UCRL-JC-107927
PREPRINT

GLOBAL EXTREMA IN TRAVELTIME TOMOGRAPHY

James G. Berryman

This paper was prepared for the
Proceedings of the Third IMACS Symposium
on Computational Acoustics
Cambridge, MA
June 26-28, 1991

July 1991

Lawrence
Livermore
National
Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

UCRL-JC--107927

DE91 017666

Global Extrema in Traveltime Tomography

James G. Berryman
University of California
Lawrence Livermore National Laboratory
P. O. Box 808 L-202
Livermore, CA 94550

Global Extrema in Traveltime Tomography

James G. Berryman
University of California
Lawrence Livermore National Laboratory
P. O. Box 808 L-202
Livermore, CA 94550

Abstract

In acoustics of inhomogeneous media, Fermat's principle of least time has often been used in bending methods of ray tracing to determine ray paths for purposes of forward modeling. Fermat's principle plays an equally important role in traveltime inversion, *i.e.*, when attempting to estimate the wave velocities in a medium through which the wave has traveled in a measured time from known source to receiver. Since measured first-arrival traveltimes are necessarily the minimum traveltimes through the medium whose sound velocity profile is to be reconstructed, Fermat's principle allows us to assign all possible wave-speed profiles to one of two classes: a model is either infeasible or feasible depending on whether or not there are any paths from source to receiver that have less traveltime than that measured. The feasible set is convex and furthermore an exact solution to the inversion problem (if any) must lie on the boundary between the feasible and infeasible sets. Thus, Fermat's principle permits the convexification of the nonlinear traveltime inversion problem, so the only extrema are global extrema.

1 Traveltime Tomography

A typical problem in seismic traveltime tomography is to infer the (isotropic) compressional-wave slowness (reciprocal of velocity) distribution of an inhomogeneous medium, given a set of observed first-arrival traveltimes between sources and receivers of known location within the medium. This problem is common for borehole-to-borehole seismic tomography in oil field applications [Justice et al., 1989]. We also consider the problem of inverting for wave slowness when the absolute traveltimes are not known, as is normally the case in earthquake seismology.

1.1 Slowness Models

We consider three kinds of slowness models. Sometimes we allow the slowness to be a general function of position, $s(\mathbf{x})$. However, we often make one of two more restrictive assumptions that (i) the model comprises homogeneous blocks, or cells, with s_j then denoting the slowness value of the jth cell, or (ii) the model is composed of a grid with values of slowness assigned at the grid points with some interpolation scheme to assign the values between grid points. Of course, we can think of blocks of constant slowness as a special case of continuous models, or continuous models as a limiting case of blocks as the blocks become infinitesimal.

When it is not important which type of slowness model is involved, we will refer to the model abstractly as a vector $\mathbf{s}$ in a vector space $\mathcal{S}$. For a block model with n blocks we have $\mathcal{S} = R^n$, the n-dimensional Euclidean vector space. A continuous slowness model is an element of a function space, e.g., $\mathcal{S} = C(R^3)$, the set of continuous functions of three real variables.

1.2 Fermat's Principle and Traveltime Functionals

The traveltime of a seismic wave is the integral of slowness along a ray path connecting the source and receiver. To make this more precise, we will define two functionals for traveltime.

Let P denote an arbitrary path connecting a given source and receiver in a slowness model $\mathbf{s}$. We will refer to P as a ***trial ray path***. We define a functional τ^P which yields the traveltime along P. Letting $\mathbf{s}$ be the continuous slowness distribution $s(\mathbf{x})$, we have

$$\tau^P(\mathbf{s}) = \int_P s(\mathbf{x})\, dl^P, \tag{1}$$

where dl^P denotes the infinitesimal distance along the path P.

Fermat's principle states that the correct ray path between two points is the one of least overall traveltime, *i.e.*, it minimizes $\tau^P(\mathbf{s})$ with respect to P. [Actually, Fermat's principle is the weaker condition that the traveltime integral is ***stationary*** with respect to variations in the ray path, but for traveltime tomography using measured first arrivals it follows that the traveltimes must be ***minima***.]

Let us define τ^* to be the functional that yields the traveltime along the Fermat (least-time) ray path. Fermat's principle then states

$$\tau^*(\mathbf{s}) = \min_{P \in Paths} \tau^P(\mathbf{s}), \tag{2}$$

where *Paths* denotes the set of all continuous paths connecting the given source and receiver. The particular path that produces the minimum in (2) is denoted P^*. If more than one path produces the same minimum traveltime value, then P^* denotes any particular member in this set of minimizing paths.

To summarize, we have

$$\tau^P(\mathbf{s}) = \int_P s(\mathbf{x})\, dl^P, \tag{3}$$

$$\tau^*(\mathbf{s}) = \int_{P^*} s(\mathbf{x})\, dl^{P^*} = \min_P \int_P s(\mathbf{x})\, dl^P. \tag{4}$$

The traveltime functional $\tau^*(\mathbf{s})$ is stationary with respect to small variations in the path $P^*(s)$.

Snell's law is well-known to be a consequence of the stationarity of the traveltime functional [Feynman et al, 1963].

1.3 Seismic Inversion

Suppose we have a set of observed traveltimes, $t_1, \ldots, t_m$, from m source-receiver pairs in a medium of slowness $s(\mathbf{x})$. Let P_i be the Fermat ray path connecting the ith source-receiver pair. In the absence of observational errors, we can write

$$\int_{P_i} s(\mathbf{x})\, dl^{P_i} = t_i, \quad i = 1, \ldots, m. \tag{5}$$

Given a block model of slowness, let l_{ij} be the length of the ith ray path through the jth cell:

$$l_{ij} = \int_{P_i \cap \mathrm{cell}_j} dl^{P_i}. \tag{6}$$

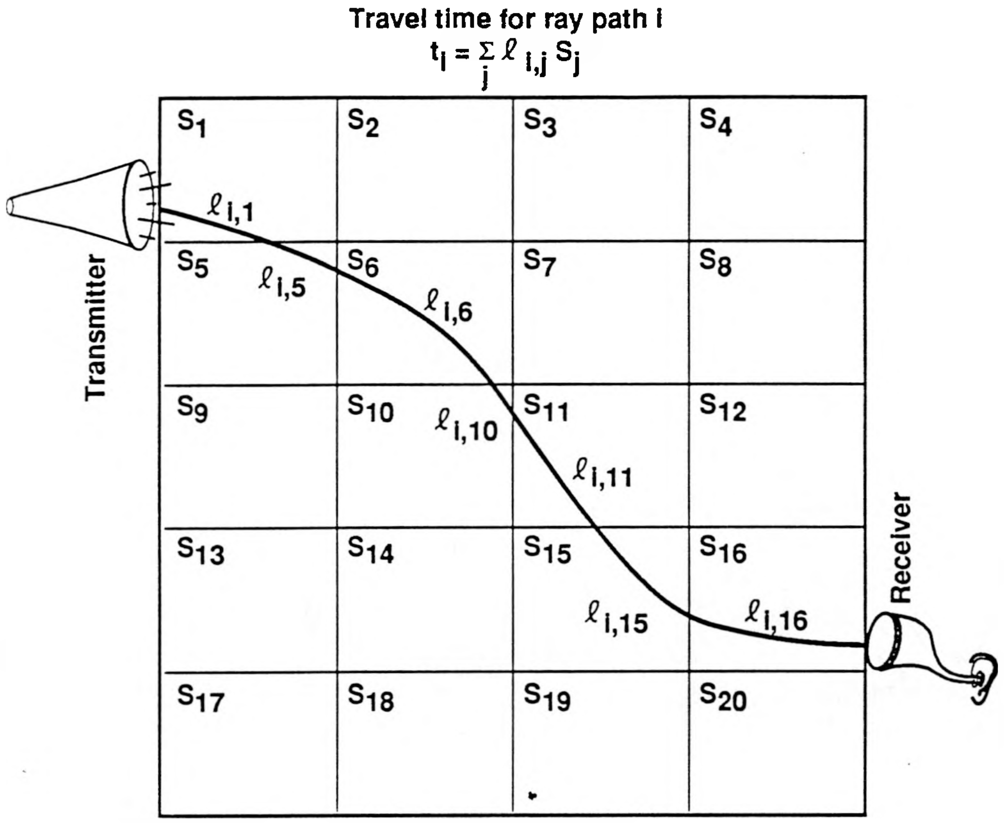

Figure 1: Schematic illustration of acoustic ray paths through a block velocity model.

Given a model with n cells, (5) can then be written

$$\sum_{j=1}^{n} l_{ij} s_j = t_i, \quad i = 1, \ldots, m. \tag{7}$$

Note that for any given i, the ray-path lengths l_{ij} are zero for most cells j, as a given ray path will in general intersect only a few of the cells in the model. Figure 1 illustrates the ray path intersections for a 2-D block model.

We can rewrite (7) in matrix notation by defining the column vectors $\mathbf{s}$ and $\mathbf{t}$ and the matrix $\mathbf{M}$ as follows:

$$\mathbf{s} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}, \quad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ l_{21} & l_{22} & \cdots & l_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & l_{mn} \end{pmatrix}. \tag{8}$$

Equation (7) then becomes

$$\mathbf{M}\mathbf{s} = \mathbf{t}. \tag{9}$$

1.4 Linear vs Nonlinear Tomography

We now define three problems in the context of Eq. (9).

In the *forward* problem, we are given s; the goal is to determine $\mathbf{M}$ and $\mathbf{t}$. This entails computing the ray path between each source and receiver (using a ray tracing algorithm) and then computing the traveltime integral along each path.

In the *linear tomography* problem, we are given $\mathbf{M}$ and t; the objective is to determine s. The assumption here is that the ray paths are known *a priori*, which is justified under a linear approximation that ignores the dependence of the ray paths on the slowness distribution. Typically, the ray paths are assumed to be straight lines connecting sources and receivers, adding a second connotation to the term "linear." Linear tomography is commonly practiced in medical imaging and in many geophysical situations as well.

In *nonlinear tomography*, we are given only t (along with the source and receiver locations); the goal is to infer both $\mathbf{M}$ and s. In this problem, the dependence of the ray paths on the slowness distribution is acknowledged. Nonlinear tomography is necessary for problems in which the slowness varies significantly across the medium of interest, which includes many seismic tomography problems. The ray paths in such media will show significant curvature (i.e., be nonlinear) in a way that cannot be known *a priori.*

The linear tomography problem can be solved with a variety of optimization techniques. In the least-squares method, for example, the normal solution for s is expressed analytically as

$$\hat{\mathbf{s}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{t}, \tag{10}$$

assuming the matrix inverse exists. If the inverse does not exist, then (10) must be "regularized." Typically, regularization is accomplished by adding a positive matrix to $\mathbf{M}^T\mathbf{M}$ and replacing the singular inverse in (10) by the inverse of the modified matrix [Herman, 1980].

In nonlinear tomography, an iterative algorithm is generally needed to find an approximate solution $\hat{\mathbf{s}}$. The basic structure of such an algorithm is as follows:

1. Set $\hat{\mathbf{s}}$ to a given initial model (a constant or the previously best-known geological model).
2. Compute the ray-path matrix $\mathbf{M}$ and traveltimes $\hat{\mathbf{t}}$ for $\hat{\mathbf{s}}$ and set $\Delta\mathbf{t} = \mathbf{t} - \hat{\mathbf{t}}$.
3. If $\Delta\mathbf{t}$ is sufficiently small, stop.
4. Find a model correction $\widehat{\Delta\mathbf{s}}$ as the solution to the linear tomography problem: $\mathbf{M}\widehat{\Delta\mathbf{s}} = \Delta\mathbf{t}$.
5. Update $\hat{\mathbf{s}}$ to the new model obtained by adding the model correction $\widehat{\Delta\mathbf{s}}$ to the previous model $\hat{\mathbf{s}}$.
6. Go to Step 2.

This algorithm looks very reasonable and in fact sometimes it actually works! But not always. For models with low slowness contrasts, the algorithm will converge to a sensible result. When the method fails, the failure mode is usually a divergence to a highly oscillatory model. *Ad hoc* procedures to reduce the possible range of slowness values and to guarantee a high degree of smoothness in the reconstructed model have commonly been introduced to deal with this instability. Such smoothness constraints come from external considerations (like the class of models we want the solution to lie in), not from the data. But a really satisfactory method of stabilizing the iteration scheme based on information in the data itself has been lacking.

Analyzing the algorithm, we see that there are really only two significant calculations contained in it. Step 2 is just the solution of the *forward problem* for $\hat{\mathbf{s}}$. This step should not introduce any instability, since it can be performed essentially as accurately as desired (if the computing budget is large enough). Step 4, on the other hand, is a *linear tomography* step imbedded in a nonlinear algorithm. We should be skeptical of this step. Linear inversion implicitly assumes that the updated model (after adding the model correction) is not so different from the previous model that the ray path matrix $\mathbf{M}$ should change significantly from one iteration to the next. If this implicit assumption is violated, then this step is not justified, and steps 4 and/or 5 in the algorithm must be modified.

Feasibility analysis supplies a set of rigorous physical constraints on the reconstruction proces. Experience has shown that smoothness constraints are not needed if feasibility constraints are applied. The next section introduces the concepts and a few of the results that follow from them in traveltime tomography.

2 Feasibility Analysis

The idea of using feasibility constraints in nonlinear programming problems is well established [Fiacco and McCormick, 1990]. However, it has only recently been realized that physical principles such as Fermat's principle actually lead to rigorous feasibility constraints for nonlinear inversion problems [Berryman, 1991]. The main practical difference between the standard analysis in nonlinear programming and the new analysis in nonlinear inversion is that, whereas the functions involved in nonlinear programming are often continuous, differentiable, and relatively easy to compute, the functionals in nonlinear inversion (*e.g.*, the traveltime functional) need not be continuous or differentiable and furthermore are very often comparatively difficult to compute.

We present the rigorous analysis here in a general setting. This analysis is important because it will help to characterize the solution set for the inversion problem, and it will help to clarify the questions about local and global minima of the inverson problem.

2.1 Feasibility Constraints

Equation (5) assumed that P_i is a Fermat (least-time) path. Now let us suppose that P_i is a trial ray path which may or may not be the least-time path. Fermat's principle allows us to write

$$\int_{P_i} s(\mathbf{x})dl^{P_i} \geq t_i. \tag{11}$$

When we discretize (11) for block models, it becomes

$$\mathbf{Ms} \geq \mathbf{t}. \tag{12}$$

Equations (11) and (12) can be interpreted as a set of inequality constraints on the slowness model s. When s obeys these m constraints, we say that s is *feasible.* When any of the constraints is violated, we say s is *infeasible.* The set of inequalities collectively will be called the *feasibility constraints.*

2.2 Properties of Traveltime Functionals

Lemma 2.1 *τ^P is a linear functional.*

The proof of this stems from the fact that integration is a linear functional of the integrand. Since it is linear, it also follows that τ^P is also convex, concave, and homogeneous.

Lemma 2.2 *τ^* is a homogeneous functional.*

Proof: Given $\gamma > 0$ we have

$$\tau^*(\gamma \mathrm{s}) = \min_P \tau^P(\gamma \mathrm{s}). \tag{13}$$

Using the linearity of τ^P,

$$\tau^*(\gamma \mathrm{s}) = \min_P \gamma \tau^P(\mathrm{s}) = \gamma \min_P \tau^P(\mathrm{s}) = \gamma \tau^*(\mathrm{s}). \quad \blacksquare \tag{14}$$

Lemma 2.3 *τ^* is a concave functional.*

Proof: Given slowness models s_1 and s_2 and $\lambda \in [0,1]$, let $\mathrm{s} = \lambda \mathrm{s}_1 + (1-\lambda)\mathrm{s}_2$. Letting $P^*(\mathrm{s})$ be the Fermat ray path for s, we have

$$\tau^*(\mathrm{s}) = \tau^{P^*(\mathrm{s})}(\mathrm{s}). \tag{15}$$

The linearity of τ^P then implies

$$\tau^*(\mathrm{s}) = \lambda \tau^{P^*(\mathrm{s})}(\mathrm{s}_1) + (1-\lambda)\tau^{P^*(\mathrm{s})}(\mathrm{s}_2). \tag{16}$$

Since τ^* minimizes τ^P for any fixed model, it must be the case that $\tau^{P^*(\mathrm{s})}(\mathrm{s}_1) \geq \tau^*(\mathrm{s}_1)$ and similarly for s_2. Further, λ and $(1-\lambda)$ are non-negative. Therefore, (16) implies

$$\tau^*(\mathrm{s}) \geq \lambda \tau^*(\mathrm{s}_1) + (1-\lambda)\tau^*(\mathrm{s}_2). \quad \blacksquare \tag{17}$$

2.3 Feasibility Sets

Given the set of observed traveltimes, t_i for $i = 1, \ldots, m$, we define two sets of models.

Definition 2.1 (local feasibility set) *The local feasibility set with respect to a set of trial ray paths $\mathcal{P} = \{P_1, \ldots, P_m\}$ and observed traveltimes $t_1, \ldots, t_m$ is*

$$\mathcal{F}^{\mathcal{P}} = \{\mathrm{s} \mid \tau_i^P(\mathrm{s}) \geq t_i, \quad \textit{for all} \quad i = 1, \ldots, m\}. \tag{18}$$

Definition 2.2 (global feasibility set) *The global feasibility set with respect to the observed traveltimes $t_1, \ldots, t_m$ is*

$$\mathcal{F}^* = \{\mathrm{s} \mid \tau_i^*(\mathrm{s}) \geq t_i, \quad \textit{for all} \quad i = 1, \ldots, m\}. \tag{19}$$

Now we show that the concavity of τ_i^P and τ_i^* implies the convexity of $\mathcal{F}^P$ and $\mathcal{F}^*$.

Theorem 2.1 *$\mathcal{F}^P$ is a convex set.*

Proof: Suppose $\mathrm{s}_1, \mathrm{s}_2 \in \mathcal{F}^P$ and let $\mathrm{s}_\lambda = \lambda \mathrm{s}_1 + (1 - \lambda)\mathrm{s}_2$ where $0 \leq \lambda \leq 1$. Since, for each i, τ_i^P is a concave (actually linear) functional, we have

$$\tau_i^P(\mathrm{s}_\lambda) \geq \lambda \tau_i^P(\mathrm{s}_1) + (1 - \lambda)\tau_i^P(\mathrm{s}_2). \tag{20}$$

(Although equality applies in the present case, the "greater than or equal to" is important in the next proof.) But $\tau_i^P(\mathrm{s}_1), \tau_i^P(\mathrm{s}_2) \geq t_i$ and λ and $(1 - \lambda)$ are non-negative. Therefore,

$$\tau_i^P(\mathrm{s}_\lambda) \geq \lambda t_i + (1 - \lambda)t_i = t_i. \tag{21}$$

Thus, $\mathrm{s}_\lambda \in \mathcal{F}^P$. ∎

Theorem 2.2 *$\mathcal{F}^*$ is a convex set.*

The proof proceeds in analogy with the previous proof, with τ_i^* replacing τ_i^P, but the inequalities come into play this time.

Theorem 2.3 *Given any model* s, *there exists $\gamma > 0$ such that $\gamma \mathrm{s} \in \mathcal{F}^*$.*

Proof: Let

$$\gamma = \max_{k \in \{1, \ldots, m\}} \frac{t_k}{\tau_k^*(\mathrm{s})}. \tag{22}$$

For any i, τ_i^* is homogeneous, implying

$$\tau_i^*(\gamma \mathrm{s}) = \gamma \tau_i^*(\mathrm{s}) = \tau_i^*(\mathrm{s}) \max_k \frac{t_k}{\tau_k^*(\mathrm{s})} \geq \tau_i^*(\mathrm{s}) \frac{t_i}{\tau_i^*(\mathrm{s})} = t_i. \tag{23}$$

We see that $\gamma \mathrm{s}$ satisfies all the feasibility constraints, so it is in $\mathcal{F}^*$.

We can decompose $\mathcal{F}^*$ into two parts: its *boundary* and its *interior.* The boundary of $\mathcal{F}^*$, denoted Bdy $\mathcal{F}^*$, comprises feasible models s which satisfy some feasibility constraint with equality, *i.e.*,

$$\mathrm{Bdy}\, \mathcal{F}^* = \{\mathrm{s} \in \mathcal{F}^* \mid \tau_i^*(\mathrm{s}) = t_i, \text{ for some } i\}. \tag{24}$$

Models in the interior of $\mathcal{F}^*$, denoted Int $\mathcal{F}^*$, satisfy all constraints with inequality:

$$\mathrm{Int}\, \mathcal{F}^* = \{\mathrm{s} \in \mathcal{F}^* \mid \tau_i^*(\mathrm{s}) > t_i, \text{ for all } i\}. \tag{25}$$

2.4 Convex Programming and Global Minima

We first define convex programming for first-arrival traveltime inversion, and then present some basic theorems about it.

Definition 2.3 *Let $\Phi(\mathrm{s})$ be any convex function of* s. *Then the convex nonlinear programming problem associated with Φ is to minimize $\Phi(\mathrm{s})$ subject to the global feasibility constraints $\tau_i^*(\mathrm{s}) \geq t_i$, for $i = 1, \ldots, m$.*

Definition 2.4 *Let $\Psi^{\mathcal{P}}(\mathrm{s}) = \sum_{i=1}^{m} w_i[\tau_i^P(\mathrm{s}) - t_i]^2$ for some positive weights $\{w_i\}$ and some set of ray paths $\mathcal{P} = \{P_1, \ldots, P_m\}$. Then, the convex linear programming problem associated with $\Psi^{\mathcal{P}}$ is to minimize $\Psi^{\mathcal{P}}(\mathrm{s})$ subject to the local feasibility constraints $\tau_i^P(\mathrm{s}) \geq t_i$, for $i = 1, \ldots, m$.*

Theorem 2.4 *Every local minimum* s^* *of the convex nonlinear programming problem associated with $\Phi(\mathrm{s})$ is a global minimum.*

Theorem 2.5 *Every local minimum* s^* *of the convex linear programming problem associated with $\Psi^{\mathcal{P}}(\mathrm{s})$ is a global minimum.*

Proof: This proof follows one given by Fiacco and McCormick [1990]. Let s^* be a local minimum. Then, by definition, there is a compact set $\mathcal{C}$ such that s^* is in the interior of $\mathcal{C} \cap \mathcal{F}^*$ and

$$\Phi(\mathrm{s}^*) = \min_{\mathcal{C} \cap \mathcal{F}^*} \Phi(\mathrm{s}). \tag{26}$$

If s is any point in the feasible set $\mathcal{F}^*$ and $0 \leq \lambda \leq 1$ such that $\mathrm{s}_\lambda^* \equiv \lambda \mathrm{s}^* + (1-\lambda)\mathrm{s}$ is in $\mathcal{C} \cap \mathcal{F}^*$, then

$$\Phi(\mathrm{s}) \geq \frac{\Phi(\mathrm{s}_\lambda^*) - \lambda\Phi(\mathrm{s}^*)}{1-\lambda} \geq \frac{\Phi(\mathrm{s}^*) - \lambda\Phi(\mathrm{s}^*)}{1-\lambda} = \Phi(\mathrm{s}^*). \tag{27}$$

The first step of (27) follows from the convexity of Φ and the second from the fact that s^* is a minimum in $\mathcal{C} \cap \mathcal{F}^*$. Convexity of $\mathcal{F}^*$ guarantees that the convex combination s_λ^* lies in the feasible set. This completes the proof of the first theorem.

The proof of the second theorem follows that of the first once we have shown that the function $\Psi^{\mathcal{P}}$ is convex. Consider a term of $\Psi^{\mathcal{P}}$

$$\begin{aligned}
[\tau_i^P(\lambda \mathrm{s}_1 + (1-\lambda)\mathrm{s}_2) - t_i]^2 &= [\lambda\tau_i^P(\mathrm{s}_1) + (1-\lambda)\tau_i^P(\mathrm{s}_2) - t_i]^2 \\
&= \lambda[\tau_i^P(\mathrm{s}_1) - t_i]^2 + (1-\lambda)[\tau_i^P(\mathrm{s}_2) - t_i]^2 \\
&\quad -\lambda(1-\lambda)[\tau_i^P(\mathrm{s}_1) - \tau_i^P(\mathrm{s}_2)]^2 \\
&\leq \lambda[\tau_i^P(\mathrm{s}_1) - t_i]^2 + (1-\lambda)[\tau_i^P(\mathrm{s}_2) - t_i]^2.
\end{aligned}$$

Then, if $\mathrm{s}_\lambda = \lambda \mathrm{s}_1 + (1-\lambda)\mathrm{s}_2$,

$$\Psi^{\mathcal{P}}(\mathrm{s}_\lambda) \leq \lambda\Psi^{\mathcal{P}}(\mathrm{s}_1) + (1-\lambda)\Psi^{\mathcal{P}}(\mathrm{s}_2), \tag{28}$$

so $\Psi^{\mathcal{P}}$ is a convex function. ▮

Although Theorem 2.4 provides a powerful result, it still requires ingenuity to find a convex functional appropriate for the nonlinear inversion problem.

Several methods of stabilizing the nonlinear inversion problem based on the feasibility constraints will now be discussed.

3 Nonlinear Reconstruction Algorithms

The introduction of feasibility constraints into the traveltime tomography problem [Berryman, 1989b; 1990; 1991; Lu and Berryman, 1990] offers a unique opportunity to develop a variety of new reconstruction algorithms. A few of the ones that have been explored will be discussed.

3.1 Linear and Nonlinear Programming

For our point of view, linear tomography maps easily into linear programming, and nonlinear tomography into nonlinear programming [Strang, 1986; Fiacco and McCormick, 1990].

If $\mathbf{u}^T = (1, \ldots, 1)$ is an m-vector of ones and $\mathbf{v}^T = (1, \ldots, 1)$ is an n-vector of ones, then

$$\mathbf{u}^T\mathbf{M} = \mathbf{v}^T\mathbf{C}, \tag{29}$$

where $\mathbf{C}$ is the coverage *matrix*, *i.e.*, the diagonal matrix whose diagonal elements are the column sums of the ray-path matrix. We will now define the coverage *vector* as

$$\mathbf{c} = \mathbf{C}\mathbf{v}. \tag{30}$$

3.1.1 Duality

The concept of duality in linear programming leads to some useful ideas both for linear and nonlinear traveltime tomography. We will first define the following:

Definition 3.1 *The primal problem for traveltime tomography is to find the minimum of* $\mathbf{c}^T\mathbf{s}$ *subject to* $\mathbf{M}\mathbf{s} \geq \mathbf{t}$ *and* $\mathbf{s} \geq 0$.

Definition 3.2 *The dual problem associated with the primal is to maximize* $\mathbf{w}^T\mathbf{t}$ *subject to* $\mathbf{w}^T\mathbf{M} \leq \mathbf{c}^T$ *and* $\mathbf{w} \geq 0$.

The m-vector $\mathbf{w}$ has no physical significance, but plays the role of a nonnegative weight vector. One of the first consequences of this formulation is that, if we multiply the primal inequality on the right by $\mathbf{w}^T$ and the dual inequality on the left by $\mathbf{s}$ for feasible $\mathbf{s}$ and $\mathbf{w}$, then

$$\mathbf{c}^T\mathbf{s} \geq \mathbf{w}^T\mathbf{M}\mathbf{s} \geq \mathbf{w}^T\mathbf{t}. \tag{31}$$

We introduce a Lagrangian functional

$$\begin{aligned} \mathcal{L}(\mathbf{s}, \mathbf{w}) &= \mathbf{c}^T\mathbf{s} + \mathbf{w}^T(\mathbf{t} - \mathbf{M}\mathbf{s}) & (32) \\ &= (\mathbf{c}^T - \mathbf{w}^T\mathbf{M})\mathbf{s} + \mathbf{w}^T\mathbf{t}. & (33) \end{aligned}$$

An admissible (feasible) weight vector is $\mathbf{w} = \mathbf{u}$. In fact, this is the only weight vector we need to consider because it saturates the dual inequality, producing equality in all components following (29) and (30). Thus, the dual problem in traveltime tomography is really trivial. We introduced it here because, despite its apparent triviality, there is one interesting feature.

In problems with nontrivial duality structure, it is possible to obtain useful bounds with inequalities equivalent to (31). Here we are left with only the condition

$$\mathbf{c}^T\mathbf{s} \geq \mathbf{u}^T\mathbf{t} = T, \tag{34}$$

which we could have derived directly from the feasibility conditions $\mathbf{M}\mathbf{s} \geq \mathbf{t}$ for $\mathbf{s}$. Equation (34) is not trivial however, and can play an important role in linear and nonlinear programming algorithms for traveltime tomography.

3.1.2 Relaxed feasibility constraints

Given the set of observed traveltimes, t_i for $i = 1, \ldots, m$, we define two more types of feasibility sets.

Definition 3.3 (relaxed local feasibility set) *The relaxed local feasibility set with respect to a set of trial ray paths $\mathcal{P} = \{P_1, \ldots, P_m\}$ and observed traveltimes $t_1, \ldots, t_m$ is*

$$\mathcal{R}^{\mathcal{P}} = \{\mathbf{s} \mid \sum_{i=1}^{m} \tau_i^P(\mathbf{s}) \geq \sum_{i=1}^{m} t_i\}. \tag{35}$$

Definition 3.4 (relaxed global feasibility set) *The relaxed global feasibility set with respect to a set of observed traveltimes $t_1, \ldots, t_m$ is*

$$\mathcal{R}^{*} = \{\mathbf{s} \mid \sum_{i=1}^{m} \tau_i^{*}(\mathbf{s}) \geq \sum_{i=1}^{m} t_i\}. \tag{36}$$

Theorem 3.1 *$\mathcal{R}^{\mathcal{P}}$ is a convex set.*

Theorem 3.2 *$\mathcal{R}^{*}$ is a convex set.*

Proof: Both theorems follow from the fact that a (nonnegatively) weighted sum of concave functionals is concave and the fact that the unit-weighted sums in the definitions of the sets $\mathcal{R}^{\mathcal{P}}$ and $\mathcal{R}^{*}$ are respectively sums of the concave functions $\tau_i^P(\mathbf{s})$ and $\tau_i^{*}(\mathbf{s})$. ▮

Theorem 3.3 *Any point $\mathbf{s}^{*}$ that lies simultaneously on the boundary of both $\mathcal{F}^{\mathcal{P}}$ and $\mathcal{R}^{\mathcal{P}}$ solves the inversion problem.*

Theorem 3.4 *Any point $\mathbf{s}^{*}$ that lies simultaneously on the boundary of both $\mathcal{F}^{*}$ and $\mathcal{R}^{*}$ solves the inversion problem.*

Proof: The boundary of $\mathcal{R}^{\mathcal{P}}$ is determined by the single equality constraint

$$\sum_{i=1}^{m} \tau_i^P(\mathbf{s}) = \sum_{i=1}^{m} t_i = T. \tag{37}$$

The boundary of $\mathcal{F}^{\mathcal{P}}$ is determined by the set of inequality constraints

$$\tau_i^P(\mathbf{s}) \geq t_i, \quad \text{for all} \quad i = 1, \ldots, m, \tag{38}$$

with equality holding for at least one of the constraints. Summing (38) gives

$$\sum_{i=1}^{m} \tau_i^P(\mathbf{s}) \geq T, \tag{39}$$

where the equality applies if and only if $\tau_i^P(\mathbf{s}) = t_i$ for all i. Therefore, any model $\mathbf{s}^{*}$ that satisfies both (37) and (38) must solve the inversion problem.

The proof of the second theorem follows the proof of the first, with $\tau^*(\mathbf{s})$ replacing $\tau^P(\mathbf{s})$ everywhere. ▮

If we have found the correct ray-path matrix for the inversion problem and the data are noise free, then we expect that the hyperplane defined by $\mathbf{c}^T\mathbf{s} = T$ will intersect the feasibility boundary exactly at the point or points that solve the inversion problem. If we have not found the correct ray-path matrix or there is uncorrelated noise in our data $\mathbf{t}$, then there will be a *splitting* between the hyperplane of constant total traveltime and the feasible region. The point (or points) of closest approach between the convex feasible set and the hyperplane may then be defined as the set of points *solving* the linear programming problem for fixed $\mathbf{M}$. An iterative nonlinear programming algorithm may then be constructed wherein the updated $\mathbf{M}$ is determined based on the solution of the last linear programming problem. This procedure converges if the degree of splitting (Euclidean distance) between the feasible set and the hyperplane of constant traveltime tends to zero from one iteration to the next.

3.2 Weighted Least-Squares

A good set of weights to use for weighted least-squares has been shown [Berryman, 1989a] to be $\mathbf{L}^{-1}$ (the inverse of the diagonal ray-length matrix) for the traveltime errors and $\mathbf{C}$ (the diagonal cell-coverage matrix) for the smoothing or regularization term in a damped least-squares method. The arguments were based on assumptions of small deviations from a constant background or on the desire to precondition the ray-path matrix so its eigenvalues were normalized to the range $-1 \leq \lambda \leq 1$.

The methods used to choose these weights were based on *linear tomography* ideas. We should now try to see if these ideas need modification for *nonlinear tomography*. Let $\mathbf{s}$ be the latest estimate of the slowness model vector in an iterative inversion scheme. Then, if $\mathbf{u}^T = (1, \ldots, 1)$ is an m-vector of ones and $\mathbf{v}^T = (1, \ldots, 1)$ is an n-vector of ones,

$$\mathbf{Ms} = \mathbf{Tu}, \tag{40}$$

$$\mathbf{M}^T\mathbf{u} = \mathbf{Cv} \equiv \mathbf{Ds}, \tag{41}$$

where $\mathbf{C}$ is the coverage matrix (diagonal matrix containing the column sums of $\mathbf{M}$) defined previously and the two new matrices ($\mathbf{T}$ and $\mathbf{D}$) are diagonal matrices whose diagonal elements are T_{ii}, the estimated traveltime for the ith ray path through the model $\mathbf{s}$,

$$T_{ii} = \sum_{j=1}^{n} l_{ij}s_j, \tag{42}$$

and D_{jj} where

$$D_{jj} \equiv C_{jj}/s_j = \sum_{i=1}^{m} l_{ij}/s_j. \tag{43}$$

For the sake of argument, let the inverse of the diagonal traveltime matrix $\mathbf{T}^{-1}$ be the weight matrix, and compute the scaled least-squares point. The least-squares functional takes the form

$$\psi(\gamma) = (\mathbf{t} - \mathbf{M}\gamma\mathbf{s})^T\mathbf{T}^{-1}(\mathbf{t} - \mathbf{M}\gamma\mathbf{s}), \tag{44}$$

which has its minimum at

$$\gamma = \frac{\mathbf{s}^T\mathbf{M}^T\mathbf{T}^{-1}\mathbf{t}}{\mathbf{s}^T\mathbf{M}^T\mathbf{T}^{-1}\mathbf{M}\mathbf{s}}. \tag{45}$$

Equation (45) can be rewritten using (40) as

$$\gamma = \frac{\mathbf{u}^T\mathbf{t}}{\mathbf{u}^T\mathbf{T}\mathbf{u}}. \tag{46}$$

The factor γ that minimizes the least-squares error is therefore the one that either increases or decreases the total traveltime of the model s so it equals that of the data. If we assume that the measurement errors in the traveltime data $\mathbf{t}$ are unbiased, then it is very reasonable to choose models that have this property, because the total traveltime $\mathbf{u}^T\mathbf{t} = T$ will tend to have smaller error (by a factor of $m^{-\frac{1}{2}}$) than the individual measurements.

We see that requiring the models s to have the same total traveltime as the data is equivalent to requiring that the models all lie in the hyperplane defined by

$$\mathbf{u}^T\mathbf{M}\mathbf{s} = \mathbf{v}^T\mathbf{C}\mathbf{s} = \mathbf{c}^T\mathbf{s} = T. \tag{47}$$

But this is precisely the same hyperplane (34) that arose naturally in the earlier discussion of linear and nonlinear programming.

To carry this analysis one step further, consider the weighted least-squares problem

$$\phi_\mu(\mathbf{s}) = (\mathbf{t} - \mathbf{M}\mathbf{s})^T\mathbf{T}^{-1}(\mathbf{t} - \mathbf{M}\mathbf{s}) + \mu(\mathbf{s} - \mathbf{s}_0)^T\mathbf{D}(\mathbf{s} - \mathbf{s}_0), \tag{48}$$

where we assume that the starting model $\mathbf{s}_0$ satisfies $\mathbf{c}^T\mathbf{s}_0 = T$. Then, the minimum of (48) occurs for $\mathbf{s}_\mu$ satisfying

$$(\mathbf{M}^T\mathbf{T}^{-1}\mathbf{M} + \mu\mathbf{D})(\mathbf{s}_\mu - \mathbf{s}_0) = \mathbf{M}^T\mathbf{T}^{-1}(\mathbf{t} - \mathbf{M}\mathbf{s}_0). \tag{49}$$

Multiplying (49) on the left by $\mathbf{s}_0^T$, we find that

$$(1 + \mu)\mathbf{c}^T(\mathbf{s}_\mu - \mathbf{s}_0) = \mathbf{u}^T(\mathbf{t} - \mathbf{M}\mathbf{s}_0) = 0, \tag{50}$$

so the solution of the weighted least-squares problem (49) also has the property that its estimated total traveltime for all rays is equal to that of the data

$$\mathbf{c}^T\mathbf{s}_\mu = \mathbf{c}^T\mathbf{s}_0 = T. \tag{51}$$

Our conclusion is that the particular choice of weighted least-squares problem (49) has the *unique* property of holding the total estimated traveltime equal to the total of the measured traveltimes, *i.e.*, it constrains the least-squares solution to lie in the hyperplane $\mathbf{c}^T\mathbf{s} = T$. Assuming that the traveltime data are themselves unbiased (*i.e.*, $\mathbf{u}^T\Delta\mathbf{t} = 0$ where $\Delta\mathbf{t}$ is the measurement error vector), the result s is an unbiased estimator of the slowness. Moreover, this property is maintained for *any* value of the damping parameter μ. This result provides a connection between the linear programming approach and weighted linear least-squares. We can now use weighted least-squares and the formula (49) in a linear program if we like as a means of moving around in the hyperplane $\mathbf{c}^T\mathbf{s} = T$.

A general analysis of the eigenvalue structure of weighted least-squares shows that

$$\frac{L_{ii}C_{jj}}{T_{ii}D_{jj}} \geq \lambda^2, \tag{52}$$

which must hold true for all values of i, j. From (43), we have $C_{jj}/D_{jj} = s_j$ so

$$\frac{L_{ii}s_j}{T_{ii}} \geq \frac{L_{ii}s_{\min}}{T_{ii}} \geq \lambda^2, \tag{53}$$

and from the definition of T_{ii} we have

$$T_{ii} = \sum_{j=1}^{n} l_{ij}s_j \geq L_{ii}s_{\min}. \tag{54}$$

We find that this choice of weight matrices constrains the eigenvalues to be bounded above by unity $1 \geq \lambda^2$.

If the matrix $\mathbf{M}$ is very large, it may be impractical to solve (49) by inverting the matrix $(\mathbf{M}^T\mathbf{T}^{-1}\mathbf{M} + \mu\mathbf{D})$. Instead, we may choose to use a method we call "simple iteration." For example, suppose that the kth iteration yields the model vector $\mathbf{s}_\mu^{(k)}$. Then, one choice of iteration scheme for finding the next iterate is

$$\mathbf{D}\mathbf{s}_\mu^{(k+1)} = \mathbf{D}\mathbf{s}_\mu^{(k)} + \mathbf{M}^T\mathbf{T}^{-1}(\mathbf{t} - \mathbf{M}\mathbf{s}_0) - (\mathbf{M}^T\mathbf{T}^{-1}\mathbf{M} + \mu\mathbf{D})(\mathbf{s}_\mu^{(k)} - \mathbf{s}_0). \tag{55}$$

It is not hard to show that this iteration scheme converges as long as the damping parameter is chosen so that $0 < \mu < 1$. Furthermore, if we multiply (55) on the left by $\mathbf{s}_0^T$, we find that

$$\mathbf{c}^T(\mathbf{s}_\mu^{(k+1)} - \mathbf{s}_\mu^{(k)}) = (1+\mu)\mathbf{c}^T(\mathbf{s}_0 - \mathbf{s}_\mu^{(k)}). \tag{56}$$

It follows from (56) that, if $\mathbf{c}^T\mathbf{s}_0 = T$ and if $\mathbf{s}_\mu^{(0)} = \mathbf{s}_0$, then

$$\mathbf{c}^T\mathbf{s}_\mu^{(k)} = T \tag{57}$$

for all k. Thus, all the iterates stay in the hyperplane of constant total traveltime. If we choose not to iterate to convergence, then this desirable feature of the exact solution $\mathbf{s}_\mu$ proven in (51) is still shared by every iterate $\mathbf{s}_\mu^{(k)}$ obtained using this scheme.

3.3 Stable Algorithm for Traveltime Tomography

Now we combine several ideas into an algorithm for nonlinear traveltime tomography. We recall that such algorithms are inherently iterative. In the general iterative algorithm posed earlier, the questionable step was how to update the current model $\hat{\mathbf{s}}$ to obtain a new model. Here we propose a method for choosing this step [Berryman, 1989b; 1990].

Let $\mathbf{s}^{(k)}$ be the current model. An algorithm for generating the updated model $\mathbf{s}^{(k+1)}$ is as follows:

1. Set $\mathbf{s}_1$ to the scaled least-squares model:

$$\mathbf{s}_1 = \hat{\mathbf{s}}_{\mathrm{LS}[\mathbf{s}^{(k)}]}.$$

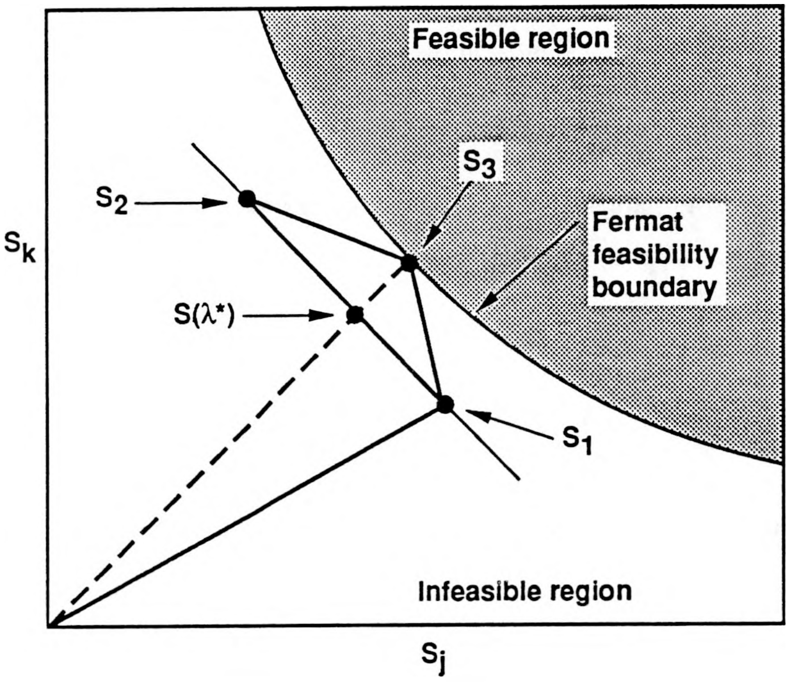

Figure 2: Snapshot of one iteration in a nonlinear tomography algorithm based on feasibility constraints.

2. Set s_2 to the damped least-squares model with respect to s_1:

$$s_2 = \hat{s}_{LS[s_1,\mu]}.$$

3. Define the family of models

$$s(\lambda) = (1 - \lambda)s_1 + \lambda s_2,$$

where $\lambda \in [0, 1]$.

4. Solve for λ^*, defined so that $s(\lambda^*)$ yields the fewest number of feasibility violations. The number of feasibility violations is defined as the number of ray paths for which $t_i > \tau^*(s(\lambda))$.

5. If λ^* is less than some preset threshold (say 0.05 or 0.1), reset it to the threshold value.

6. Set $s^{(k+1)} = s(\lambda^*)$.

The algorithm is illustrated in Fig. 2. The model labeled s_3 is a scaled version of $s(\lambda^*)$, scaled so that s_3 is on the boundary of the feasible region ($\mathcal{F}^*$). The iteration sequence stops when the perimeter of the triangle formed by s_1, s_2 and s_3 drops below a prescribed threshold.

This algorithm has been tested on several problems both with real and with synthetic data and also compared with a traditional damped least-squares algorithm (i.e., setting $\lambda^* = 1$ on each iteration). The new algorithm was found to be very stable and avoids the large oscillations in slowness often found in traditional least-squares methods.

3.4 Earthquake Sources

When we do not have control over the seismic source location and timing as in the case of earthquakes, the absolute traveltimes are not accurately known and it is important to understand how relative traveltimes may be used in seismic tomography [Aki, Christoffersson, and Husebye, 1977].

Rigorous application of the feasibility constraints $\mathbf{Ms} \geq \mathbf{t}$ requires knowledge of the absolute traveltimes. When such information is sparse or unavailable, we can use the known gross geological structure of the region to estimate the mean traveltime. Then we remove the meaningless mean of the relative data T/m and add back in the geological mean τ_0.

The *remove-the-mean operator* $\mathbf{R}$ for an m-dimensional vector space is defined as

$$\mathbf{R} = \mathbf{I} - \mathbf{u}\frac{1}{m}\mathbf{u}^T, \tag{58}$$

where $\mathbf{u}^T = (1, \ldots, 1)$ is an m-vector of ones. Note that $\mathbf{RR} = \mathbf{R}$ so $\mathbf{R}$ is a projection operator. Then, we see that $\mathbf{R}$ applied to the traveltime vector $\mathbf{t}$ gives

$$\mathbf{Rt} = \mathbf{t} - \frac{T}{m}\mathbf{u}, \tag{59}$$

where $T/m = \mathbf{u}^T\mathbf{t}/m$ is the mean traveltime of the data set. Applying $\mathbf{R}$ to the ray-path matrix, we have

$$\mathbf{RM} = \mathbf{M} - \mathbf{u}\frac{T}{m}\mathbf{v}^T\mathbf{C} = \mathbf{M} - \mathbf{u}\frac{T}{m}\mathbf{c}^T. \tag{60}$$

The standard procedure for this problem is to solve the equation

$$\mathbf{M}'\mathbf{s} = \mathbf{t}', \tag{61}$$

where $\mathbf{M}' = \mathbf{RM}$ and $\mathbf{t}' = \mathbf{Rt}$. To apply the feasibility constraints, we must modify the problem to

$$\mathbf{Ms} \geq \mathbf{Rt} + \tau_0\mathbf{I}_m. \tag{62}$$

Hidden in this analysis is the fact that the earthquake sources are often far from the region to be imaged, so the "effective" source locations may be placed at the boundaries of the region to be imaged.

If we have predetermined the mean for the traveltime data, then it is clearly desirable to use an inversion procedure that preserves this mean, *i.e.*, choosing $\Delta\mathbf{s}$ so that

$$\frac{\mathbf{u}^T\mathbf{M}(\mathbf{s} + \Delta\mathbf{s})}{m} = \tau_0 \tag{63}$$

for all $\Delta\mathbf{s}$. Preserving the mean is equivalent to preserving the total traveltime along all ray paths, so

$$\mathbf{c}^T(\mathbf{s} + \Delta\mathbf{s}) = m\tau_0. \tag{64}$$

In other words, vary $\mathbf{s}$ so it stays in the hyperplane determined by (64). But we have studied just this problem using linear programming (34) and also using weighted least-squares (51). So we do not need to develop any new inversion methods for this special case.

3.5 Parallel Computation

Traveltime tomography algorithms tend to be parallelizable in a variety of ways. The use of the feasibility constraints only increases the degree of parallelism that is achievable by these algorithms.

First, the *forward modeling* may be parallelized. If the forward problem is solved using either shooting or bending methods [Prothero et al., 1988; Nelder and Mead, 1965], then it is straightforward to parallelize the code because each ray may be computed independently of the others, and therefore in parallel. If the forward problem is solved using a finite difference algorithm or a full wave equation method, then whether the algorithm is parallelizable or not depends on the details of the particular algorithm. For example, Vidale's method [Vidale, 1988; 1990] is not directly parallelizable, but a recent related method by von Trier and Symes [1991] is.

Second, the use of the feasibility constraints in inversion algorithms suggests that it might be advantageous to map out the feasibility boundary and then use the information gained to search for improved agreement between the model and the data. Mapping the feasibility boundary can be done completely in parallel. Each model $\mathbf{s}$ may be treated in isolation, computing the best ray-path matrix for the model, and then finding the scaled model in the direction of $\mathbf{s}$ that intersects the feasibility boundary. The difficulty with this method is that it requires a figure of merit (in real problems) to help us determine whether one point on the feasibility is better than another. In ideal circumstances (no data error and infinite precision in our computers), the figure of merit would be the number of ray paths that achieve equality while satisfying the feasibility constraints

$$\mathbf{M}\mathbf{s} \geq \mathbf{t}. \tag{65}$$

When that number equals the number of ray paths, we have found an exact solution and, as the number increases towards this maximum value during an iterative procedure, the trial models $\mathbf{s}$ must be converging towards this solution. But in real problems, a figure of merit based on the number of equalities in (65) is not useful.

In a series of numerical experiments [joint work with A. J. DeGroot], we have found that a useful figure of merit for real problems is the nonlinear least-squares functional

$$\Psi(\mathrm{s}) = \sum_{i=1}^{n} w_i [\tau_i^*(\mathrm{s}) - \mathrm{t}_i]^2. \tag{66}$$

If we have found an exact solution s^* to the inversion problem, (66) will vanish at that point on the feasibility boundary — the global minimum. As we approach this minimum, (66) is evaluated at an arbitrary point on the feasibility boundary and the values in a cluster of such points are compared, our analysis of convex programming [Berryman, 1991] shows that the points with the smallest values of (66) form a convex set. The smallest value we find may not be zero, in which case no exact solution to our inversion problem has been found. This procedure has been implemented on a parallel processing machine, and the results obtained using this algorithm with the figure of merit (66) are comparable to those of the stable algorithm discussed earlier.

Acknowledgments

I thank A. J. DeGroot, S.-Y. Lu, and W. L. Rodi for helpful conversations. This work was performed under the auspices of the U. S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48 and supported specifically by the Geosciences Research Program of the DOE Office of Energy Research within the Office of Basic Energy Sciences, Division of Engineering and Geosciences.

References

Aki, K., A. Christoffersson, and E. S. Husebye, 1976, Determination of the three-dimensional seismic structure of the lithosphere, *J. Geophys. Res.* **82**, 277–296.

Berryman, J. G., 1989a, Weighted least-squares criteria for seismic traveltime tomography, *IEEE Trans. Geosci. Remote Sensing* **27**, 302–309.

Berryman, J. G., 1989b, Fermat's principle and nonlinear traveltime tomography, *Phys. Rev. Lett.* **62**, 2953–2956.

Berryman, J. G., 1990, Stable iterative reconstruction algorithm for nonlinear traveltime tomography, *Inverse Problems* **6**, 21–42.

Berryman, J. G., 1991, Convexity properties of inverse problems with variational constraints, *J. Franklin Inst.* **328**, 1–13.

Feynman, R. P., R. B. Leighton, and M. Sands, 1963, *The Feynman Lectures on Physics, Vol. I*, Addison-Wesley, Reading, Massachusetts, Chapter 26.

Fiacco, A. V. and G. P. McCormick, 1990, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, SIAM, Philadelphia, Chapter 6, 86–112.

Herman, G. T., 1980, *Image Reconstruction from Projections – The Fundamentals of Computerized Tomography*, Academic, New York, Chapter 6, 100–107.

Justice, J. H., A. A. Vassiliou, S. Singh, J. D. Logel, P. A. Hansen, B. R. Hall, P. R. Hutt, and J. J. Solanski, 1989, Tomographic imaging in hydrocarbon reservoirs, *J. Imaging Sys. Tech.* **1**, 62–72.

Lu, S.-Y., and J. G. Berryman, 1990, Inverse scattering, seismic traveltime tomography, and neural networks, *Intern. J. Imaging Sys. Tech.* **2**, 112–118.

Nelder, J. A., and R. Mead, 1965, A simplex method for function minimization, *Computer J.* **7**, 308–313.

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

Strang, G., 1986, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA, Chapter 8, 665–734.

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

Vidale, J. E., 1988, Finite-difference calculation of travel time, *Bull. Seismol. Soc. Am.* **78**, 2062–2076.

Vidale, J. E., 1990, Finite-difference calculation of travel time in 3-D, *Geophysics* **55**, 521–526.