FERMAT'S PRINCIPLE AND NONLINEAR TRAVELTIME TOMOGRAPHY

James G. Berryman

Lawrence Livermore National Laboratory
P. O. Box 808 L-156
Livermore, CA 94550
and

Courant Institute of Mathematical Sciences
New York University
251 Mercer St.
New York, NY 10012

ABSTRACT

Fermat's principle shows that a definite convex set of feasible slowness models – depending only on the traveltime data – exists for the fully nonlinear traveltime inversion problem. In a new iterative reconstruction algorithm, the minimum number of nonfeasible ray paths is used as a figure of merit to determine the optimum size of the model correction at each step. The numerical results show that the new algorithm is robust, stable, and produces very good reconstructions even for high contrast materials where standard methods tend to diverge.

PACS numbers: 42.30.Wb, 03.40.Kf, 43.60.+d, 93.85.+q

Traveltime tomography reconstructs a slowness (reciprocal wave speed) model from measured traveltimes for first arrivals. The locations of sources and receivers are assumed known, but the actual ray paths are not known and must be determined along with the model slowness. Fermat's principle[1] – that the path taken is the one of least traveltime – has been used extensively in forward modeling, *i.e.*, given the slowness model Fermat's principle determines the ray paths. However, Fermat's principle may also be applied in an entirely different way during the reconstruction of the slowness model using first arrival traveltime data, as we shall show.

To set notation, let t be the measured traveltime m-vector such that $t^T = (t_1, \ldots, t_m)$, where t_i is the traveltime along the i-th ray path (a superscript T implies the transpose). We form our model in two-dimensions by dividing the rectangular region enclosed by our sources and receivers into rectangular cells of constant slowness. In three dimensions, the cells are blocks of constant slowness. Then, s is the model slowness n-vector $s^T = (s_1, \ldots, s_n)$, with s_j being the slowness of the j-th cell. For forward modeling, s and t are related by the equation

$$Ms = t, \tag{1}$$

where M is an $m \times n$ matrix whose matrix elements $l_{i,j}$ are determined by the length of the i-th ray path as it passes through the j-th cell. Eq. (1) simply states that the total traveltime along a ray path is the sum of the traveltimes through each of the cells traversed by the ray. Fermat's principle is often used in forward modeling to determine M and therefore t when s is given.

The inverse problem associated with (1) starts with traveltime data t and attempts to find the corresponding slowness model s and ray-path matrix M. We will now depart from traditional methods by applying Fermat's principle in a new way. The forward problem (1) is replaced by the m feasibility constraints

$$(Ms)_i \geq t_i. \tag{2}$$

This fact follows from Fermat's principle: the first arrival necessarily followed the path of minimum traveltime for the model s. Thus, (2) must be satisfied by *any ray-path matrix* M if s is the true model and therefore any model that violates (2) along any of the ray paths is not a feasible model. An exact solution to the inversion problem is found if and only if all of inequalities in (2) become identities for some choice of model slowness vector s. (Solutions are generally not unique unless the ray-path matrix M is of full rank. Uniqueness is obtained by using additional physical constraints.) For each of the m inequality constraints (2), the limiting equality is the equation for a hyperplane in the n-dimensional slowness model space. The feasible region is bounded by these hyperplanes and by the planes determined by positivity of slowness in all cells j,

$$s_j > 0. \tag{3}$$

The two sets of inequalities (2) and (3) guarantee that the feasible region of the model space is convex. Thus, for fixed ray-path matrix M, the set of all feasible models s includes all models either inside the feasible region or on the feasibility boundary determined by M and t.

So far the argument has been pertinent only to linear traveltime tomography (*i.e.*, fixed ray-path matrix M). However, it is a small step to see that the constraints (2) imply the existence of a definite convex set in the model space containing *all* the feasible models for arbitrary choices of the ray-path matrix: Since any point s that is nonfeasible for *any particular choice of* M must lie outside of the global feasibility set, it follows that the intersection of the feasible sets for all choices of M determines the global (nonlinear) feasibility set. This global set must be convex since it is the intersection of convex sets. Furthermore, an exact solution of the inverse problem (*i.e.*, assuming the data are consistent) must lie on the boundary of this global convex set. Finally, we note that the location of the global feasibility boundary depends only on the set of measured traveltimes t.

Another new concept that is useful in computations is that of "feasibility violation number" $N_M(s)$. For any combination of ray-path matrix M, slowness vector s, and measured traveltimes t, the number of rays violating the constraints (2) is given by

$$N_M(s) \equiv \sum_{i=1}^{m} \theta[t_i - (Ms)_i], \tag{4}$$

where the step function $\theta(x)$ is defined by

$$\theta(x) = \begin{cases} 0, & \text{for } x \le 0; \\ 1, & \text{for } x > 0. \end{cases} \tag{5}$$

The number $N_M(s)$ is equal to zero in the feasible region. Furthermore, it is clearly a monotonically increasing function of distance from the local feasibility boundary associated with M – once one of the hyperplanes of (2) is crossed we will never cross it again if we keep moving in the same direction in the model space. Thus, $N_M(s)$ is cheap to compute and gives us a rough idea of how close we are to the feasibility boundary.

The feasibility violation number may be used to classify all models in the nonfeasible region of the model space. For example, contours of constant N_M may be drawn starting with $N_M = 0$ on the feasibility boundary.

Now define $u^T = (1, \ldots, 1)$, an m-vector of ones, and $v^T = (1, \ldots, 1)$, an n-vector of ones. Then,

$$Mv = Lu \tag{10}$$

and

$$M^T u = Cv. \tag{11}$$

The diagonal matrices L and C have the row sums L_i and column sums C_j

$$L_i = \sum_{j=1}^{n} l_{i,j}, \qquad C_j = \sum_{i=1}^{m} l_{i,j} \tag{12}$$

as their diagonal elements. The quantity L_i is seen to be the total length of path i. The quantity C_j is the total length of all the ray-path segments that pass through cell j, so we will call this the "coverage" of cell j. Any cell with $C_j = 0$ is uncovered and therefore lies outside the span of our data for the current choice of ray paths. We retain only the covered cells in the reduced slowness vector $\tilde{s}$ of length $\tilde{n} \le n$. The matrix M may similarly be reduced to $\widetilde{M}$ by deleting the corresponding columns of zeros. Finally, the diagonal matrix C is modified to include only the nonzero sums in (12). For simplicity of notation, we assume that $\tilde{n} = n$ in the following discussion.

An eigenvalue problem pertinent for high contrast reconstructions takes the form

$$\begin{pmatrix} 0 & M \\ M^T & 0 \end{pmatrix} \begin{pmatrix} w_\lambda \\ x_\lambda \end{pmatrix} = \lambda \begin{pmatrix} \hat{T} & 0 \\ 0 & \hat{D} \end{pmatrix} \begin{pmatrix} w_\lambda \\ x_\lambda \end{pmatrix}. \tag{13}$$

where, for $\lambda = 1$, $w_1 = u$, and $x_1 = s_b$ is the current best estimate of the reconstructed slowness. Then, by analogy with (10) and (11),

$$Ms_b = \hat{T}u, \tag{14}$$

$$M^T u = Cv = \hat{D} s_b \tag{14}$$

where $\hat{T}$ and $\hat{D}$ are diagonal matrices with elements given by

$$\hat{T}_i = \sum_{j=1}^{n} l_{i,j} (s_b)_j \qquad \text{for } 1 \le i \le m \tag{16}$$

and

$$\hat{D}_j = \sum_{i=1}^{m} l_{i,j}/(s_b)_j = C_j/(s_b)_j \qquad \text{for } 1 \le j \le n. \tag{17}$$

The $\hat{T}_i$s are the traveltimes of the current ray paths through the current model s_b. The $\hat{D}_j$ is the coverage of the j-th cell divided by the slowness of that cell; the dimensions of $\hat{D}_j$ are $(\text{length})^2/\text{time}$ — or, the same as that of a diffusion coefficient.

Now we will transform (13) to a canonical form using

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} = \begin{pmatrix} 0 & \hat{T}^{-\frac{1}{2}} M \hat{D}^{-\frac{1}{2}} \\ \hat{D}^{-\frac{1}{2}} M^T \hat{T}^{-\frac{1}{2}} & 0 \end{pmatrix} \tag{18}$$

and

$$\begin{pmatrix} y_\lambda \\ z_\lambda \end{pmatrix} = \begin{pmatrix} \hat{T}^{\frac{1}{2}} w_\lambda \\ \hat{D}^{\frac{1}{2}} x_\lambda \end{pmatrix}. \tag{19}$$

The eigenvalue problem (12) then is transformed into

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} y_\lambda \\ z_\lambda \end{pmatrix} = \lambda \begin{pmatrix} y_\lambda \\ z_\lambda \end{pmatrix}. \tag{20}$$

We wish to emphasize now that, with the normalization that has been performed to produce A, *the current slowness model s_b gives rise to the eigenvector of (20) with the highest eigenvalue and that eigenvalue is unity — i.e., $A^T A z_b = z_b$.*

Now we will consider two weighted least-squares fitting problems.[2] In both examples, the weights $\hat{T}$ and $\hat{D}$ have been incorporated directly into the normalization factors for the matrices A, and the eigenvector components y_λ and z_λ. The first problem is to find the slowness $s = \gamma s_b$ in the direction of s_b giving the best least-squares fit to the measured traveltime data. If the normalized traveltime measurement vector is given by

$$\hat{y} = \hat{T}^{-\frac{1}{2}} t, \tag{21}$$

then the problem is to find γ such that

$$\psi(\gamma) = (\hat{y} - A\gamma z_b)^T(\hat{y} - A\gamma z_b) \tag{22}$$

achieves its minimum. This value is found to be

$$\gamma = \frac{z_b^T A^T \hat{y}}{z_b^T A^T A z_b} = \frac{z_b^T A^T \hat{y}}{z_b^T z_b}. \tag{23}$$

since $A^T A z_b = z_b$. If the optimum scale factor has already been found and the value z_b scaled appropriately, then $\gamma = 1$ in (23). We will assume this is the case for the remainder of this discussion.

Now consider a second weighted least-squares problem. Consider the objective function

$$\phi_\mu(z) = (\hat{y} - Az)^T(\hat{y} - Az) + \mu(z - z_b)^T(z - z_b) \tag{24}$$

where μ is a damping parameter.[3,4] The minimum of (24) occurs at $z = z_\mu$ where z_μ satisfies

$$(A^T A + \mu I)(z_\mu - z_b) = A^T\hat{y} - z_b. \tag{25}$$

To arrive at (25), we again used the fact that $A^T A z_b = z_b$.

Now notice that the right hand side of (25) is orthogonal to z_b, *i.e.*,

$$z_b^T(A^T\hat{y} - z_b) = 0. \tag{26}$$

Eq. (26) follows from (23) when $\gamma = 1$. Applying z_b^T to (25) then gives

$$(1 + \mu)z_b^T(z_\mu - z_b) = 0, \tag{27}$$

so that z_μ lies in a hyperplane orthogonal to z_b.

Figure 1 illustrates the ideas presented so far and will also help to clarify the ideas underlying the new algorithm we have developed.

The key ideas underlying our new algorithm may now be summarized as follows: Given a set of transmitter-receiver pairs and any model slowness s, Fermat's principle may be used to find the ray path matrix M associated both with s and with any slowness γs (where $\gamma > 0$) in the same direction as s. An optimum scale factor γ may be found by doing a weighted least-squares fit to the traveltime data.

Having found the optimum slowness $s_b = \gamma s$ in the given direction, we next attempt to improve the model by finding another direction in the slowness vector space that gives a still better fit to the traveltime data. As many others have done, we first compute a damped least-squares solution $s_\mu = \hat{D}^{-\frac{1}{2}} z_\mu$ using (25). Next we note that both of the points found so far are guaranteed to lie in the nonfeasible part of the vector space – at least one and generally about half of the ray paths for both of these models will have traveltimes shorter than that of the measured data. Furthermore, although the point s_μ gives a better fit to the traveltime data, this fit is certainly spurious to some extent because it is based on the wrong ray path matrix; the ray path matrix M used in the computation of s_μ from s_b is the one that was correct for slownesses along the direction s_b. Thus, both of the points we have found so far lie off the feasibility boundary and the second point s_μ is of questionable worth because its value was also obtained in an essentially inconsistent manner.

Recall that the solution of (1), if one exists, must lie on the feasibility boundary. So we would like to use s_μ and s_b to help us find a point on this boundary that is optimum in the sense that it is as consistent as possible (*i*) with the ray path matrix M, (*ii*) with the measured traveltimes t, and (*iii*) with the feasibility constraints. The fact that traveltime error may be reduced by moving in the direction of s_μ may still give us an important clue about the best direction to move in the vector space, *i.e.*, we may want to move in the direction $s_\mu - s_b$ but perhaps we should stop before we arrive at s_μ. How far should we move in this direction?

If we consider Figure 1, we are reminded that the feasible region is convex. Therefore, there may exist a point s_l between the points s_μ and s_b that is closer to the feasible region than either of the two end points. If we could find this point s_l and then scale up to the point in the same direction lying on the feasibility boundary, then we have found s_f in the figure. In principle, it is possible to find the point on the line $s_\mu - s_b$ closest to the feasibility boundary. However, it is much easier to compute the feasibility violation number $N_M(s)$. As we move in the direction $s_\mu - s_b$ from s_b, we generally find that this number achieves a minimum value at some intermediate point. This point of minimum $N_M(s)$ is the point s_l in the figure.

It is not hard to prove that all three of the points s_b, s_μ, and s_f are distinct unless we have found an exact solution to the inversion problem. Our conclusion is that $s_\mu = s_b$ if and only if s_b is an exact solution of (1). This same conclusion is reached by noting that the absolute minimum (*i.e.*, zero) of the objective function (24) is attained only when $s = s_b$ and $Ms = t$. From previous arguments, we also know that the least-squares solutions s_μ and s_b never lie on the feasibility boundary unless the corresponding slownesses are exact solutions of (1). So unless we have already solved the problem, these three points form a triangle and the size of the triangle gives us an estimate of how far we are from a solution.

These ideas have all been repeatedly confirmed in a large number of reconstructions on synthetic examples. In Figure 2, three examples of *typical* results (in the middle) obtained using our new reconstruction algorithm on a structure with a low speed anomaly (20%, 50%, or 100% lower than background) on top and with a high speed anomaly (20%, 50%, or 100% higher than background) on the bottom are compared with the *best* results of a standard damped least-squares algorithm (on the left) and with the target model (on the right). The reconstructions were performed on a model structure with 8×16 cells using 320 rays – including 256 rays from left to right and 64 rays from top to bottom. The most startling example (not shown) is one in which the wave speed contrasts are 100%. Then, we find that the standard damped least-squares algorithm becomes so unstable after about 20 iterations that the results become singular, while our new algorithm is completely stable and produces the result shown in Figure 2 after about 10 iterations — and also after hundreds of iterations, if we force it to make a minimum percentage (say 1-10%) correction at every step (otherwise the method converges quite rapidly). Our algorithm has been checked on many other synthetic examples and also on real seismic and electromagnetic data, and it has always been found to converge to a reasonable solution set.

This visual comparison is not really "fair" to our method in the sense that we have chosen *typical* results from the convergence set for our new algorithm while we compare them to the *best* results for the standard algorithm. We can make this comparison for these synthetic examples because we know the correct answer and therefore know when the standard algorithm starts to diverge. In all examples shown, the standard damped least-squares method produces its best results in just a few iterations and then wanders away from it — producing better and better fits to the traveltime data, but worse and worse fits to the target model.

For real problems with high contrasts and noisy data, the damped least-squares method does *not* converge and we never know when to terminate the iteration sequence. By contrast, our new algorithm converges quickly (in 15-20 iterations) to a solution in its convergence set (*i.e.*, not a single point, but a small region of the slowness vector space with qualitatively very similar models). Stable iteration to such a convergence set is the most that could be expected when the traveltime data have errors and are therefore inconsistent.

Superimposed on the slowness models of Figure 2 are examples of the bent ray paths used either in the reconstructions, or in the forward modeling. The ray tracing method used in the reconstructions is based on a Fourier sine series expansion of the graph of the ray and a simplex method for finding the optimum coefficients of the expansion.[5] This method has been found to be as fast and as accurate as a finite difference method[6] for traveltime computation using 100 times as many cells in the model. Note the strong ray bending around the slowest regions in the models.

We have shown that Fermat's principle plays an essential role in wave speed reconstruction via traveltime tomography. Not only does this principle determine the ray paths once a slowness vector is given, but it also determines which slowness vectors are feasible and nonfeasible. Furthermore, it has provided the insight needed to find a new, stable iterative reconstruction algorithm for nonlinear traveltime tomography.

ACKNOWLEDGMENTS

Work 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 Engineering Research Program and by the Earth Sciences Department Institutional Research and Development Program. This work was also supported in part by the Air Force Office of Scientific Research though grants AFOSR 85-0017 and AFOSR 86-0352.

REFERENCES

1. M. Born and E. Wolf, *Principles of Optics* (Pergamon, London, 1959), pp. *xix-xx*, 127-129.
2. J. G. Berryman, *IEEE Trans. Geosci. Remote Sensing* **27**, 302 (1989).
3. D. W. Marquardt, *J. Soc. Indust. Appl. Math.* **11**, 431 (1963).
4. G. T. Herman, *Image Reconstruction from Projections – The Fundamentals of Computerized Tomography* (Academic, New York, 1980), pp. 1-18, 103-106.
5. T. J. Yorkey and J. G. Berryman, LLNL UCRL-100770, March, 1989.
6. J. Vidale, *Bull. Seismol. Soc. Am.* **78**, 2062 (1988).

FIGURE CAPTIONS

Fig. 1. Diagram to illustrate the key points in the new nonlinear traveltime tomography algorithm. The variable z is the wave slowness s weighted by the square root of the modified coverage matrix D [see Eq. (17)]. The axes are the weighted slownesses for any two cells (j and k) in the model. Point z_b is the initial value for the next step of the iteration scheme. Point z_μ is the solution of the damped and weighted least-squares problem. Point z_l is a linear combination of z_b and z_μ chosen because it has the smallest number of feasibility violations. Point z_f is the unique point on the feasibility boundary in the same direction as z_l.

Fig. 2. Three examples comparing the slowness results (from left to right) of a standard damped least-squares reconstruction with the results of our new algorithm and with the exact target solution. From top to bottom, the examples have anomalies that are 20%, 50%, and 100% compared to the background. In each case the top anomaly is slower than the background while the bottom anomaly is faster. Superimposed on the reconstructions and on the target images are samples of some of the bent ray paths either used in the reconstructions or obtained in the forward calculations of the traveltime data.