Stanford Exploration Project, Report 72, December 18, 1997, pages 1–254

Stanford Exploration Project, Report 72, December 18, 1997, pages 1–254

Traveltime tomography without ray tracing

Lin Zhang[1]

keywords: *ray tracing, slant stack, traveltime, modeling, inversion, finite difference, algorithm, least squares, amplitudes*

ABSTRACT

The computational tasks involved in the nonlinear tomographic traveltime inversion can be efficiently accomplished by using finite difference methods. The traveltimes can be computed by solving the eikonal equation. The derivatives of traveltimes with respect to the parameters of a slowness model can also be computed by solving a first-order linear partial differential equation. This new approach greatly reduces the computational cost imposed by the conventional ray tracing method. It ensures the proper treatment of the first-arrival traveltimes. Furthermore, the method handles the slowness model expanded with arbitrary basis functions, which is useful in incorporating geological structure interpretation into the inversion process. I expect that the new method will improve the result of the inversion.

INTRODUCTION

Tomographic traveltime inversion is a process of reconstructing slowness models using the traveltimes of seismic events. In recent years, this technique has been successfully used in exploration geophysics for unraveling complex geological structures. Many new methods have been developed to improve the results of the slowness model reconstruction (Paige and Saunders, 1982; Nolet, 1985; Scales, 1987). Traveltime inversion is usually formulated as a nonlinear optimization problem and is solved iteratively by using a gradient method. The largest portion of the computational effort in each iteration of the gradient method is the computation of the traveltimes and the back-projections of the traveltime residuals through ray tracing. Although ray tracing can produce the correct answer, but it is computationally intensive, frequently encounters shadow zones, and sometimes picks raypaths other than the first arrival, which is miserable for the first-arrival traveltime inversion. Using appropriate basis functions for representing the slowness model not only reduces the computational effort but also constrains the inversion process (Harlan, 1989; Van Trier, 1989; Michelena and Harris, 1991). It allows us to incorporate the information from the interpretation

[1]**email:** not available

of geological structures into the inversion process. However, implementing such an algorithm with ray tracing is awkward, for ray trajectories generally have irregular geometries.

The finite difference calculation of traveltimes has been proved to be an accurate and efficient way to compute, on a regular grid, the traveltimes of first-arrival seismic waves through virtually any velocity structure (Vidale, 1988; Van Trier and Symes, 1991; Podvin and Lecomte, 1991). This new approach removes several difficulties imposed by ray tracing. When Vidale first published this method, he claimed that the method promises to aid in tomographic inversion. Van Trier (1990) applied this method in his tomographic inversion after depth migration, and obtain promising results.

In this paper, I present an application of the finite difference method to cross-well tomographic traveltime inversion. I show that we can completely abandon ray tracing and do the required computations with the finite difference method. The new method greatly reduces the computational cost imposed by the conventional ray tracing method. It also ensures the proper treatment of the first arrival traveltimes. Furthermore, the method handles the slowness model expanded with arbitrary basis functions. The first part of this paper reviews general concepts in the tomography of cross-well data and identifies the three major computational tasks that need to be done. Readers who are familiar with this subject can skip this part. The second part explains how to use the finite difference method to compute the traveltimes, gradient vectors, and conjugate gradient vectors that are required in the conjugate gradient algorithm, and derives the corresponding partial differential equations and initial conditions. This paper emphasizes the descriptions of theories. The details of the algorithms can be found in some of my other papers (Zhang, 1991a and 1991b).

NONLINEAR TRAVELTIME TOMOGRAPHY

Tomographic traveltime inversion can be applied to surface reflection data, VSP data, and cross-well data. For the sake of simplicity, I only address the problem for the cross-well geometry. However, the principles and algorithms are equally applicable to other geometries.

Traveltime and slowness model

In a cross-well experiment, the seismic source is located at various positions in one well and receivers are deployed in the other well. The seismic waves generated by the source propagate through the medium within the cross-well region, and are recorded by the receivers. Each pair of source and receiver generates a seismic trace from which we can pick the traveltime of the first arrival (or first break). Therefore, from a cross-well experiment, one can obtain a set of first arrival traveltimes $\{t_i(r);\ i = 1, 2, \cdots, N\}$

that are the functions of receiver depth r. Subscript i indicates the source number, and N is the total number of sources used in the experiment.

The medium between two wells is characterized by a slowness model. The slowness model is usually described as a linear superposition of a set of basis functions:

$$m(x,z) = \sum_{j=1}^{M} m_j \beta_j(x,z), \tag{1}$$

where $\{\beta_j(x,z);\ j = 1,2,\cdots,M\}$ is a set of known basis functions, and $\{m_j;\ j = 1,2,\cdots,M\}$ is a set of unknown coefficients that determines the slowness model. The choices of basis functions should depend on the experimental geometry, prior knowledge of the slowness model, and the resolution and accuracy of the reconstruction. The goal of the traveltime tomographic inversion is to reconstruct the slowness model of the medium between the two wells by using the traveltime information picked from the data.

Formulation of nonlinear inversion

The tomographic inversion is usually done through the optimization process. Given a slowness model, we can simulate the wave propagation between two wells and compute the first arrival traveltime, denoted by $\{\tau_i(r);\ i = 1,2,\cdots,N\}$. We want to find a set of parameters $\{m_j;\ j = 1,2,\cdots,M\}$ that minimizes the differences between the traveltimes picked from the recorded data and the traveltimes computed from the model, subjected to some linear constraints. The mathematical description of this problem is as follows:

$$\min_{\{m_j\}} \left\{ \sum_{i=1}^{N} \int_r \left[\tau_i(\mathbf{m}) - t_i\right]^2 dr + \|\mathbf{B}\mathbf{m} + \mathbf{c}\|^2 \right\}, \tag{2}$$

where $\mathbf{m}^T = (m_1\ m_2\ \cdots\ m_M)$ is the unknown slowness parameter vector. Matrix $\mathbf{B}$ and vector $\mathbf{c}$ are chosen to constrain the inversion process. Because the traveltimes are generally nonlinear functions of the slowness model parameters, equation (2) defines a nonlinear optimization problem. A nonlinear optimization is usually accomplished iteratively. In each iteration, we linearize the traveltime functions as follows:

$$\tau_i(r) \approx \hat{\tau}_i(r) + \sum_{j=1}^{M} \frac{\partial \tau_i}{\partial m_j}(m_j - \hat{m}_j), \tag{3}$$

where $\{\hat{m}_j;\ j = 1,2,\cdots,M\}$ are the slowness model parameters obtained in the previous iteration, and $\{\hat{\tau}_i(r);\ i = 1,2,\cdots,N\}$ are the traveltimes calculated from the previous model. We define $\delta m_j = m_j - \hat{m}_j$ and $\delta t_i(r) = t_i(r) - \hat{\tau}_i(r)$, and then form two vectors as follows:

$$\mathbf{\Delta m}^T = (\,\delta m_1 \quad \delta m_2 \quad \cdots \quad \delta m_M\,),$$

$$\mathbf{\Delta t}^T = (\,\delta t_1(r) \quad \delta t_2(r) \quad \cdots \quad \delta t_{N(r)}\,)\,.$$

We also define $\mathbf{A}$ to be an $N \times M$ matrix with the elements

$$a_{ij}(r) = \frac{\partial \tau_i}{\partial m_j}$$

and define

$$\mathbf{\Delta c} = \mathbf{B}\hat{\mathbf{m}} - \mathbf{c}.$$

Then, equation (2) can be approximated as

$$\min_{\{m_j\}} \left\{ \int_r \|\mathbf{A}\mathbf{\Delta m} - \mathbf{\Delta t}\|^2 dr + \|\mathbf{B}\mathbf{\Delta m} - \mathbf{\Delta c}\|^2 \right\}. \tag{4}$$

Now the problem becomes a linear least squares problem. We can use a gradient method to solve it. If we use the steepest descent method, we need to calculate the gradient vector defined as follows:

$$\mathbf{g} = \mathbf{g}^{(1)} + \mathbf{g}^{(2)} = \int_r \mathbf{A}^T \mathbf{\Delta t} dr + \mathbf{B}^T \mathbf{\Delta c}. \tag{5}$$

This vector has a dimension of M. If we choose the conjugate gradient method, we need to calculate, in addition to the gradient vector, the conjugate gradient vector defined as follows:

$$\mathbf{h}(r) = \begin{pmatrix} \mathbf{h}^{(1)}(r) \\ \mathbf{h}^{(2)}(r) \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} \mathbf{g}, \tag{6}$$

which has a dimension of N plus the number of conditions.

FINITE DIFFERENCE METHODS

In each iteration of the nonlinear optimization process, the traveltimes, gradient vector, and conjugate gradient vector need to be computed. The conventional method does these computations through ray tracing. After finding the ray trajectories, one integrates the slowness function along the rays to get traveltimes, spreads the traveltime residuals along the rays to compute the gradient vector, and integrates the gradient function along rays to compute the conjugate gradient vector. In this section, I describe a method that does these computations with the finite difference technique.

The computation of traveltimes

Initiated by Vidale (1988), several finite difference algorithms for calculating traveltimes have been developed (Van Trier and Symes, 1990; Podvin and Lecomte, 1991). These algorithms solve the eikonal equation as follows:

$$\left(\frac{\partial \tau_i}{\partial x}\right)^2 + \left(\frac{\partial \tau_i}{\partial z}\right)^2 = m^2(x,z), \tag{7}$$

where subscript i indicates the source number. The initial condition is that the traveltime function $\tau = \tau_i(x, z)$ is equal to zero at the source location. These finite difference algorithms can efficiently compute the traveltimes on an even grid of (x, z). The traveltime $\tau_i(r)$ can be found by evaluating function $\tau_i(x, z)$ at each receiver location.

Using the advantages of the existing algorithms, I have developed an adaptive finite difference scheme that is always stable and fully vectorizable (Zhang, 1991a). Further, I interpreted the finite difference method as local wavefront propagations through local ray tracing, and proposed a more accurate method that I call the local paraxial ray method (Zhang, 1991b).

Computation of matrix A

The elements of matrix $\mathbf{A}$ are the derivatives of the traveltimes with respect to the slowness model parameters. Let us define

$$p_{ij}(x, z) = \frac{\partial \tau_i}{\partial m_j}. \tag{8}$$

Taking the derivatives of both sides of the eikonal equation with respect to m_j gives

$$\frac{\partial \tau_i}{\partial x}\frac{\partial p_j}{\partial x} + \frac{\partial \tau_i}{\partial z}\frac{\partial p_j}{\partial z} = m(x, z)\beta(x, z). \tag{9}$$

If we assume that $\tau_i(x, z)$ has been computed by solving the eikonal equation, then equation (9) is a first-order linear partial differential equation (PDE). Because the traveltime at the source location is always zero no matter how the slowness model changes, equation (9) should be solved with the initial condition that $p_{ij}(x, z) = 0$ at the source location. Evaluating the function $p_{ij}(x, z)$ at receiver locations gives the elements of matrix $\mathbf{A}$.

Computation of the gradient vector

The gradient vector defined in equation (5) is the sum of traveltime residual term $\mathbf{g}^{(1)}$ and condition term $\mathbf{g}^{(2)}$. The computation of the condition term is straightforward because matrix $\mathbf{B}$ is given. In principle, we can compute the traveltime residual term by first computing matrix $\mathbf{A}$, and then performing the matrix multiplication. However, matrix $\mathbf{A}$ has $N \times M$ elements that are functions of r; hence, both its calculation and storage are of high cost. If we can compute the vector $\mathbf{g}^{(1)}$ directly using a finite difference method, we can considerably reduce the cost in computation time and storage.

If we solve the first-order linear PDE (9) using the method of characteristics, we find that

$$a_{ij}(r) = \int_0^{s_i(r)} \beta_j(\hat{x}_i(\xi, r), \hat{z}_i(\xi, r))d\xi, \tag{10}$$

where $\hat{x}_i(s,r)$ and $\hat{z}_i(s,r)$ are the coordinates of the ray that starts from the number i source and ends at the receiver at depth r, and s is the arc-length of the ray. Clearly, the elements of matrix $\mathbf{A}$ are the integrations of the basis functions along rays. The component of the gradient vector can be computed as follows:

$$
\begin{aligned}
g_j^{(1)} &= \int_r \sum_{i=1}^{N} a_{ij}(r)\delta t_i(r)dr \\
&= \sum_{i=1}^{N} \int_r \int_0^{s_i(r)} \beta_j(\hat{x}_i(\xi,r),\hat{z}_i(\xi,r))\delta t_i(r)d\xi dr \qquad (11)\\
&= \sum_{i=1}^{N} \int_x \int_z \beta_j(x,z)\delta t_i(\hat{r}_i(x,z))Jdxdz,
\end{aligned}
$$

where J is the Jacobian of the coordinate transformation:

$$
J = \begin{vmatrix} \frac{\partial s}{\partial x} & \frac{\partial s}{\partial z} \\ \frac{\partial r}{\partial x} & \frac{\partial r}{\partial z} \end{vmatrix} = \sqrt{\left(\frac{\partial r}{\partial x}\right)^2 + \left(\frac{\partial r}{\partial z}\right)^2}. \qquad (12)
$$

The last step of equation (11) is an area integration that is easy to implement. Appendix A shows that function $r = \hat{r}_i(x,z)$ satisfies the first-order linear PDE, as follows:

$$
\frac{\partial \tau_i}{\partial x}\frac{\partial \hat{r}_i}{\partial x} + \frac{\partial \tau_i}{\partial z}\frac{\partial \hat{r}_i}{\partial z} = 0, \qquad (13)
$$

with the initial condition that function $\hat{r}_i(x,z)$ is equal to the receiver depth at each receiver location. We can use a finite difference algorithm, similar to the one used for traveltime calculation, to solve equation (13). The algorithm should start from receiver locations, and extrapolate function $\hat{r}_i(x,z)$ in the opposite directions of the wave propagations, towards the source.

Computation of the conjugate gradient vector

The conjugate gradient vector has two parts. The first part $\mathbf{h}^{(1)}(r)$ is related to the traveltime residuals, and the second part $\mathbf{h}^{(2)}(r)$ is related to the constraints. The computation of $\mathbf{h}^{(2)}(r)$ is simple because matrix $\mathbf{B}$ is known. We want to compute

$\mathbf{h}^{(1)}(r)$ with a finite difference method. The components of $\mathbf{h}^{(1)}(r)$ are as follows:

$$\begin{aligned} h_i^{(1)}(r) &= \sum_{j=1}^{M} a_{ij}(r) g_j \\ &= \sum_{j=1}^{M} \int_0^{s_i(r)} \beta_j(\hat{x}_i(\xi,r), \hat{z}_i(\xi,r)) g_j d\xi \qquad (14) \\ &= \int_0^{s_i(r)} \sum_{j=1}^{M} g_j \beta_j(\hat{x}_i(\xi,r), \hat{z}_i(\xi,r)) d\xi. \end{aligned}$$

It is shown in Appendix A that if we solve the first-order linear PDE

$$\frac{\partial \tau_i}{\partial x}\frac{\partial q_i}{\partial x} + \frac{\partial \tau_i}{\partial z}\frac{\partial q_i}{\partial z} = m(x,z) \sum_{j=1}^{M} g_j \beta_j(x,z) \qquad (15)$$

with the initial condition that $q_i(x,z)$ is equal to zero at the source location, then $h_i^{(1)}(r)$ is equal to $q_i(x,z)$ at the receiver locations. Equation (15) has a form similar to the eikonal equation; hence we can use the algorithm for the traveltime calculation, after being slightly modified, to solve it.

CONCLUSIONS

This paper presents the theoretical aspects of a new tomographic traveltime inversion algorithm that does not require ray tracing. I show that the major computational tasks that are involved in the nonlinear optimization process and that are conventionally done through ray tracing can be efficiently done by solving the first-order partial differential equations with finite difference methods. The new algorithm allows us to use arbitrary basis functions for expanding the slowness model, and ensures that the traveltimes of the first arrivals are properly used in the inversion process. Although the algorithm is described in two dimensions, it can be generalized to three dimensions without much difficulty. Because we can also compute the amplitudes of the first arrivals with the finite difference method (Vidale and Houston, 1990; Zhang, 1991a), we cau use the same approach to invert the absorption field of a medium.

REFERENCES

Podvin, P., and Lecomte, I., 1991, Finite difference computation of traveltimes in very contrasted velocity models: a massively parallel approach and its associated tools: Geophys. J. Int., **105**, 271–284.

Harlan, W.S., 1989, Tomographic estimation of seismic velocities from reflected raypaths: Presented at the 59th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 922–924.

Michelena, R.J., and Harris, J.M., 1991, Tomographic traveltime inversion using natural pixels: Geophysics, **56**, 635–644.

Nolet, G., 1985, Solving or resolving inadequate and noisy tomographic systems: J. Comp. Physics, **61**, 463–482.

Paige, C.C., and Saunders, M.A., 1982, LSQR: An algorithm for sparse linear equations and sparse least squares: ACM Trans. Math. Software, **8**, 43–71.

Scales, J.A., 1987, Tomographic inversion via the conjugate gradient method: Geophysics, **52**, 179–185.

Van Trier, J., 1988, Migration velocity analysis using geological constraints: Presented at the 58th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 897–900.

Van Trier, J., and Symes, W.W., 1991, Upwind finite-difference calculation of traveltimes: Geophysics, **56**, 812–821.

Vidale, J.E., 1988, Finite-difference calculation of travel times: Bull. Seism. Soc. Am., **78**, no. 6, 2062–2076.

Vidale, J.E., and Houston, H., 1990, Rapid calculation of seismic amplitudes: Geophysics, **55**, 1504–1507.

Zhang, L., 1991a, Finite difference calculation of Green's functions: SEP–**72**, 143–159.

Zhang, L., 1991b, Local paraxial ray method: SEP–**72**, 161–166.

APPENDIX A

In this appendix, I derive equations (13) and (15), and the initial conditions associated with them.

The Cartesian coordinates of the trajectory of a ray are functions of so-called ray coordinates (s, r), as follows:

$$\left\{ \begin{array}{lcl} x & = & \hat{x}_i(s, r) \\ z & = & \hat{z}_i(s, r), \end{array} \right. \qquad (A.1)$$

where s measures the arc-length of the ray and r is the depth at which the ray reaches the receiver well. Subscript i indicates the source number. If we solve equation (A.1) for r, we obtain a function $r = \hat{r}_i(x, z)$ that has a constant value along each ray. Consequently, the gradient directions of function $\hat{r}_i(x, z)$ are orthogonal to rays. Because the gradient directions of traveltime function $\tau_i(x, z)$ are tangential to rays, we have the relation $\nabla\tau_i \cdot \nabla\hat{r}_i = 0$, which, in two dimensions, yields

$$\frac{\partial \tau_i}{\partial x}\frac{\partial \hat{r}_i}{\partial x} + \frac{\partial \tau_i}{\partial z}\frac{\partial \hat{r}_i}{\partial z} = 0. \tag{A.2}$$

At any receiver location, function $\hat{r}_i(x, z)$ is equal to the depth of the receiver, which acts as an initial condition required in solving equation (A.2).

To derive equation (15), we first define

$$q_i(x, z) = \int_0^{s_i(x,z)} \sum_{j=1}^{M} g_j \beta_j(\hat{x}_i(\xi, r), \hat{z}_i(\xi, r)) d\xi. \tag{A.3}$$

At the source location, $s_i(x, z)$ is equal to zero; hence $q_i(x, z)$ is equal to zero. At the receiver locations, $s_i(x, z) = s_i(r)$. Comparing equation (A.3) with equation (15), we can conclude that $q_i(x, z)$ is equal to $h^{(1)}(r)$ at the receiver locations. Now, what remains is to show that $q_i(x, z)$ defined in equation (A.3) satisfies the first-order linear partial differential equation (15). To do so, we take the derivative of both sides of equation (A.3) with respect to s, which yields

$$\frac{\partial q_i}{\partial x}\frac{dx}{ds} + \frac{\partial q_i}{\partial z}\frac{dz}{ds} = \sum_{j=1}^{M} g_j \beta_j(\hat{x}_i(s, r), \hat{z}_i(s, r)). \tag{A.4}$$

Functions $x = \hat{x}_i(s, r)$ and $z = \hat{z}_i(s, r)$ define the Cartesian coordinates of the ray. They satisfy the ray equation as follows:

$$\left\{ \begin{array}{rcl} \dfrac{dx}{ds} & = & \dfrac{1}{m(x,z)}\dfrac{\partial \tau_i}{\partial x} \\[2ex] \dfrac{dz}{ds} & = & \dfrac{1}{m(x,z)}\dfrac{\partial \tau_i}{\partial z} \end{array} \right. \qquad \left\{ \begin{array}{rcl} \dfrac{d}{ds}\left[\dfrac{\partial \tau_i}{\partial x}\right] & = & m_x(x, z) \\[2ex] \dfrac{d}{ds}\left[\dfrac{\partial \tau_i}{\partial z}\right] & = & m_z(x, z), \end{array} \right. \tag{A.5}$$

where subscripts x and z denote partial derivatives with respect to x and z, respectively. Substituting these functions into equation (A.4) gives

$$\frac{\partial \tau_i}{\partial x}\frac{\partial q_i}{\partial x} + \frac{\partial \tau_i}{\partial z}\frac{\partial q_i}{\partial z} = m(x, z) \sum_{j=1}^{M} g_j \beta_j(x, z), \tag{A.6}$$

which is equation (15).