COMM. MATH. SCI.
Vol. 4, No. 1, pp. 249–266

AN ADJOINT STATE METHOD FOR THREE-DIMENSIONAL TRANSMISSION TRAVELTIME TOMOGRAPHY USING FIRST-ARRIVALS*

SHINGYU LEUNG[†] AND JIANLIANG QIAN[‡]

Abstract. Traditional transmission travel-time tomography hinges on ray tracing techniques. We propose a PDE-based Eulerian approach to travel-time tomography so that we can avoid using the cumbersome ray-tracing technique. We start from the eikonal equation, define a mismatching functional and derive the gradient of the nonlinear functional by an adjoint state method. The resulting forward and adjoint problems can be efficiently solved by using the fast sweeping method; a limited memory BFGS method is used to drive the mismatching functional to zero with quadratic convergence. 2-D and 3-D numerical results as well as Marmousi synthetic velocity model demonstrate the robustness and the accuracy of the method.

Key words. Eikonal equations, traveltime tomography, fast sweeping, ray tracing, adjoint state methods.

AMS subject classifications. 65K10, 86A22, 35R30

1. Introduction

Estimation of wave-speed distribution from acoustic, seismic or electromagnetic first-arrival travel-time data is the goal of transmission travel-time tomography. In seismics velocity analysis is often an important step in prospect evaluation in areas where lithology and structure undergo significant lateral change. In this work we propose a new, robust and efficient tomography method which is aimed at such applications.

All the traditional methods of travel-time tomography are directly based on Fermat's least travel-time principle and bear a close link to the X-ray computerized tomography (CT) used in medical diagnosis. In medical CT the measured data are assumed to be modeled by line integrals of wave amplitude attenuation for straight ray-paths passing through the body, and the Radon transform provides the foundation for medical CT. However, in seismics the ray-path curvature has to be taken into account in that lithology and structure usually have strong inhomogeneity, and the resulting ray-paths can depend strongly on the unknown wave speeds. To achieve such a purpose, ray-tracing based travel-time tomography methods require very complicated data structure to trace curved rays through each pixel [4]; see [25] for 3-D examples. In addition, such ray-tracing based methods inevitably produce irregular ray coverage of the computational domain, and the resulting system of equations may not be well-conditioned [1, 2, 3]. In this paper we propose a PDE-based Eulerian approach to travel-time tomography so that we can avoid using the cumbersome ray-tracing technique.

Recall that a necessary condition for Fermat's least travel-time principle to hold is characterized by the eikonal equation for travel-time [11], and the viscosity solution for the eikonal equation with a point-source condition is the least travel-time from

*Received: October 25, 2005; accepted (in revised version): January 30, 2006. Communicated by Shi Jin.

[†]Department of Mathematics, UCLA, Los Angeles, CA 90095-1555, USA (syleung@math.ucla.edu).

[‡]Department of Mathematics and Statistics, Wichita State University, Wichita, KS 67260-0033, USA (qian@math.wichita.edu).

the source to an arbitrary point connected by a shortest ray-path, as observed by [13, 16]. Because of its continuous dependence on the wave-speed distribution and source locations, the viscosity solution can be computed by various numerical schemes stably. In this work, we model travel-times from a single source to multiple receivers by using the eikonal equation and propose a fast sweeping based adjoint state method for transmission travel-time tomography. The new approach not only overcomes some shortcomings inherited in the traditional ray-tracing based travel-time tomography but also enjoys quadratic convergence, thus it is very fast and robust.

However, first-arrival based transmission travel-time tomography usually has very limited resolution. Since the output from travel-time tomography is mainly used for building a macro velocity model in seismic velocity analysis, it is important to have very fast, efficient tomography tools, even though we may have to use only first-arrivals. On the other hand, since multi-valued travel-times and resulting multipathings are common in complex velocity structures, it is necessary to take into account all the arrivals systematically. As is well known, ray-tracing methods can yield all arrivals, and the works presented in [8, 7] have used multivalued travel-times from ray tracing methods, but those works are on reflection travel-time tomography which is different from transmission tomography in that rays start at the surface, reflect off interfaces whose depths are to be determined, and return to the surface. To use all arrivals in transmission tomography in an Eulerian framework, we proposed to formulate transmission tomography by using the Liouville equation based PDE framework in phase space [15, 14].

In this work, we will only concentrate on first-arrival based traveltime tomography. We start from a mismatching functional between measured and simulated data and drive the functional to zero by a well designed limited memory BFGS (L-BFGS) optimization method. Although our approach shares some similarities with that in Sei-Symes [21, 22], that work was based on the paraxial formulation of the eikonal equation and only illustrated the feasibility of computing the travel-time gradient by using the adjoint state method. Instead of the paraxial formulation of the eikonal equation, we derive the gradient of the mismatching functional directly from the steady eikonal equation by using the adjoint state method; furthermore, we apply the fast sweeping method [26, 24, 12] to solve the eikonal equation directly (the forward problem) and design a new fast sweeping method to solve the adjoint equation of the linearized eikonal equation (the adjoint problem) so that the required gradient can be computed efficiently; finally a limited memory BFGS optimization method drives the mismatching functional to zero with quadratic convergence.

In Section 2, we first explain the variational method for inverting the velocity model using measurements on the boundary of a specified computational domain. To minimize the energy in the variational formulation, we derive the gradient of the nonlinear functional. To efficiently compute the gradient direction, in Section 3 we apply the fast sweeping method to the eikonal equation and design a new fast sweeping method for the adjoint equation of the linearized eikonal equation. Sections 4, 5 and 6 show various numerical examples to demonstrate the feasibility and the robustness of the new formulation. Section 7 will then conclude the paper.

2. Governing equations

We start from the eikonal equation with a point source condition in an isotropic medium which occupies an open, bounded rectangular domain $\Omega_p \subset \mathcal{R}^3$. By isotropy here we mean the wave velocity has no directional dependence. The equation is as follows,

$$|\nabla T| = \frac{1}{c} \tag{2.1}$$

with the point source condition

$$T(\mathbf{x}_s) = 0, \tag{2.2}$$

where $T(\mathbf{x})$ is the travel-time of the wave from the source $\mathbf{x}_s$ to the point $\mathbf{x}$, and $c \in C^1(\Omega_p)$ is a positive velocity function.

For a given velocity model c, the viscosity solution of this equation can be computed efficiently by fast sweeping methods, and such solutions correspond to the least travel-time or the first-arrival travel-time according to [16].

In this work we are interested in the related inverse problem, the so-called transmission travel-time tomography problem: given both the first-arrival travel-time measurements on the boundary $\partial\Omega_p$ and the location of the point source $\mathbf{x}_s \in \Omega_p$, invert for the velocity field $c(\mathbf{x})$ inside the domain Ω_p.

To achieve this we propose to invert for the velocity model by minimizing the following mismatching functional (energy),

$$E(c) = \frac{1}{2} \int_{\partial\Omega_p} |T - T^*|^2, \tag{2.3}$$

where $T^*|_{\partial\Omega_p}$ is the measurement and $T|_{\partial\Omega_p}$ is computed by solving (2.1) with a point source condition (2.2). In other words, this energy measures the L^2-difference between the experimental measurement, T^*, and the solution from the eikonal equation, T, on the boundary of the computational domain.

To minimize this energy, we use the method of gradient descent. We first perturb the velocity field c by $\epsilon\tilde{c}$, which causes a corresponding change in T by $\epsilon\tilde{T}$. The change in the energy is then given by

$$\delta E = \epsilon \int_{\partial\Omega_p} \tilde{T}(T - T^*) + O(\epsilon^2). \tag{2.4}$$

From the state equation (2.1), the perturbations in c and T are related by

$$T_x \tilde{T}_x + T_y \tilde{T}_y + T_z \tilde{T}_z = -\frac{\tilde{c}}{c^3}. \tag{2.5}$$

We need to determine the perturbation in c, $\tilde{c}$, so as to decrease the energy $E(c)$. The main difficulty is that the perturbation in E, δE, depends on $\tilde{c}$ implicitly through $\tilde{T}$ and the partial differential equation (2.5). To efficiently compute $\tilde{c}$ which minimizes E, we apply the adjoint state method.

Multiplying (2.5) by $\epsilon\lambda$, integrating it over Ω_p, applying integration by parts, and adding the resulting expression to (2.4), we have

$$\begin{aligned} \frac{\delta E}{\epsilon} = &\int_{\partial\Omega_p} \tilde{T}(T - T^*) + \int_y \int_z \lambda T_x \tilde{T}|_{x_{\min}}^{x_{\max}} + \int_x \int_z \lambda T_y \tilde{T}|_{y_{\min}}^{y_{\max}} + \int_x \int_y \lambda T_z \tilde{T}|_{z_{\min}}^{z_{\max}} \\ &- \int_{\Omega_p} \tilde{T}[(\lambda T_x)_x + (\lambda T_y)_y + (\lambda T_z)_z] + \int_{\Omega_p} \frac{\tilde{c}\lambda}{c^3} + O(\epsilon). \end{aligned} \tag{2.6}$$

Next, we choose λ satisfying

$$[(-T_x)\lambda]_x + [(-T_y)\lambda]_y + [(-T_z)\lambda]_z = 0, \tag{2.7}$$

with the boundary condition,

$$(\mathbf{n}\cdot\nabla T)\lambda = T^* - T, \tag{2.8}$$

on the boundary $\partial\Omega_p$, where $\mathbf{n}$ is the unit outward normal of the boundary. By introducing this adjoint state equation, one can eliminate the dependence of $\tilde{T}$ when determining the gradient of E with respect to c.

Ignoring the higher than linear order terms in the energy perturbation we have

$$\frac{\delta E}{\epsilon} = \int_{\Omega_p} \frac{\tilde{c}\lambda}{c^3}. \tag{2.9}$$

To minimize the energy using the method of gradient descent one could choose the perturbation $\tilde{c} = -\lambda/c^3$. This implies

$$\delta E = -\epsilon \int_{\Omega_p} \tilde{c}^2 \leq 0 \tag{2.10}$$

and the equality holds when $||\tilde{c}||_{H^0(\Omega_p)} = 0$. However, it is not straight-forward how one can guarantee the following two properties,

1. $\tilde{c}^k|_{\partial\Omega_p} = 0$;
2. $c^{k+1} = c^k + \epsilon\tilde{c}^k$ smooth.

The first condition assumes that we can measure c on the boundary $\partial\Omega_p$, denoted by $c^*|_{\partial\Omega_p}$, which is a reasonable assumption. This means that the variations of the velocity function on the boundary should be zero.

The second condition is a regularity condition on c^k. This regularity seems to be too restrictive in practice. In general, one only needs $c^k \in C^1$ to guarantee well-posedness of the state equation (2.1). However, assuming that one uses $\tilde{c}^k = -\lambda/c^3$ directly, it is not clear whether this function would give us the desired regularity. Even if this perturbation is in C^1, the numerical solution may have jumps or spikes. These irregularities will force one to pick a very small step-size, ϵ^k, in the minimization process. Therefore, to have faster convergence, we impose the above regularity in each iteration.

One way to satisfy the above two properties is to use the descent direction

$$\tilde{c} = -(I - \nu\Delta)^{-1}\left(\frac{\lambda}{c^3}\right), \tag{2.11}$$

where I is the identity operator, Δ is the Laplacian operator and $\nu \geq 0$ controls the amount of regularity that one wants. The homogeneous boundary condition is imposed in inverting the operator $(I - \nu\Delta)$. With this particular $\tilde{c}$, we have

$$\delta E = -\epsilon \int_{\Omega_p} (\tilde{c}^2 + \nu|\nabla\tilde{c}|^2) \leq 0. \tag{2.12}$$

We notice that this process amounts to seeking updates in some weighted Sobolev space in the case $\nu > 0$. Then the above equality holds when $\|\tilde{c}\|_{H^1_\nu(\Omega_p)} = 0$.

In the above calculation, we use the first-arrivals at different receivers associated with a single point source. If we perform multiple such experiments, namely, we have many such data sets, then those can be easily incorporated into the formulation. For example, we can assume that there are N point sources located at $\mathbf{x}_s^i$, for $i=1,\cdots,N$, and N sets of first-arrival travel-time measurements T_i^* associated with these N sources are available. Then we can simply define a new energy

$$E^N(c)=\frac{1}{2}\sum_{i=1}^{N}\int_{\partial\Omega_p}|T_i-T_i^*|^2, \tag{2.13}$$

where T_i is the solution from the eikonal equation with the corresponding point source condition $T(\mathbf{x}_s^i)=0$. Utilizing the same approach as above, we have the following perturbation in the energy

$$\frac{\delta E^N}{\epsilon}=\int_{\Omega_p}\frac{\tilde{c}}{c^3}\sum_{i=1}^{N}\lambda_i, \tag{2.14}$$

where λ_i is the adjoint variable of T_i satisfying

$$\{[-(T_i)_x]\lambda_i\}_x+\{[-(T_i)_y]\lambda_i\}_y+\{[-(T_i)_z]\lambda_i\}_z=0, \tag{2.15}$$

with the boundary condition,

$$(\mathbf{n}\cdot\nabla T_i)\lambda_i=T_i^*-T_i, \tag{2.16}$$

for $i=1,\cdots,N$.

Consequently, we can choose the following gradient direction to minimize the energy $E^N(c)$,

$$\tilde{c}=-(I-\nu\Delta)^{-1}\left(\frac{1}{c^3}\sum_{i=1}^{N}\lambda_i\right). \tag{2.17}$$

We remark that the above updating procedure is similar to the so-called simultaneous iterative reconstruction technique frequently used in medical imaging; it is also possible to adopt the algebraic reconstruction type technique as used in [9] to update the velocity.

3. Algorithm and numerical implementations

3.1. Tomography algorithm.

Here we give an algorithm for this tomography problem.

Tomography Algorithm:

1. Initialize c^k for $k=0$ by solving
$$(I-\nu\Delta)c^0=0, \tag{3.1}$$
with the boundary condition $c^0|_{\partial\Omega}=c_{\text{exact}}|_{\partial\Omega}$.
2. Obtain $T(x,z)$ by solving (2.1) with the point source condition (2.2) using $c=c^k$.
3. Obtain $\lambda(x,z)$ by solving (2.7) with the boundary condition (2.8).
4. Obtain $\tilde{c}^k$ using (2.11).
5. Determine ϵ^k using, for example, the Armijo-Goldstein rule or simply $\epsilon^k=\epsilon$.
6. Update
$$c^{k+1}=c^k+\epsilon^k\tilde{c}^k.$$

7. Go back to Step 2 until $||\tilde{c}^k(x,z)||_2 \leq \delta$ or $k \geq k_{\max}$, where δ and $k_{\max}$ are given convergence parameters.

To start the iteration, we need to initialize c^0. Here in the algorithm we assume that we can measure the velocity at receivers, giving $c^0|_{\partial\Omega} = c_{\text{exact}}|_{\partial\Omega}$. This condition can be replaced by other assumptions. In practice, due to the nonlinearity of the problem, different initial guesses will generally lead to different energy minimizers. This non-uniqueness can be overcome by some a priori knowledge of the model. For example, the above assumption can be relaxed by replacing the Dirichlet conditions on both the left and right boundaries with the Neumann boundary conditions $\partial c^0/\partial x|_{x=x_{\min}} = \partial c^0/\partial x|_{x=x_{\max}} = 0$.

The convergence could be sped up by replacing the gradient descent method with BFGS-type iterations. To solve the elliptic equation in Step 4, we use the FFT. In Step 2 and Step 3, both the equations (2.1) and (2.7) can be solved by the Fast Sweeping Method [26, 24, 12], which we detail next.

3.2. Fast sweeping method for equation (2.1).

The fast sweeping method was originated in Boue and Dupis [5]. Its first PDE formulation was for implicit and non-parametric shape reconstruction from unorganized points using a variational level set method [27]; Zhao [26] proved the $O(N)$ convergence of the method for the eikonal equation based on the Godunov Hamiltonian on Cartesian meshes; later on, the fast sweeping method was extended to treat Hamilton-Jacobi equations with convex Hamiltonians based on the Godunov Hamiltonian [24] and handle Hamilton-Jacobi equations with non-convex Hamiltonians based on the Lax-Friedrichs Hamiltonian [12]; see [24, 12] and references therein for the fast sweeping method on Cartesian meshes and [20] for the method on triangulated meshes. Certainly, one may also use other methods such as the fast marching method [23].

To be self-contained, we give a short summary of the fast sweeping method for eikonal equations. To avoid cluttered notations we present the algorithm for the 2-D case only; see [26] for more details.

First we discretize the rectangular domain $\Omega \subset \mathcal{R}^2$ into a uniform mesh with mesh points $\mathbf{x}_{i,j}$ and mesh sizes $\Delta x = \Delta z = h$, and we denote the numerical solution at $\mathbf{x}_{i,j}$ by $T_{i,j}$. Applying the Godunov numerical Hamiltonian to the eikonal equation, for $i = 2,\cdots,I-1$, $j = 2,\cdots,J-1$, we have

$$[(T_{i,j} - T_{xmin})^+]^2 + [(T_{i,j} - T_{zmin})^+]^2 = \frac{h^2}{c_{i,j}^2}, \tag{3.2}$$

where

$$T_{xmin} = \min(T_{i-1,j}, T_{i+1,j}),\ T_{zmin} = \min(T_{i,j-1}, T_{i,j+1}),$$

and $(x)^+$ denotes the positive part of x. At the boundary of the computational domain one sided difference is used.

Fast Sweeping Algorithm:

1. Initialize the point source condition $T(\mathbf{x}_s) = 0$ by assigning the exact value if $\mathbf{x}_s$ is a mesh point, or assigning to grid points near $\mathbf{x}_s$ exact values which are computed by using the constant velocity at the point source. These values are fixed in later iterations. Assign larger positive values at all other grid points, and these values will be updated later.
2. Update the solution by Gauss-Seidel iterations with alternating sweeping. At each grid point $\mathbf{x}_{i,j}$ whose value was not fixed during the initialization, compute

the candidate solution, denoted by $\bar{T}$ of (3.2) from the current values of its neighbors $T_{i\pm1,j}$, $T_{i,j\pm1}$ and then update $T_{i,j}$ to be the smaller one between $\bar{T}$ and its current value; i.e., $T_{i,j}^{new}=\min(T_{i,j}^{old},\bar{T})$. We sweep the whole domain with four alternate ordering repeatedly: $i=1:I, j=1:J$; $i=1:I, j=J:1$; $i=I:1, j=1:J$; $i=I:1, j=J:1$. Here i and j are the running indices along x and y directions.

3. Test the convergence: given convergence criterion $\epsilon>0$, check whether $\|T^{n+1}-T^n\|_{L^1}\leq\epsilon$.

We remark that the sweeping strategy can be used for more general Hamilton-Jacobi equations as long as an efficient local solver is available at each grid point, so that an iterative procedure is well defined at each local grid point. On the other hand, we may apply the sweeping strategy to solve equation (2.7), which reduces to a symmetrical Gauss-Seidel-type iterative method.

3.3. Fast sweeping method for equation (2.7). Next we design a fast sweeping method for equation (2.7). Once again to simplify the notation, we give a 2-D formulation only; the extension to a 3-D formulation is straightforward.

The adjoint state equation (2.7) can be written in the following form

$$(a\lambda)_x+(b\lambda)_z=0, \tag{3.3}$$

where a and b are given functions of (x,z).

Considering a computational cell centered at (x_i,z_j) and discretizing the equation in conservation form, we have

$$\begin{aligned}&\frac{1}{\Delta x}\left(a_{i+1/2,j}\lambda_{i+1/2,j}-a_{i-1/2,j}\lambda_{i-1/2,j}\right)\\&+\frac{1}{\Delta z}\left(b_{i,j+1/2}\lambda_{i,j+1/2}-b_{i,j-1/2}\lambda_{i,j-1/2}\right)=0.\end{aligned} \tag{3.4}$$

The values of λ on the interfaces, $\lambda_{i\pm1/2,j}$ and $\lambda_{i,j\pm1/2}$, are determined according to the propagation of characteristics. In the case when $a_{i+1/2,j}>0$, the characteristic for determining λ goes from the left hand side of the interface to the right hand side, and this suggests that we use the value $\lambda_{i,j}$ to define $\lambda_{i+1/2,j}$; otherwise, we have $\lambda_{i+1/2,j}=\lambda_{i+1,j}$. The terms $\lambda_{i,j\pm1/2}$ can be defined in a similar way.

Introducing the following notations

$$\begin{aligned}a^{\pm}_{i+1/2,j}&=\frac{a_{i+1/2,j}\pm|a_{i+1/2,j}|}{2}, & a^{\pm}_{i-1/2,j}&=\frac{a_{i-1/2,j}\pm|a_{i-1/2,j}|}{2},\\ b^{\pm}_{i,j+1/2}&=\frac{b_{i,j+1/2}\pm|b_{i,j+1/2}|}{2} \text{ and } & b^{\pm}_{i,j-1/2}&=\frac{b_{i,j-1/2}\pm|b_{i,j-1/2}|}{2},\end{aligned}$$

we have

$$\begin{aligned}&\frac{1}{\Delta x}\Big((a^{+}_{i+1/2,j}\lambda_{i,j}+a^{-}_{i+1/2,j}\lambda_{i+1,j})-(a^{+}_{i-1/2,j}\lambda_{i-1,j}+a^{-}_{i-1/2,j}\lambda_{i,j})\Big)+\\&\frac{1}{\Delta z}\Big((b^{+}_{i,j+1/2}\lambda_{i,j}-b^{-}_{i,j+1/2}\lambda_{i,j+1})-(b^{+}_{i,j+1/2}\lambda_{i,j-1}-b^{-}_{i,j+1/2}\lambda_{i,j})\Big)=0,\end{aligned} \tag{3.5}$$

which can be rewritten as

$$\left(\frac{a^+_{i+1/2,j}-a^-_{i-1/2,j}}{\Delta x}+\frac{b^+_{i,j+1/2}-b^-_{i,j-1/2}}{\Delta z}\right)\lambda_{i,j}=\frac{a^+_{i-1/2,j}\lambda_{i-1,j}-a^-_{i+1/2,j}\lambda_{i+1,j}}{\Delta x}$$
$$+\frac{b^+_{i,j-1/2}\lambda_{i,j-1}-b^-_{i,j+1/2}\lambda_{i,j+1}}{\Delta z}. \tag{3.6}$$

This gives an expression to build up a fast sweeping-type iterative method.

To apply this iterative scheme to equation (2.7), we need to specify the function values of a and b not at the cell centers (x_i,z_j), but on the cell interfaces $(x_{i\pm1/2},z_j)$ and $(x_i,z_{j\pm1/2})$. This can be done easily using central differences. For example, we have $a_{i+1/2,j}=-(T_{i+1,j}-T_{i,j})/\Delta x$ and $a_{i-1/2,j}=-(T_{i,j}-T_{i-1,j})/\Delta x$. In addition, we have to incorporate the boundary condition (2.8) into the above linear system for λ as well. Then we can show that the coefficient matrix of the resulting linear system for λ is irreducibly diagonally dominant; therefore the alternating symmetrical Gauss-Seidel iteration converges.

Fast Sweeping Algorithm for equations (2.7) and (2.8):

1. On the boundary, compute $(\mathbf{n}\cdot\nabla T)$ from the solution of the eikonal solver using one side difference. Next, compute the boundary condition for λ according to (8). These values will be fixed in the following computations.
2. Update $\lambda_{i,j}$ at the interior points according to (22). As in the Fast Sweeping method for (1), we sweep the whole domain with four alternate orderings.
3. For some given convergence criterion $\epsilon>0$, repeat 2 until $||\lambda^{n+1}-\lambda^n||_{L^1}\le\epsilon$.

We point out that the above fast sweeping method is different from the fast marching method used in [10], in that our method is iterative and theirs is constructive based on upwinding properties.

3.4. L-BFGS method. In the Tomography Algorithm, we update the approximation to the velocity by a typical gradient descent method, where

$$c^{k+1}=c^k-\epsilon^k\tilde{c}^k. \tag{3.7}$$

Although it is simple to implement, the method is not efficient because it takes a large number of iterations to converge to a steady state solution.

To speed up the convergence, we can apply the quasi-Newton method defined by

$$c^{k+1}=c^k+\epsilon^k s^k, \tag{3.8}$$

where $s^k=-A_k^{-1}E'(c^k)$ and A_k is a positive definite operator satisfying the secant condition

$$A_{k+1}(c^{k+1}-c^k)=E'(c^{k+1})-E'(c^k). \tag{3.9}$$

In this iteration, the operator A_{k+1} is updated by modifying the previous operator A_k.

One possible way to modify this operator is defined by the Broydon-Fletcher-Goldfarb-Shanno (BFGS) procedure,

$$A_k v=A_{k-1}v+\alpha<p,v>p+\beta<q,v>q, \tag{3.10}$$

where

$$p = y/||y||, q = A_{k-1}s/||A_{k-1}s||,$$
$$\alpha = ||y||^2/<y,s>, \beta = -||A_{k-1}s||^2/<s,A_{k-1}s> \tag{3.11}$$

with $s = c^k - c^{k-1}$, $y = E'(c^k) - E'(c^{k-1})$ and $A_0 = I$.

However, in practice, the condition number of A_k can be increased significantly throughout the iteration, which makes the computation inaccurate. To alleviate this, one can modify the iteration using the limited memory BFGS (L-BFGS) given by

$$A_k v = v + \sum_{j=k-L+1}^{k} (\alpha_j <p_j,v> p_j + \beta_j <q_j,v> q_j). \tag{3.12}$$

In this paper, we adapt the L-BFGS-B code from [6]. This code requires the user to provide only subroutines to compute both the energy to be minimized and the gradient of this energy. The step size ϵ^k is automatically determined.

4. Two-dimensional numerical examples

In the following examples, we use 129×129 grid points in the x-z space. Using the above formulation, we need measurements, denoted by ϕ^* and T^*, on the boundary $\partial\Omega_p$. If the point source is located inside Ω_p, the characteristics of the eikonal equation always flow out of the domain. Therefore, in synthetic experiment the boundary measurements can be obtained by solving the equation (2.1) directly using the Fast Sweeping Method together with the exact velocity c.

For each velocity model below, we have implemented the following two cases - one source and ten sources. For the one source case, we use the boundary measurements from the only point source located at $(x,z) = (0,0.1)$. In the cases with ten point sources, we use nine more sets of boundary measurements, and these correspond to source locations at $(x,z) = (\pm 0.25, 0.1), (\pm 0.5, 0.1), (0, 1.9), (\pm 0.25, 1.9)$ and $(\pm 0.5, 1.9)$, respectively. However, to save some space we only present the results corresponding to the case of ten sources.

To start the algorithm, we initialize the velocity c^0 by solving the above elliptic equation (3.1) with $\nu = 1$.

4.1. Example 1. Constant model.

The exact velocity model is given by $c \equiv 1$.

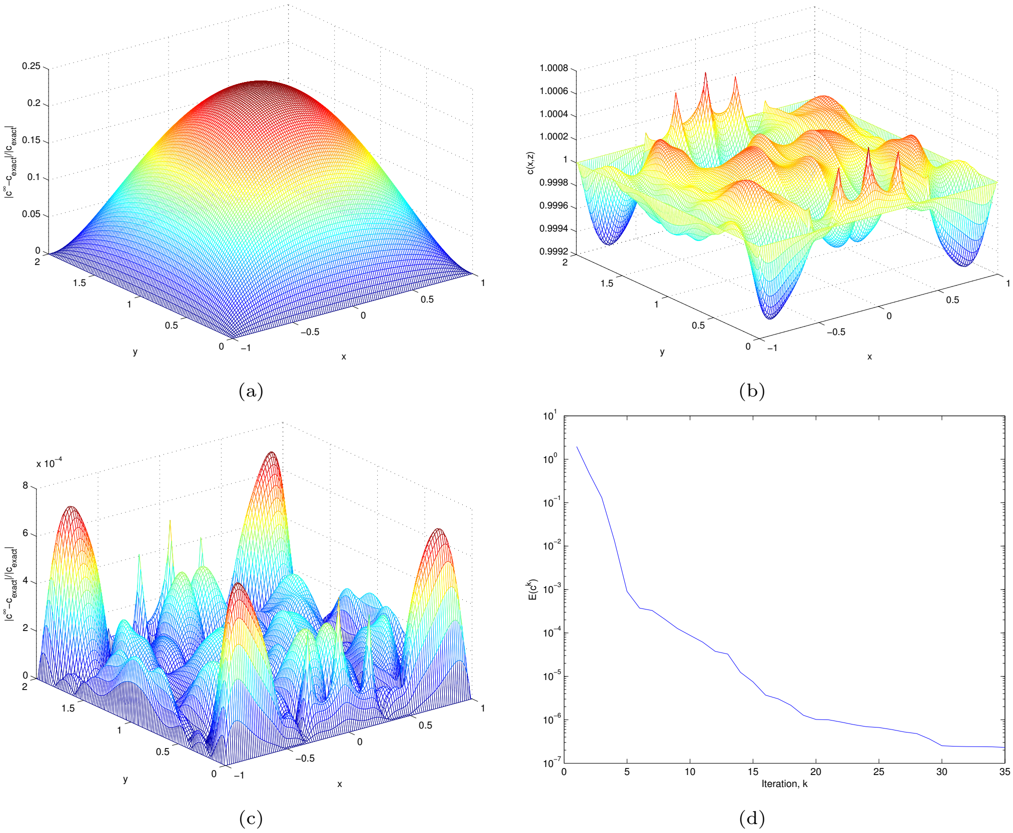
We use the BFGS method to invert for the velocity. The results are shown in Figure 4.1. As we can see, the recovered velocity is almost exact, the relative error is almost negligible, and we observe the typical quadratic convergence of the algorithm due to the L-BFGS method.

4.2. Example 2. Waveguide model.

The exact velocity model is given by

$$c(x,z) = 3 - 2.5\exp\left(-\frac{x^2}{2}\right). \tag{4.1}$$

We apply the BFGS method to invert for the velocity. Figure 4.2 shows the relative error at convergence and the convergence history. In Figure 4.3, we show slices of the solution along $z = 1$ and $x = 0$. As we can see, we are able to recover the velocity very well.



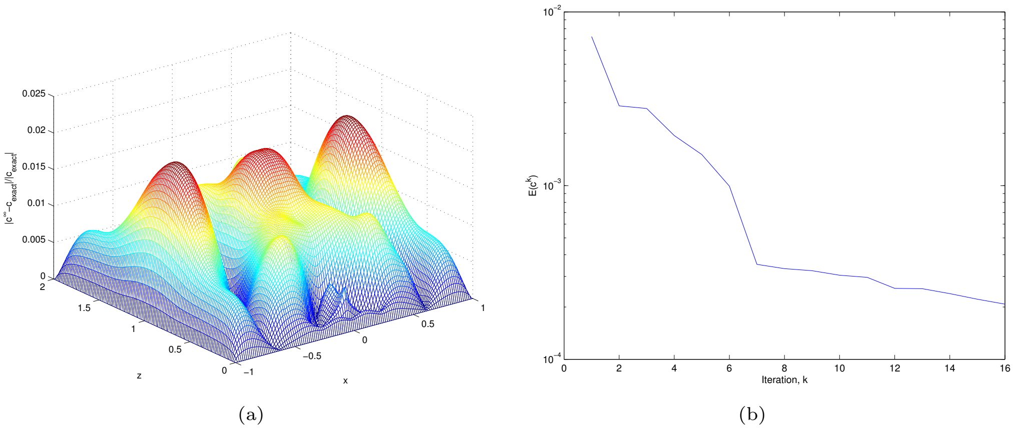
(a)

(b)

(c)

(d)

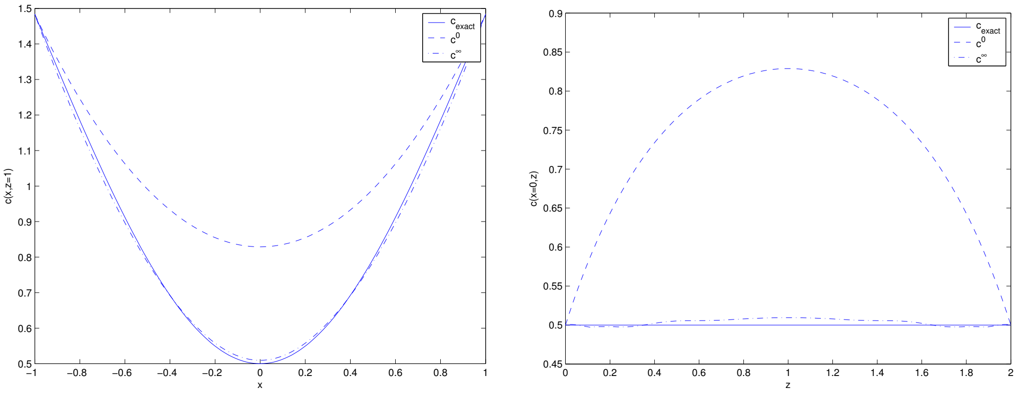
FIG. 4.1. *(Example 1. Ten Sources) BFGS. (a): the initial guess; (b): the final approximated c; (c): the relative error in the solution; (d): the convergence history of energy.*



(a)

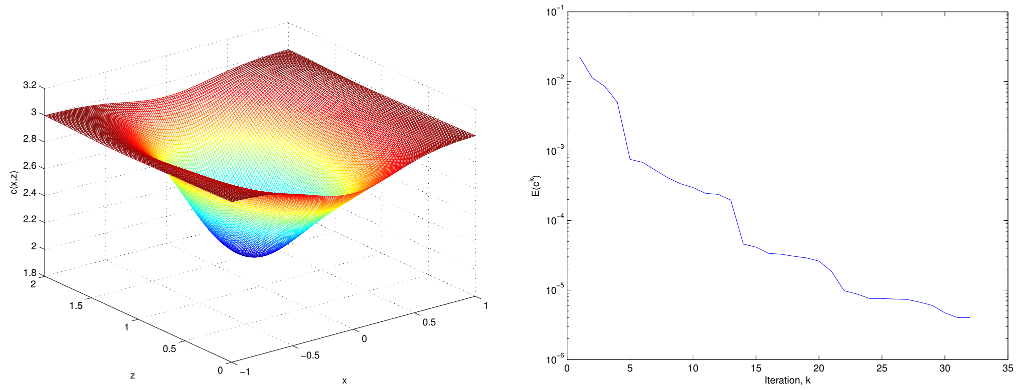
(b)

FIG. 4.2. *(Example 2. Ten Sources) (a): the relative error in the solution and (b): the convergence history of energy.*



(a) (b)

FIG. 4.3. *(Example 2. Ten Sources) Cross-sections of the solutions. (a): $z=1$ and (b): $x=0$.*



(a) (b)

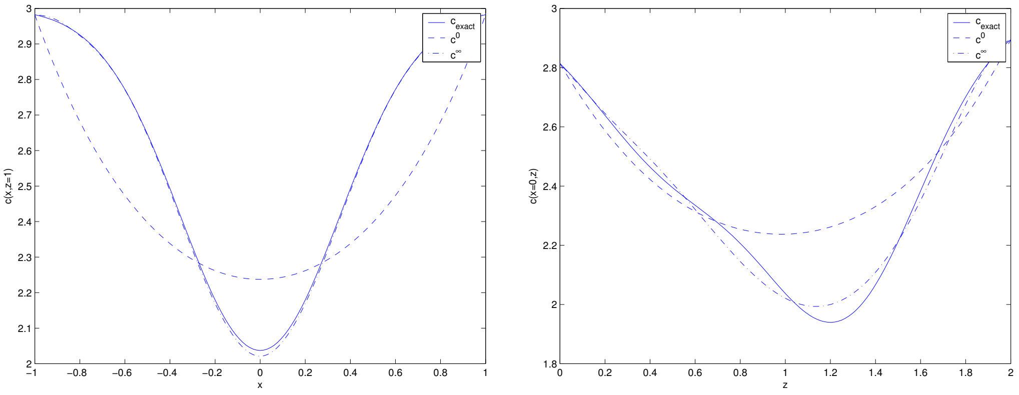
FIG. 4.4. *(Example 3. Ten Sources) BFGS. (a): the final approximated c and (b): the convergence history of energy.*

4.3. Example 3. Gaussian Model. The exact velocity model is given by

$$c(x,z)=3-\frac{1}{2}\exp\left(-\frac{x^2+(z-0.5)^2}{0.5^2}\right)-\exp\left(-\frac{x^2+(z-1.25)^2}{0.5^2}\right). \tag{4.2}$$

Next we apply the BFGS method to invert for the velocity. In Figure 4.4, we show the convergent velocity and the convergence history of the algorithm; once again, we observe quadratic convergence. In Figure 4.5, we show slices of the final converged velocity; as we can see, they fit well with the exact velocity.

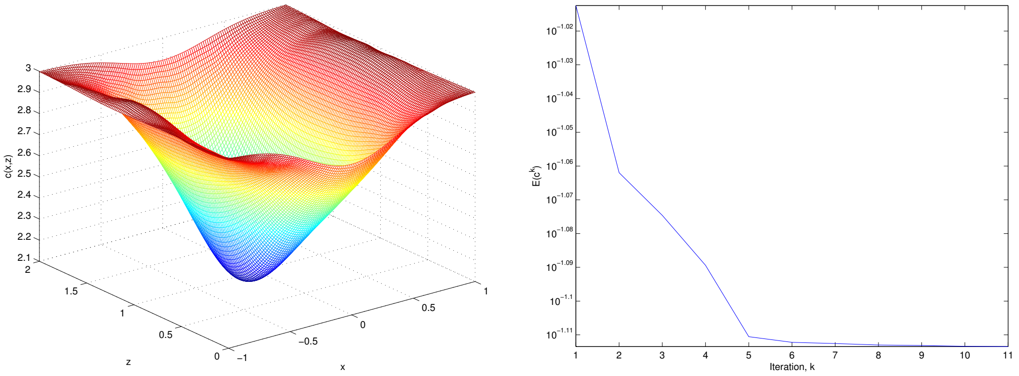
To further test the algorithm, we repeat the experiment but perturb the synthetic data T^* with some noise. Using the same velocity model, we first compute the traveltime on the boundary of the domain. These measurements are added 5% Gaussian noise with zero mean. Figures 4.6 and 4.7 show that we have robust convergence as well. As shown in Figure 4.6(b), we are not able to drive the energy to zero. This is



(a)

(b)

FIG. 4.5. *(Example 3. Ten Sources) BFGS. Cross-sections of the solutions. (a): $z=1$ and (b): $x=0$.*



(a)

(b)

FIG. 4.6. *(Example 3 with added noise. Ten Sources) BFGS. (a): the final approximated c; (b): the convergence history of energy.*

expected because the boundary measurements are highly oscillatory, and in general we cannot find a smooth velocity c which produces exactly the same travel-times as those noisy data.

5. Three-dimensional numerical examples

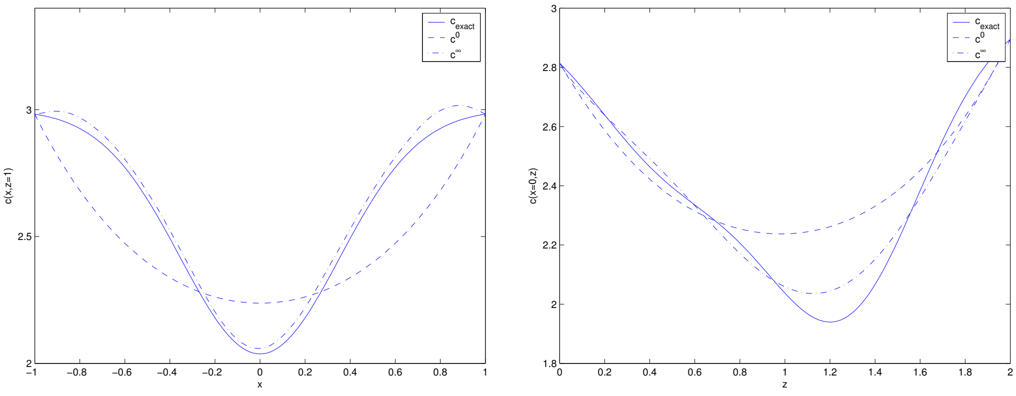
In the following examples, we use $65\times 65\times 65$ grid points in the three-dimensional space. Using the above formulation, we need measurements, denoted by T^*, on the boundary $\partial\Omega_p$.

For each velocity model shown below, we have implemented the following case; 49 sources on the levels $z=0.1$ and $z=1.9$, and we have 98 sets of measurements in total.

To start the algorithm, we initialize the velocity c^0 by solving the elliptic equation (3.1) with $\nu=1$.

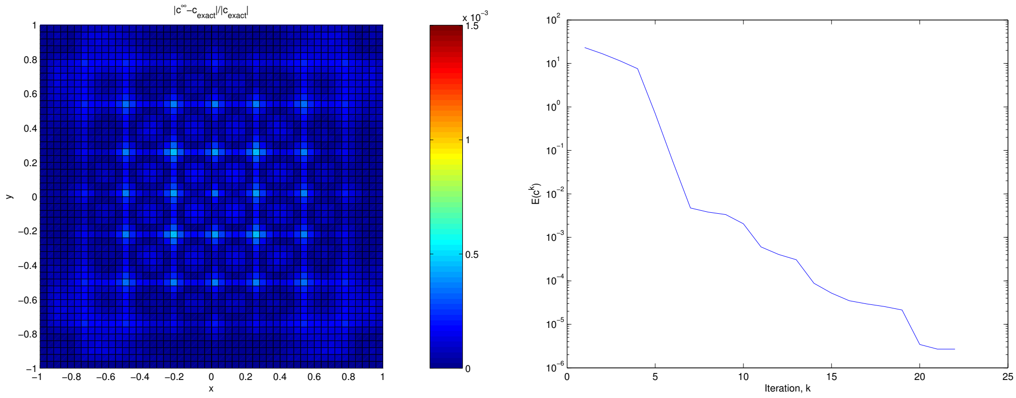
5.1. Example 1. Constant model.

The exact velocity model is given by $c\equiv 1$.



(a) (b)

FIG. 4.7. *(Example 3 with added noise. Ten Sources) BFGS. Cross-sections of the solutions: (a): $z=1$ and (b): $x=0$.*



(a) (b)

FIG. 5.1. *(Example 1. 98 Sources) 3-D case. (a): the relative error in the solution on the cross-section $z=1$ and (b): the convergence history of energy.*

We use the BFGS method to invert for the velocity. The results are shown in Figure 5.1; we observe the quadratic convergence once again.

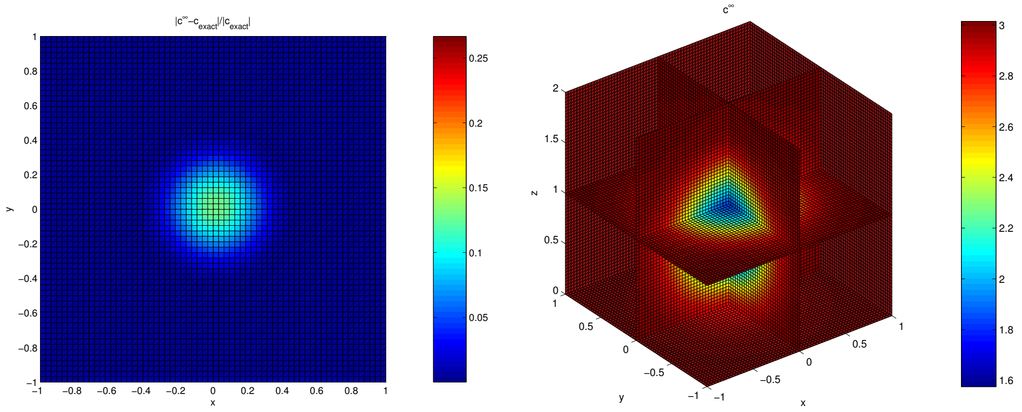
5.2. Example 2. Gaussian model. The exact velocity model is given by

$$c(x,y,z)=3-\frac{1}{2}\exp\left(-\frac{x^2+y^2+(z-0.5)^2}{0.5^2}\right)-\exp\left(-\frac{x^2+y^2+(z-1.25)^2}{0.5^2}\right). \quad (5.1)$$

We use the gradient descent method to invert for the velocity. The results are shown in Figure 5.2.

6. Synthetic Marmousi model

The Marmousi model from the 1996 INRIA Workshop on Multi-arrival Traveltimes is a synthetic model which will challenge the adjoint state method used here.

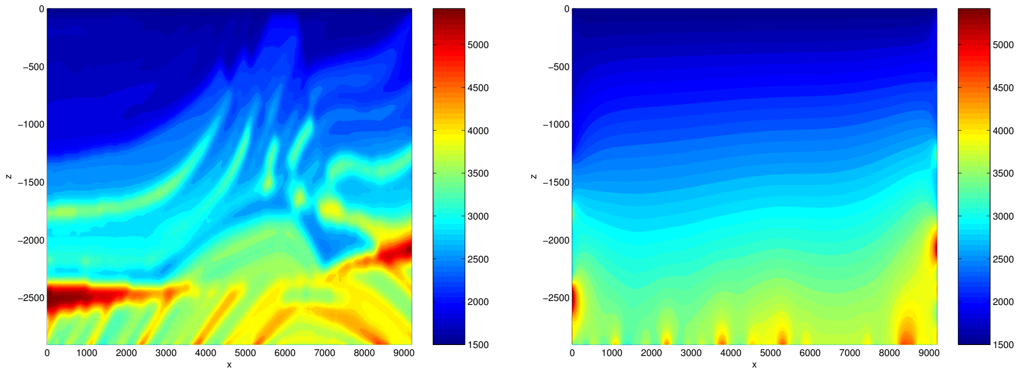


(a) (b)

FIG. 5.2. *(Example 2. 98 Sources) 3-D case. (a): the relative error in the solution on the cross-section $z=1$ and (b): the final approximated c.*

The original Marmousi model is sampled on a 24m by 24m grid, consisting of 384 samples in the x-direction and 122 samples in the z-direction; therefore the model dimension is 9.192km long in the x-direction and 2.904km deep in the z-direction.

In the computational results presented here, we use 20 sources and their (x,z)-coordinates are (200, 2800), (1000:1000:9000, 2800), (200, 100) and (1000:1000:9000, 100), respectively, where we have used by now the standard Matlab colon notation.

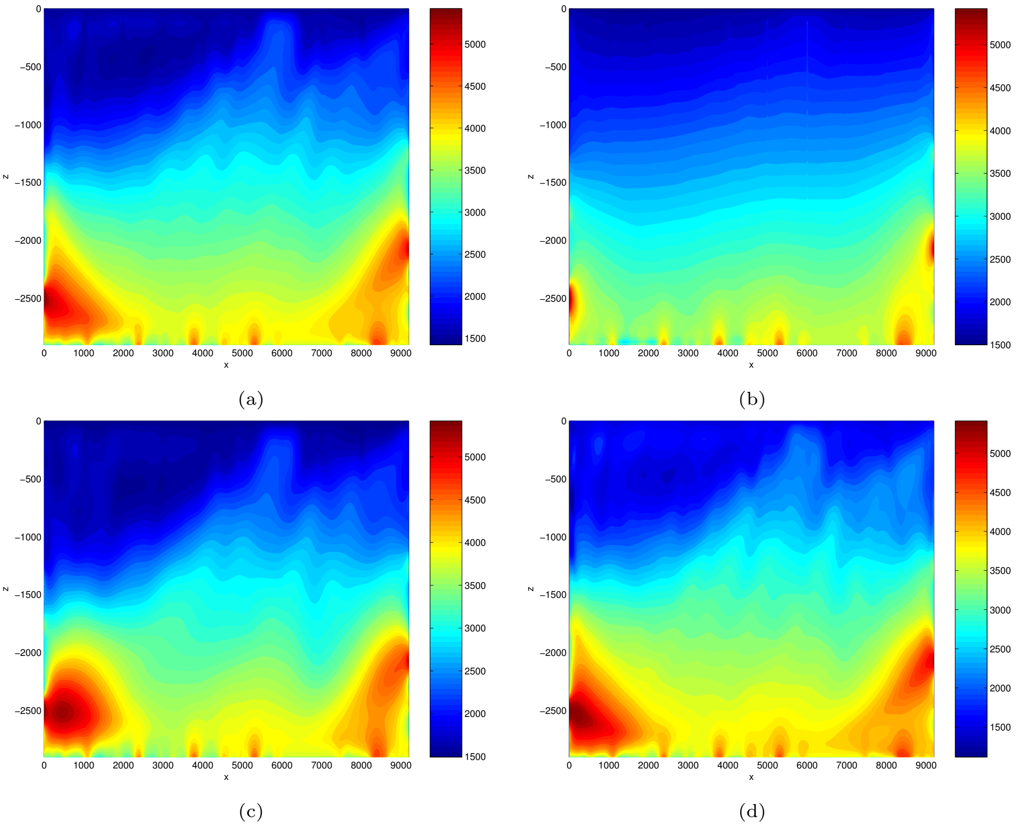


(a) (b)

FIG. 6.1. *(Marmousi model) (a): the true velocity distribution and (b): the initial profile c^0.*

The true Marmousi velocity model is illustrated in Figure 6.1(a). As we can see, this velocity model has high contrast with variations of different scales. On the one hand, since the fast sweeping method used here is unconditionally stable, the forward eikonal solver will not have difficulty in computing traveltime to first-order accuracy. On the other hand, viscosity-solution based first-arrival traveltimes will not be able to give us too much information about variations of small scales occurring in the velocity model; to retain the information related to small scales, we have to use multiple arrivals, which in turn call for multiple-arrival based traveltime tomography. In this regard, for computing multiple arrivals of the Marmousi model in the Eulerian framework, see [17, 18] for more.

To start the algorithm, we initialize the velocity c^0 by solving the Laplace equation $-\Delta c^0 = 0$ with $c^0|_{\partial\Omega} = c_{\text{exact}}|_{\partial\Omega}$. The solution is plotted in Figure 6.1(b).



(a)

(b)

(c)

(d)

FIG. 6.2. *(Marmousi model) Converged solutions. (a): $\nu = 10^4$ and $\Delta x = 24$; (b): $\nu = 10^2$ and $\Delta x = 24$; (c): $\nu = 10^6$ and $\Delta x = 24$; (d): $\nu = 10^4$ and $\Delta x = 12$.*

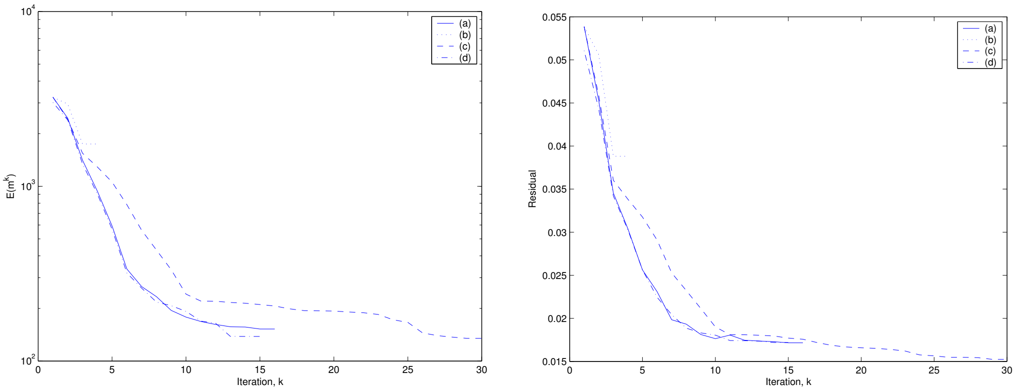
We use the BFGS method to invert for the velocity. Figure 6.2 presents the inversion results for different cases in terms of the sampling size Δx and the parameter ν.

Comparing Figure 6.2(a) with the true model, Figure 6.1(a), we have succeeded in imaging the macro scale variations of the velocity model and we were not able to image finer scale variations of the velocity model which does exist in the true velocity model. However, transmission tomography usually has very limited resolution, and we believe that this result is near optimal using the current approach.

To confirm this, we refine the velocity model by doubling the number of grid points in each direction while keeping the regularization parameter ν fixed; the corresponding solution is shown in Figure 6.2(d). We also check the following residual in the solution defined by

$$R = \frac{\Sigma_{i=1}^{N} \int_{\partial\Omega} |T_i - T_i^*|/T_i^* ds}{N \int_{\partial\Omega} ds} \tag{6.1}$$

where N is the number of sources defined above. This quantity essentially is the average relative error in the first-arrival time per source per receiver. If the above residual is not changing too much, we will accept the inversion result since there is not much model misfit left to drive improvement. Indeed, as shown in Figure 6.3, even if we refine the velocity model, the residuals are almost the same after 15 BFGS steps. In fact, the solutions from the coarse and fine resolution are similar, as shown in Figures 6.2(a) and 6.2(d).



(I) (II)

FIG. 6.3. *(Marmousi model) The change in (I): the energy and (II): the residual. Legend: (a): $\nu = 10^4$ and $\Delta x = 24$; (b): $\nu = 10^2$ and $\Delta x = 24$; (c): $\nu = 10^6$ and $\Delta x = 24$; (d): $\nu = 10^4$ and $\Delta x = 12$.*

Using a relative small $\nu = 10^2$, the BFGS iteration has difficulty in converging to a smooth solution. This is clearly seen in Figure 6.2(b). The BFGS iteration stops at the fourth iteration with $E(m^4) \simeq 1700$, where $m = \log c$; see Figure 6.3. This difficulty comes from the sharp spikes in the solution near the source locations, where the traveltime field is not differentiable [19]. These sudden changes will degrade accuracy in the computed gradient and make it hard for BFGS to search for a descent direction.

Increasing the magnitude of ν (from 10^4 to 10^6), on the other hand, we have better convergent results. As seen in the energy plot, the energy which uses the larger ν (the dashed line) reaches a lower state than that using $\nu = 10^4$ (the solid line). Theoretically we penalize the gradient of $\tilde{c}$ so that it is small in a weighted Sobolev space as illustrated in equation (2.12).

Concerning the speed, the computational time for the cases with $\nu = 10^4$ using $\Delta x = 24$ and 12 are 53 minutes and 387 minutes, respectively. We also list in Table 1 the number of iterations required to solve both the eikonal equation and the adjoint equation for each given velocity field. These numbers are obtained for the case $\nu = 10^4$ with only one point source located at $(5000, -2800)$ in the first BFGS iteration. The first row shows the number of iterations with $\Delta x = 24$, while the second row corresponds to the case with $\Delta x = 12$.

Δx	Eikonal equation	Adjoint equation
24	20 (6.68×10^{-10})	17 (1.62×10^{-9})
12	28 (9.60×10^{-11})	25 $(3.38 \times ^{-8})$

TABLE 6.1. *Iteration count for the fast sweeping methods. The numbers in the brackets are the errors in the corresponding iteration,* $||T^{n+1} - T^n||$ *or* $||\lambda^{n+1} - \lambda^n||$.

7. Conclusion

We have proposed a PDE-based Eulerian approach to traveltime tomography so that we can avoid using the cumbersome ray-tracing technique in inversion. We started from the eikonal equation, defined a mismatching functional and derived the gradient of the nonlinear functional by an adjoint state method. The resulting forward and adjoint problems can be efficiently solved by using the fast sweeping method. In addition, we have used a limited memory BFGS method to drive the functional to zero with quadratic convergence. Numerical results for 2-D, 3-D, and Marmousi synthetic velocity models demonstrated the robustness and the accuracy of the method.

The methodology proposed here is quite general and can be extended to many other situations without any major difficulty. For example, instead of point sources we can easily modify the formulation to accommodate plane waves as the source condition, which can be achieved by using the boundary condition resulting from the plane wave condition in equation (2.2). If the domain to be imaged is irregular or non-rectangular, then we can use the fast sweeping method designed in [20] to solve the eikonal equation efficiently; although we cannot directly apply the standard FFT technique when regularizing the gradient direction (2.11), we may still use other fast solvers like multi-grid methods to solve the Possion equation. To further improve the resolution of inverted velocity models, one can also incorporate the amplitude information into the formulation; this generalization is the so-called diffraction tomography, which consists of an ongoing project.

Acknowledgement. The authors thank Professors S. Osher, W. W. Symes and H.-K Zhao for discussions and encouragement when carrying out this project. This work was finished when the second author was visiting the Center for Scientific Computation and Mathematical Modeling (CSCAMM), University of Maryland, College Park, and he is grateful to the Director, Eitan Tadmor, and staff of CSCAMM for providing excellent working environment. Leung is supported by ONR Grant N00014-02-1-0720. Qian is supported by NSF DMS-0542174.

REFERENCES

[1] J. Berryman, *Stable iterative reconstruction algorithm for nonlinear traveltime tomography*, Inverse Problems, 6, 21-42, 1990.

[2] J. Berryman, *Analysis of approximate inverses in tomography I. resolution analysis of common inverses*, Optimization and Engineering, 1, 87-115, 2000.

[3] J. Berryman,*Analysis of approximate inverses in tomography II. iterative inverses*, Optimization and Engineering, 1, 437-473, 2000.

[4] T. N. Bishop, K. P. Bube, R. T. Cutler, R. T. Langan, P. L. Love, J. R. Resnick, R. T. Shuey, D. A. Spindler and H. W. Wyld, *Tomographic determination of velocity and depth in laterally varying media*, Geophysics, 50, 903-923, 1985.

[5] M. Boue and P. Dupuis, *Markov chain approximations for deterministic control problems with affine dynamics and quadratic costs in the control*, SIAM J. Numer. Anal., 36, 667-695, 1999.

[6] R. H. Byrd, P. Lu, J. Nocedal and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM J. Sci. Comp., 16, 1190-1208, 1995.

[7] R. A. Clarke, B. Alazard, L. Pelle, D. Sinuquet, P. Lailly, F. Delprat-Jannaud and L. Jannaud, *3D traveltime reflection tomography with multi-valued arrivals*, 71st Ann. Internat. Mtg., Soc. Expl. Geophys., 1601-1604, 2001.

[8] F. Delprat-Jannaud and P. Lailly, *Reflection tomography: how to handle multiple arrivals?*, J. Geophys. Res., 100, 703-715, 1995.

[9] O. Dorn, H. Bertete-Aguirre, J. G. Berryman and G. C. Papanicolaou, *A nonlinear inversion method for 3D electromagnetic imaging using adjoint fields*, Inverse Problems, 15, 1523-1558, 1999.

[10] S. Fomel and J. Sethian, *Fast phase space computation of multiple traveltimes*, Proc. Nat. Aca. Sci., 99, 7329-7334, 2002.

[11] I. M. Gelfand and S. V. Fomin, *Calculus of Variation*, Prentice-Hall, 1963.

[12] C. Y. Kao, S. J. Osher and J. Qian, *Lax-Friedrichs sweeping schemes for static Hamilton-Jacobi equations*, J. Comp. Phys., 196, 367-391, 2004.

[13] S. N. Kruzkov, *Generalized solutions of nonlinear first order equations with several independent variables. II.* Math. USSR-Sb., 1, 93-116, 1967.

[14] S. Leung and J. Qian, *A transmission tomography problem based on multiple arrivals from paraxial Liouville equations*, 75th Ann. Internat. Mtg., Soc. Expl. Geophys., 2589-2592, 2005.

[15] S. Leung and J. Qian, *Transmission traveltime tomography based on paraxial Liouville equations and level set formulations*, preprint, 2005.

[16] P. L. Lions, *Generalized solutions of Hamilton-Jacobi equations*, Pitman Advanced Publishing Program, 1982.

[17] J. Qian and S. Leung, *A level set method for paraxial multivalued traveltimes*, J. Comp. Phys., 197, 711-736, 2004.

[18] J. Qian and S. Leung, *A local level set method for paraxial multivalued geometric optics*, SIAM J. Sci. Comp., to appear., 2005.

[19] J. Qian and W. W. Symes, *Adaptive finite difference method for traveltime and amplitude*, Geophysics, 67, 167-176, 2002.

[20] J. Qian, Y-T Zhang and H-K. Zhao, *Fast sweeping methods for eikonal equations on triangulated meshes*, SIAM J. Numer. Anal., Submitted, 2005.

[21] A. Sei and W. W. Symes, *Gradient calculation of the traveltime cost function without ray tracing*, 65th Ann. Internat. Mtg., Soc. Expl. Geophys., 1351-1354, 1994.

[22] A. Sei and W. W. Symes, *Convergent finite-difference traveltime gradient for tomography*, 66th Ann. Internat. Mtg., Soc. Expl. Geophys., 1258-1261, 1995.

[23] J. A. Sethian, *Level Set Methods*, Cambridge Univ. Press, second edition, 1999.

[24] R. Tsai, L.-T. Cheng, S. J. Osher and H. K. Zhao, *Fast sweeping method for a class of Hamilton-Jacobi equations*, SIAM J. Numer. Anal., 41, 673-694, 2003.

[25] J. K. Washbourne, J. W. Rector and K. P. Bube, *Crosswell traveltime tomography in three dimensions*, Geophysics, 67, 853-871, 2002.

[26] H. K. Zhao, *Fast sweeping method for eikonal equations*, Math. Comp., 74, 603-627, 2005.

[27] H. K. Zhao, S. Osher, B. Merriman and M. Kang, *Implicit and non-parametric shape reconstruction from unorganized points using variational level set method*, Computer Vision and Image Understanding, 80, 295-319, 2000.