of

Table 1. (a) Summary of ray tracing trials using three linear gradient models: ΔT is the traveltime error, ΔD is the distance between endpoint and target in km, and *Itns* is the number of iterations required to converge to within 0.5 km of the target. (b) Summary of rays traced through a complex 1-D model where the ray is now forced to converge to within 0.05 km from the target, and (c) heterogeneous 3-D model for the same tolerance.

a

	Model 1			Model 2			Model 3		
Step length (s)	ΔT	ΔD	Itns	ΔT	ΔD	Itns	ΔT	ΔD	Itns
1.0	.038	.37	4	.031	.038	3	.04	.38	4
0.5	.046	.41	4	.004	.042	3	.05	.42	4
0.2	.026	.13	4	.009	.009	3	.03	.12	4
0.1	.022	.10	4	.017	.046	3	.02	.10	4
|∇V| Horiz				~ 0.03 - 0.045 s^{-1}					
|∇V| Vert				~ 0.06 - 0.09 s^{-1}					

b

Complex 1-D model

Step length	Max ΔT	Average ΔD	Average Itns	No. of rays traced
1.0	~ 0.01	< 0.05	~ 5	25
2.0	~ 0.015	< 0.05	~ 5	25
3.0	~ 0.01	< 0.05	~ 4 - 5	25
4.0	~ 0.02	< 0.05	~ 4	25

c

Heterogeneous 3-D model

Step length	Max ΔT	Average ΔD	Average Itns	No. of rays traced
0.5	-	< 0.05	~ 7	24
1.0	~ 0.01 s	< 0.05	~ 7	24
4.0	~ 0.01 s	< 0.05	~ 8	25

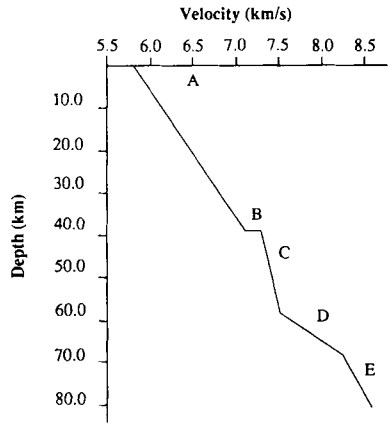

Figure 3. 1-D velocity model used in demonstration of the ray tracing algorithm.

traveltime error produced by the algorithm. When tuning the ray tracer to a particular type of velocity model this aspect must be kept in mind.

The calculations described here serve only as an illustration of the type of errors that may be introduced into the algorithm. All errors in traveltime will be reduced by decreasing the step length of the numerical integration, but obviously this will increase the computational cost. The best combination of step length and tolerances must be sort in any particular application. In the tests performed here, each two-point ray was found in under 0.5 s CPU on a Perkin–Elmer 3230. It is likely that some increase in efficiency could be gained by changing the time step as the take-off angles are improved i.e. beginning with a large step and improving the accuracy as the ray converged, since an accurate traveltime is only required when the final ray is traced. This has not been used in any of the trials performed here. The success of the two-point ray tracing is dependent on the update of take-off angles at each iteration. An examination of this area in more detail provides some interesting insight into the workings of the algorithm.

3.2 Updating the take-off angles: a geophysical optimization problem

To update the initial take-off angles i_0 and j_0 the linear system (8) must be solved at each iteration of the algorithm. In general, the relationship between i_0 and j_0 and the endpoint coordinates x_c and y_c on the target plane is non-linear, and so (8) is merely a local linearization about the current best guess $i_0^{(n)}$ and $j_0^{(n)}$. At each iteration we seek to reduce the mis-match between the calculated endpoint, determined by x_c and y_c, and the desired coordinates x_T and y_T. We may therefore view this problem as one of optimization. Equation (8) is rewritten in the form

$$\mathbf{G}_n \Delta \mathbf{m}_{n+1} = \Delta \mathbf{d}_n$$

where $\Delta \mathbf{m}_{n+1}$ is the vector representing the perturbation of the two parameters $i_0^{(n)}$ and $j_0^{(n)}$, $\Delta \mathbf{d}_n$ is the vector with components $x_T - x_c^{(n)}$ and $y_T - y_c^{(n)}$, and $\mathbf{G}_n$ is the matrix of partial derivatives calculated from the nth iteration. The distance between the endpoint of the ray on the target plane and the target is given by $(\Delta \mathbf{d}^T \Delta \mathbf{d})^{1/2}$, which may be represented as a surface in 3-D when plotted as a function of i_0 and j_0. For some simple velocity models it is possible to calculate the shape of the 'misfit surface' and thereby directly examine the type of optimization problem which results from (8).

As an example, consider the five-layer 1-D model in Fig. 3. By placing a source at a point $(0, 0, z_s)$ in the first layer we can calculate the range as a function of i_0 & j_0. In this case the correct azimuth j_0 is known immediately from the position of the target $(x_T, y_T, 0)$ and so we really have to consider only a 1-D optimization problem. Fig. 4 shows a plot of the distance of the calculated endpoint, from a source at 2 km depth and target at 167 km, as a function of i_0, and Fig. 5 shows how the misfit surface varies as a function of i_0 and j_0. A rather complex shape results even though the velocity model is relatively straightforward. The five distinctive sections of the curve (labelled A–E) may be directly attributed to the five segments of the velocity model. There are in fact three places where the curve

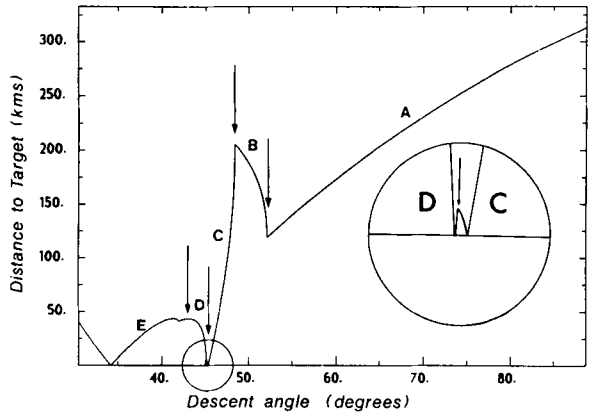

Figure 4. A plot of the distance from the target against declination angle of a ray i_0 for the correct azimuth of the target (taken as zero). The curve is a cut through the surface shown in Fig. 5 at $j_0 = 0$. Note the three places the curve touches the zero distance line and the two local minima.

touches the zero-distance line. The first two are close together in sections C and D respectively (see inset) and the third is within segment E. Each of these corresponds to a physical ray between source and target. In addition there exist two local minima, a minor one in section E and the more prominent one between sections A and B. Overall this presents a rather difficult optimization problem. The iterative solution of (8) will therefore depend crucially on the quality of the starting guess of i_0. Since the distance between the target and the endpoint of the ray is the square root of $(\Delta \mathbf{d}^T \Delta \mathbf{d})$, then a single iteration of (8) is approximately equivalent to moving along the tangent of the curve in Fig. 4, at the current i_0, until it crosses the axis. Obviously if our starting guess lies in segment C or the right-hand portion of segment D then we might expect to converge to either of the two closely space minima, whereas a value less than 40° would converge to the single minima. Any other would most probably lead to either of the local minima.

The existence of three physical rays between source and target is somewhat of a special case. It arises because the range of the target lies within a triplication of the traveltime curve for the velocity model. The single minima is actually the earliest arrival and is due to a ray bottoming in the deepest layer E, while the double minima are due to two near identical rays, one which grazes the interface between segments C and D from just above, and the other from just below. The triplication in the traveltime curve is a result of the velocity gradient in layer D being larger than both of its neighbours. The minor local minimum is also a consequence of the triplication. However, the major one is a result of the discontinuity in the model and poses some difficulties for the ray-tracing algorithm. If the initial guess for i_0 is a poor one then the algorithm may well become trapped in the significant local minimum. Unlike with most optimization problems, in this case it is possible to detect whether the algorithm is trapped in a local minimum of the misfit surface, since the distance to the target must be zero for a physical ray and all local minima have non-zero values. If a non-zero minimum is detected during the iterative solution of (8) we should invoke some sort of 'climb out' procedure

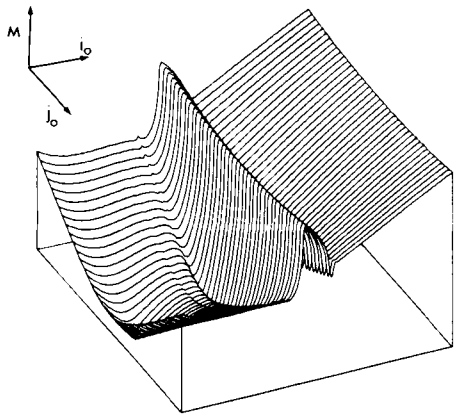

Figure 5. A plot of the misfit surface against the take-off angles i_0 and j_0 of a ray for the ranges $30° < i_0 < 70°$, $-22.5° < j_0 < 22.5°$. Note that amplitude of the major ridge decreases for an increase, or decrease, of j_0 away from the central zero value.

to escape i.e. force a perturbation to the current take-off angles sufficient to move them out of the local minimum. Obviously the size and direction of the perturbation necessary to overcome the effects of a local minimum will vary for different target distances and different velocity models. The procedure which has been most effective in the models considered here is to perturb only the initial declination $i_0^{(n)}$ by a pre-determined magnitude Δi_0, in a direction which depends on the horizontal range of the calculated ray. More precisely, if a local minimum is detected and

$$[(x_c^{(n)} - x_S)^2 + (y_c^{(n)} - y_S)^2]^{1/2} > [(x_T - x_S)^2 + (y_T - y_S)^2]^{1/2}$$

then $i_0^{(n)}$ is decreased by Δi_0, otherwise it is increased by Δi_0. One can introduce additional refinements by requiring that the perturbation, Δi_0, increase in magnitude if the algorithm falls back into the same local minimum, or even vary with target range. For the case in Fig. 4 a perturbation of $-6°$ is sufficient to move out of the local minimum. [It is interesting to note that the depth of the major local minimum seen in Fig. 5 actually decreases as the azimuth j_0 varies away from the true (central) value. In fact the 'valley' feature disappears completely when the azimuthal parameter is in error by about 50° (not shown), which suggests that one might overcome a local minima by making large swings in azimuth (at least in laterally homogeneous models).] For most rays, however, local minima can be avoided altogether if a reasonable starting value for i_0 is used. This value will also determine to which of the physical rays the algorithm converges and so some procedure for choosing an initial i_0 and j_0 will be important in heterogeneous 3-D models.

The existence of multiple phases is a real feature of the model and cannot be avoided. If one merely wishes to determine the earliest arrival and no other phases, then a procedure, which has proven to be useful in trials performed here, is to use the take-off angles of the ray which arrives first in the laterally averaged velocity model, as a first guess for i_0 and j_0. This is equivalent to Thurber & Ellsworth's (1980) 'ray initializer method'. For the 1-D problem described above these starting angles would be the correct values for the target and so no iteration would be required. It seems likely that the approach will be most useful for 3-D models when lateral velocity gradients are small compared with the vertical gradients. Another useful extension is to apply some step length damping i.e. take a fraction of the perburbation to i_0 and j_0 suggested by (8). Step length damping is commonly applied to locally linearized problems of this type to aid stability and increase efficiency. In trials performed here, damping was introduced if the misfit distance increased over any iterative step. In this case the step was repeated with half or three quarters of the original step length until the misfit decreased.

It is perhaps interesting to note that the determination of the initial take-off angles which give a physical ray is an example of a geophysical non-linear inverse problem whose exact solution can be found for some simple velocity models. In this discrete problem the 'data' are represented by the coordinates of the ray on the target plane and the sought after parameters are obviously i_0 and j_0. The study of this type of problem has been of interest to geophysicists for a number of years, and much literature has been devoted to the subject [see Menke (1984) for a review of discrete theory]. The most common approach is to cast the problem as one of optimization (as we have done here); however, it is rarely possible to examine the shape of the misfit function to determine whether any particular method has a reasonable chance of success. The relatively simple two-parameter problem dealt with here generates a rather complicated misfit surface even though it is the result of a very straightforward seismic problem. In particular we note that the surface contains not only significant local minima but also discontinuities in its first derivative which are manifested as 'crease'- like features in the misfit surface (see Figs 4 and 5). With this example in mind then we should perhaps be rather cautious when approaching other geophysical inverse problems which are 'solved' by means of optimization techniques.

In light of the above comments, the two-point ray tracer should be put to a more thorough test to see whether it is able to avoid local minima in the ways suggested. An additional 5000 rays were traced through the model in Fig. 3 using varying source–receiver distances and source depths. In this test a total of 20 rays failed to converge (0.4 per cent) on the correct first arrival between source and receiver. These were all found to be cases where the climb-out procedure ($\pm 6°$) was insufficient to escape from the local minima. In fact all of these local minima possessed a misfit value very close to zero i.e. the ray converged on a position close to the target but outside of our tolerance limits. The efficiency of the algorithm also decreased considerably for rays whose true declination i_0 was close to $\pi/2$ i.e. a near horizontal take-off from the source. Even though the model is actually 1-D here, the few cases of failure give a good indication of when the two-point algorithm has difficulties in finding a solution. Experience with fully 3-D laterally heterogeneous models shows that the cases of failure increase by at least a factor of two or three. In these trials the heterogeneous models were obtained by maximum 3 per cent random perturbations of the model in Fig. 3 in the same way as described above, although it is important to note that the cases of failure were again associated with local minima caused by discontinuities in the original 1-D model and not due to the lateral heterogeneity alone. The

more complex laterally heterogeneous model simply causes more rays to be trapped in the minima produced by discontinuities.

Overall the two-point ray tracer proposed has been reasonably successful in complex 1-D media and moderately perturbed 3-D media. In the types of velocity model considered here, no shadow zones were generated. However in general 3-D heterogeneous media such effects may well be present. In such cases no ray can be found if the receiver lies in a shadow of the source, and the algorithm proposed here will, at best, merely oscillate about the boundary of the shadow zone on the target plane. However, because the proposed two-point scheme is of a shooting type, it is possible to investigate whether shadow zones are likely to exist beforehand i.e. by shooting a spread of rays through the structure and examining the arrival distribution on the target plane. In this way the user may at least be aware of their existence and avoid needless calculations with the two-point scheme.

For highly irregular 3-D velocity fields it is conceivable that the take-off angle/endpoint relationship may become chaotic i.e. small perturbations of the take-off angles lead to large and mostly uncorrelated variations in the position of the ray on the target plane. This effect has not occurred in any of the tests performed here; however, if it had, then the ray tracing technique would obviously fail.

4 DISCUSSION

In the preceding section we have attempted to point out the areas where the proposed technique may suffer from error introduction and the type of situations where it is likely to encounter difficulty. We have been able to make several suggestions for dealing with these situations as they arise. The major strengths of the approach lie in its versatility and to a certain extent in its 'tunable accuracy'. Its versatile nature is a result of using numerical integration for the initial value solver. In this way few demands are made on the form of the velocity model; in particular, one does not have to divide it up into a series of regular shaped cells, and so the algorithm may be applied to a wide class of velocity models. Discontinuities in the velocity field can be handled by using the appropriate expressions in the Appendix, although it is incumbent upon the user to provide the value of the local wavespeed and derivatives in all parts of the velocity model and to detect when the ray has crossed a pre-defined discontinuous surface. Even in 3-D, this is essentially a geometrical problem and can usually be solved in most cases. [For a typical example see the 3-D crustal velocity model in Fig. 6 (derived from the work of Lambeck, Burgess & Shaw 1988) which contains many intersecting planes across which the velocity is discontinuous. In this case it is possible to detect in which region the end point of the ray lies by subdividing the model into a group of concave prisms i.e. all vertices are less than or equal to 90°. Any type of velocity field may then be used inside of each prism and the initial value ray tracing is accomplished without difficulty.] Finally it is worthwhile noting that the partial derivatives in (8) may be used to calculate amplitude attenuation due to geometrical spreading (see Julian 1970).

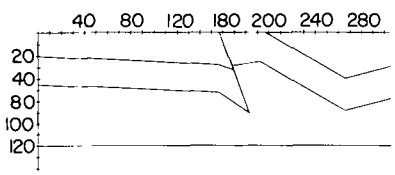

Figure 6. A diagram to illustrate the type of velocity model with many planar interfaces that can be handled by the initial value ray tracer. Across each interface the local velocity is discontinuous. The velocity varies in only two dimensions although the model itself is extended out of the plane in the diagram, and rays may be traced in all three dimensions.

Although a Cartesian coordinate system has been used throughout, equations (3) and (9) may equally well be formulated in spherical coordinates, which can also be found in Julian (1970). However to ensure numerical stability, the initial value problem may require a coordinate transformation when rays are close to turning points (see Červený 1987). The two-point ray tracing scheme proposed here has been successful in the velocity models described above. However there can be no guarantee that it will be equally successful in all cases. Ultimately the user must decide whether it forms a viable approach for his or her own particular problem.

ACKNOWLEDGMENTS

M.S.S. would like to acknowledge the support of an H.O. Wood Fellowship from the Carnegie Institution of Washington, Department of Terrestrial Magnetism during the preparation of this paper. He would also like to thank David Gubbins for some early discussions on 3-D ray tracing and the initial impetus for the approach taken.

REFERENCES

Červený, V., 1987. Ray tracing algorithms in three-dimensional laterally varying layered structures, in *Seismic Tomography*, pp. 99–133, ed. Nolet, G., Reidel, Dordrecht.

Červený, V. & Pšenčík, I., 1981. 2D seismic ray package, *Research Report*, Charles University, Prague.

Červený, V., Klimeš, L. & Pšenčík, I., 1984. Paraxial ray approximation in the computation of seismic wavefield in inhomogeneous media, *Geophys. J.R. astr. Soc.*, **79**, 89–104.

Conte, S. D. & de Boor, C., 1980. *Elementary Numerical Analysis, an Algorithmic Approach*, 3rd edn, McGraw-Hill, Singapore.

Eliseevnin, V. A., 1965. Analysis of rays propagating in an inhomogeneous medium, *Sov. Phys. Acoust.* (English Translation) **10**, 242–245.

Julian, B. R., 1970. Ray tracing in arbitrarily homogeneous media, *Technical Note 1970–45*, Lincoln Laboratory, Massachusetts Institute of Technology.

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

Koch, M., 1985. A numerical study on the determination of 3-D structure of the lithosphere by linear and nonlinear inversion of teleseismic travel times, *Geophys. J.R. astr. Soc.*, **80**, 73–93.

Lambeck, K., Burgess, G. & Shaw, R. D., 1988. Teleseismic travel-time anomalies and deep crustal structure in central Australia, *Geophys. J.*, **99**, 105–124.

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

Menke, W., 1984. *Geophysical Data Analysis: Discrete Inverse Theory*, Academic Press, Orlando, FL.

Pereyra, V., Lee, W. H. K. & Keller, H. B., 1980. Solving two-point seismic ray tracing problems in a heterogeneous medium, Part I: A general adaptive finite difference method, *Bull. seism. Soc. Am.*, **70**, 79–99.

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

Williamson, P. R., 1986. Tomographic inversion of travel time data in reflection seismology, *PhD thesis*, Cambridge University, UK.

APPENDIX: JUMP CONDITIONS FOR THE GEOMETRICAL SPREADING EQUATIONS ACROSS AN INTERNAL SURFACE OF DISCONTINUITY

Here we give the jump conditions for all ten dependent variables of the geometrical spreading equations (9) across an interface involving a discontinuity in wavespeed, its first or second derivatives. These expressions are well known and may be derived from the R/T law, equations (5) and (6), for both a reflected and transmitted ray.

(a) The transmitted ray

For a known source and velocity field, the position of a ray's endpoint is completely determined by its take-off angles at the source, (i_0, j_0), and the traveltime t_0. We may therefore write $\mathbf{x}(j_0, j_0, t_0)$, where $\mathbf{x}$ is the vector representing the endpoint coordinates. The position of the endpoint at some later time t, is written $\mathbf{x}(i_0, j_0, t)$. Performing a first-order Taylor expansion of $\mathbf{x}$ about t_0 gives

$$\mathbf{x}(i_0, j_0, t) = \mathbf{x}(i_0, j_0, t_0) + \frac{\partial \mathbf{x}}{\partial t}\bigg|_{t=t_0} (t - t_0)$$
$$= \mathbf{x}(i_0, j_0, t_0) + v^2\mathbf{S}(t - t_0) \qquad \text{(A1)}$$

where $\mathbf{S}$ is the slowness vector at time t_0, given by (4), and we have made use of the first three equations in (1). If we let t_0 be the traveltime at the interface between media I and II, then from (A1) we find that the position of the ray at a later time t, in medium II, is given by

$$\mathbf{x}_{\mathrm{II}}(i_0, j_0, t) = \mathbf{x}_{\mathrm{II}}(i_0, j_0, t_0) + v_2^2\mathbf{S}_2(i_0, j_0, t_0)(t - t_0). \qquad \text{(A2)}$$

Since the ray is continuous across the interface we have

$$\mathbf{x}_{\mathrm{II}}(i_0, j_0, t) = \mathbf{x}_{\mathrm{I}}(i_0, j_0, t_0) + v_2^2\mathbf{S}_2(i_0, j_0, t_0)(t - t_0). \qquad \text{(A3)}$$

If we write q as either i_0 or j_0, then the jump conditions for the first six dependent variables in (9) are found by differentiating (A3) with respect to q along the interface. To do this we must remember that at the interface the traveltime t_0 is a function of the take-off angles i_0 and j_0, and so the interface derivative takes the form

$$\frac{\partial \mathbf{f}}{\partial q}(i_0, j_0, t_0) = \frac{\partial \mathbf{f}}{\partial q}(i_0, j_0, t)\bigg|_{t=t_0} + \frac{\partial \mathbf{f}}{\partial t}(i_0, j_0, t)\bigg|_{t=t_0} \frac{\partial t_0}{\partial q}. \qquad \text{(A4)}$$

Using (A4) we obtain upon differentiation of (A3) with respect to q

$$\frac{\partial \mathbf{x}_{\mathrm{II}}}{\partial q}(t) = \frac{\partial \mathbf{x}_{\mathrm{I}}}{\partial q}(t')\bigg|_{t'=t_0} + \frac{\partial \mathbf{x}_{\mathrm{I}}}{\partial t'}(t')\bigg|_{t'=t_0} \frac{\partial t_0}{\partial q}$$
$$+ \frac{\partial}{\partial q}(v_2^2\mathbf{S}_2)(t - t_0) - v_2^2\mathbf{S}_2 \frac{\partial t_0}{\partial q}$$

where we have dropped the arguments i_0 and j_0 for convenience. Evaluating this expression at $t = t_0$ and using the first three equations from (1), in medium I, we have

$$\frac{\partial \mathbf{x}_{\mathrm{II}}}{\partial q} = \frac{\partial \mathbf{x}_{\mathrm{I}}}{\partial q} + (v_2^2\mathbf{S}_1 - v_2^2\mathbf{S}_2)\frac{\partial t_0}{\partial q}. \qquad \text{(A5)}$$

Equation (A5) (for $q = i_0$ or j_0) provides the jump conditions for the six spatial derivative variables in the geometrical spreading equations (9). All terms on the right-hand side are known except $\partial t_0/\partial q$ which can be obtained from the equation of the interface.

$$\Sigma\{\mathbf{x}[i_0, j_0, t_0(i_0, j_0)]\} = 0 \qquad \text{(A6)}$$

and so

$$\frac{\partial \Sigma}{\partial q} = 0$$

or

$$\underline{\nabla}\Sigma \cdot \left(\frac{\partial \mathbf{x}_{\mathrm{I}}}{\partial q}\bigg|_{t=t_0} + \frac{\partial \mathbf{x}_{\mathrm{I}}}{\partial t}\bigg|_{t=t_0} \frac{\partial t_0}{\partial q}\right) = 0. \qquad \text{(A7)}$$

Rearranging we get

$$\frac{\partial t_0}{\partial q} = -\frac{\mathbf{n} \cdot \dfrac{\partial \mathbf{x}_{\mathrm{I}}}{\partial q}}{v_1^2(\mathbf{S}_1 \cdot \mathbf{n})} \qquad \text{(A8)}$$

where we have again used equation (1) and $\mathbf{n}$ is the unit normal to the interface at the point of intersection. Since all terms on the right-hand side of (A8) are known, the interface derivative $\partial t_0/\partial q$ may be calculated for $q = i_0$ or j_0.

The jump conditions for the remaining four angular derivatives in (9) are found in a similar way to the spatial derivatives. First we expand the slowness vector at the interface, in medium II, using a first order Taylor series

$$\mathbf{S}_2(i_0, j_0, t) = \mathbf{S}_2(i_0, j_0, t_0) + \frac{\partial \mathbf{S}_2}{\partial t}\bigg|_{t=t_0} (t - t_0) \qquad \text{(A9)}$$

where t_0 is again the traveltime to the interface. Using the R/T law for transmission, equation (5), and the last three ray equations in (1) we may rewrite (A9) in the form,

$$\mathbf{S}_2(i_0, j_0, t) = \mathbf{S}_1(i_0, j_0, t_0) + (\varepsilon Q - P)\mathbf{n} - v_2^{-1}\underline{\nabla}v_2(t - t_0) \qquad \text{(A10)}$$

where $P = (\mathbf{S}_1 \cdot \mathbf{n})$, $Q = (v_2^{-2} - v_1^{-2} + P^2)^{1/2}$ and both depend on i_0, j_0 and t_0. Differentiating (A10) with respect to q along the interface using (A4) we obtain

$$\frac{\partial \mathbf{S}_2}{\partial q}(t) = \frac{\partial \mathbf{S}_1}{\partial q}(t_0) + \left(\varepsilon\frac{\partial Q}{\partial q} - \frac{\partial P}{\partial q}\right)\mathbf{n} + (\varepsilon Q - P)\frac{\partial \mathbf{n}}{\partial q}$$
$$- \frac{\partial}{\partial q}(v_2^{-1}\underline{\nabla}v_2)(t - t_0) + v_2^{-1}\underline{\nabla}v_2 \frac{\partial t_0}{\partial q}.$$

Evaluating this expression for $t = t_0$ we get

$$\frac{\partial \mathbf{S}_2}{\partial q}(t_0) = \frac{\partial \mathbf{S}_1}{\partial q}(t_0) + \left(\varepsilon\frac{\partial Q}{\partial q} - \frac{\partial P}{\partial q}\right)\mathbf{n}$$
$$+ (\varepsilon Q - P)\frac{\partial \mathbf{n}}{\partial q} + v_2^{-1}\underline{\nabla}v_2 \frac{\partial t_0}{\partial q}. \qquad \text{(A11)}$$

This vector equation has three components which allow us to determine the new values of $\partial i/\partial q\,|_{II}$ and $\partial j/\partial q\,|_{II}$ (for $q = i_0$ or j_0) from known quantities in medium I. When all terms are evaluated it will simply be a system of three linear equations in two unknowns (one equation is redundant) and can therefore be solved easily. To complete the description of the jump conditions for the angular derivatives we must write all remaining unknowns in (A11) either in terms of known quantitites in medium I, or the unknown variables $\partial i/\partial q\,|_{II}$ and $\partial j/\partial q\,|_{II}$. We consider each term in succession. The left-hand side can be rewritten by using the definition of the slowness vector in (4). The three components become

$$
\begin{aligned}
(x)\colon\ & v_2^{-1}\cos j\cos i\frac{\partial i}{\partial q}\bigg|_{II} - v_2^{-1}\sin j\sin i\frac{\partial j}{\partial q}\bigg|_{II} \\
& - v_2^{-2}\frac{Dv_2}{Dq}\bigg|_{II}\sin i\cos j, \\
(y)\colon\ & v_2^{-1}\cos j\sin i\frac{\partial j}{\partial q}\bigg|_{II} + v_2^{-1}\sin j\cos i\frac{\partial i}{\partial q}\bigg|_{II} \\
& - v_2^{-2}\frac{Dv_2}{Dq}\bigg|_{II}\sin i\sin j, \\
(z)\colon\ & -\frac{\sin i}{v_2}\frac{\partial i}{\partial q}\bigg|_{II} - v_2^{-2}\frac{Dv_2}{Dq}\bigg|_{II}\cos i,
\end{aligned}
\tag{A12a}
$$

where

$$
\frac{Dv_2}{Dq}\bigg|_{II} = \frac{\partial v_2}{\partial x}\left(\frac{\partial x}{\partial q}\right)_{II} + \frac{\partial v_2}{\partial y}\left(\frac{\partial y}{\partial q}\right)_{II} + \frac{\partial v_2}{\partial z}\left(\frac{\partial z}{\partial q}\right)_{II} \tag{A12b}
$$

and the terms in brackets are given by the jump conditions already determined in (A5). [Note: since the left-hand side of (A11) is, in general, a function of t and not t_0, we did not have to invoke (A4) in differentiating $\mathbf{S}_2(t)$.] Since all other terms are known, the three expressions in (A12a) are linear functions of the unknowns $\partial i/\partial q\,|_{II}$ and $\partial j/\partial q\,|_{II}$. The right-hand side of (A11) is more involved but can be written entirely in terms of known quantities. The last vector term is already known from the derivatives of the velocity field and equation (A8). The three components of the first term may be expanded in a similar way to the left-hand side, only now both i and j are functions of t_0. We have

$$
\begin{aligned}
(x)\colon\ & v_1^{-1}\cos j\cos i\frac{Di}{Dq}\bigg|_{I} - v_1^{-1}\sin j\sin i\frac{Dj}{Dq}\bigg|_{I} \\
& - v_1^{-2}\frac{Dv_1}{Dq}\bigg|_{I}\sin i\cos j, \\
(y)\colon\ & v_1^{-1}\cos j\sin i\frac{Dj}{Dq}\bigg|_{II} + v_1^{-1}\sin j\cos i\frac{Di}{Dq}\bigg|_{I} \\
& - v_1^{-2}\frac{Dv_1}{Dq}\bigg|_{I}\sin i\sin j, \\
(z)\colon\ & -\frac{\sin i}{v_1}\frac{Di}{Dq}\bigg|_{I} - v_1^{-2}\frac{Dv_1}{Dq}\bigg|_{I}\cos i,
\end{aligned}
\tag{A13}
$$

where (A4) is used to give

$$
\begin{aligned}
\frac{Di}{Dq}\bigg|_{I} &= \frac{\partial i}{\partial q}\bigg|_{I} + \frac{\partial i}{\partial t}\bigg|_{I}\left(\frac{\partial t_0}{\partial q}\right), \\
\frac{Dj}{Dq}\bigg|_{I} &= \frac{\partial j}{\partial q}\bigg|_{I} + \frac{\partial j}{\partial t}\bigg|_{I}\left(\frac{\partial t_0}{\partial q}\right)
\end{aligned}
\tag{A14}
$$

and

$$
\frac{Dv_1}{Dq}\bigg|_{I} = \frac{\partial v_1}{\partial x}\left(\frac{\partial x}{\partial q}\right)_{I} + \frac{\partial v_1}{\partial y}\left(\frac{\partial y}{\partial q}\right)_{I} + \frac{\partial v_1}{\partial z}\left(\frac{\partial z}{\partial q}\right)_{I}. \tag{A15}
$$

The q derivatives in brackets are taken along the interface and so we again use (A4) to get

$$
\begin{aligned}
\left(\frac{\partial x}{\partial q}\right)_{I} &= \frac{\partial x}{\partial q}\bigg|_{t=t_0} + \frac{\partial x}{\partial t}\bigg|_{t=t_0}\left(\frac{\partial t_0}{\partial q}\right) = \frac{\partial x}{\partial q}\bigg|_{t=t_0} + v_1\frac{\partial t_0}{\partial q}\sin i\cos j, \\
\left(\frac{\partial y}{\partial q}\right)_{I} &= \frac{\partial y}{\partial q}\bigg|_{t=t_0} + \frac{\partial y}{\partial t}\bigg|_{t=t_0}\left(\frac{\partial t_0}{\partial q}\right) = \frac{\partial y}{\partial q}\bigg|_{t=t_0} + v_1\frac{\partial t_0}{\partial q}\sin i\sin j, \\
\left(\frac{\partial z}{\partial q}\right)_{I} &= \frac{\partial z}{\partial q}\bigg|_{t=t_0} + \frac{\partial z}{\partial t}\bigg|_{t=t_0}\left(\frac{\partial t_0}{\partial q}\right) = \frac{\partial z}{\partial q}\bigg|_{t=t_0} + v_1\frac{\partial t_0}{\partial q}\cos i.
\end{aligned}
\tag{A16}
$$

All terms in equations (A14)–(A16) that are not already known, may be obtained directly from the ray and geometrical spreading equations (3) and (9) respectively. All three components of the first term in (A11) may therefore be evaluated. The second term in (A11) requires the derivative of $P(t_0)$ and $Q(t_0)$ with respect to q along the interface. We have

$$
\frac{\partial P}{\partial q} = \mathbf{n}\cdot\frac{\partial \mathbf{S}_1}{\partial q}(t_0) + \mathbf{S}_1(t_0)\cdot\frac{\partial \mathbf{n}}{\partial q}. \tag{A17}
$$

Since the derivative of $\mathbf{S}_1$ is the same as the first term in (A11), this is already known, and we require only the derivative on the unit normal, which is found by inserting each component of $\mathbf{n}$ for G in the expression

$$
\frac{\partial G}{\partial q} = \underline{\nabla} G\cdot\frac{\partial \mathbf{x}}{\partial q}\bigg|_{I} \tag{A18}
$$

where the components of the last term are given by (A16). Note the components of $\partial\mathbf{n}/\partial q$ describe the change in curvature of the interface at the point of contact with the ray and are zero for a planar interface. [In the more general case of a curved interface one must know spatial derivatives of the components of $\mathbf{n}$, which are represented by the $\underline{\nabla}\mathbf{G}$ term in (A18).] The derivative of P may now be evaluated from (A17) and the derivative of Q is given by

$$
\frac{\partial Q}{\partial q} = Q^{-1}\left(v_2^{-3}\frac{Dv_2}{Dq}\bigg|_{I} - v_1^{-3}\frac{Dv_1}{Dq}\bigg|_{I} + 2P\frac{\partial P}{\partial q}\right). \tag{A19}
$$

The only new term here is $Dv_2/Dq\,|_{I}$ which is found using (A15) with v_1 replaced by v_2. With the derivatives of P and Q both known, all three components of the second term in (A11) can be evaluated. The third term in (A11) introduces no new terms and so may be evaluated easily. (It is zero for a planar interface.) All terms on the right-hand side of (A11) have now been dealt with and so the complete three-component vector may now be evaluated and the

linear system solved for $\partial i/\partial q\,|_{II}$ and $\partial j/\partial q\,|_{II}$ (q equal to i_0 or j_0).

(b) The reflected ray

The jump conditions for the geometrical spreading variables in the reflection case can be easily obtained from the corresponding conditions in the transmission case. The only modification required to the above is the use of the reflection law equation (6) instead of the transmission law (5). Since (6) is a special case of (5), we need only replace v_2 by v_1 in all of the above expressions and substitute $-Q$ for Q in (A10) and all subsequent equations.