PAPER G

CROSSWELL TRAVELTIME TOMOGRAPHY USING DIRECT AND REFLECTED ARRIVALS: PART 1: THEORY AND IMPLEMENTATION

Mark A. Van Schaack

ABSTRACT

In past STP consortia meetings I have outlined several approaches for integrating crosswell reflection imaging and traveltime tomography. The basis of these approaches has been to obtain reflection traveltimes from the processed wavefield data and to invert these traveltimes simultaneously with direct arrival traveltimes. In all these approaches I assumed the reflector depths were known or were solved through an iterative method of successive approximations using traveltime tomography and XSP-CDP mapping. This year I have modified my scheme for Crosswell Reflection Traveltime Tomography (CRTT) so that the 2-D velocity field and reflector depths are solved for simultaneously in a single traveltime inversion.

I present my inversion scheme in two papers. In this paper, Part 1, I discuss the theory and implementation of a combined direct and reflected arrival traveltime inversion. In Part 2, found in the next paper, I present several synthetic examples of this combined inversion and a field example using data collected from the McElroy field in west Texas.

INTRODUCTION

In previous years I have described methods for integrating traveltime tomography and reflection imaging (Van Schaack, 1994; Van Schaack and Lazaratos, 1993). The basis of these methods has been to extract traveltimes from the reflection events and to invert those simultaneously with direct arrival traveltimes to obtain a velocity model which is consistent with both. One fundamental shortcoming of these methods is that the depths of the reflectors, a function of offset between wells, are assumed to be known in the traveltime inversion. Unfortunately the actual reflector depths are never known *a priori* and the reflector depths used in these inversion have been only estimates.

One approach for obtaining better estimates of the reflector depths is to use interpretations of the crosswell reflection image to update the depths (Van Schaack, 1994). This requires an iterative approach of traveltime tomography with reflector depth estimates and XSP-CDP mapping using the new velocity field to obtain better estimates. Unfortunately, the traveltime tomography and XSP-CDP mapping must be repeated for each iteration making this scheme inefficient.

I have revised my approach to CRTT to explicitly solve for reflector depths in the inversion of direct and reflected arrival traveltimes. Reflector depths have been shown to be well resolved in synthetic studies using typical crosswell acquisition geometries (Bube and

Langan, 1995). In fact, these synthetic studies predict errors in reflector depths on the order of only *inches*. I have confirmed this prediction with several synthetic studies discussed in Part 2 found in the next paper.

In this paper I describe the theory and implementation of my combined direct and reflected arrival traveltime inversion program where I simultaneously solve for a 2-D velocity field and reflector depths. I begin with a description of the basic theory and setup of the inverse problem followed by a more specific discussion of the parameterizations used in my inversion code. Next, I describe the forward modeling used in the inversion: raytracing and traveltime calculations. Finally, I discuss the details of the solution of the combined direct and reflected arrival traveltime inversion.

DIRECT AND REFLECTED ARRIVAL INVERSION: BASIC THEORY

Seismic traveltime tomography is typically a non-linear problem due to large variations of velocity within the Earth. Traveltime tomography is non-linear in the sense that the velocity field is required to calculate the raypaths which are used to solve for the velocity field. One procedure used to solve the non-linear tomography problem is the Gauss-Newton method (Marquardt, 1963). In the Gauss-Newton method the non-linear problem is linearized by using an initial estimate of the model parameters. The solution of the linearized problem is an updated estimate of the model parameters. This procedure is performed in an iterative fashion until some convergence criteria is finally reached; usually convergence is assumed when the difference between the updated solution and the initial estimate becomes negligible.

To describe how reflection traveltimes and reflector depths are added to the tomography problem I will focus on a single linearized step:

$$\mathbf{A}\,\mathbf{x} = \mathbf{t} \tag{1}$$

where **A** is the traveltime derivative matrix, **x** is the vector of model parameters, and **t** is the vector of observed traveltimes. The dimensions of **A** are m x n, where m is the number of traveltime observations and n is the number of parameters being calculated. Usually the matrix **A** is overdetermined so the problem is solved using least squares or some other norm. Also, since an estimate of **x** is used to define **A**, the problem typically is defined in terms of residual traveltimes ($\Delta\mathbf{t}$) and perturbations to the parameters ($\Delta\mathbf{x}$). To keep the explanations simple I limit the discussion in this section to the setup of Eqn. 1. Later in this paper I describe the solution of the actual inverse problem.

The exact form of **A** and **x** are related to how the model is parameterized. In order to be more specific about the setup of matrix **A** I use a simple model parameterization. One of the simplest ways to parameterize the slowness model is as rectangular constant slowness (1/velocity) cells. Figure 1 shows a simple numbering scheme for this constant slowness parameterization and how the values are stored in vector **x**. The values of matrix **A** are the derivatives of traveltime with respect to the model parameters, in this case slowness only. Each row of **A** corresponds to an individual raypath of a source-receiver pair. Each column in that row contains the contribution of a particular cell to the traveltime of that raypath. Only cells crossed by a ray have non-zero traveltime derivatives in the constant slowness cell parameterization and the value of the traveltime derivative is the length of the ray in that cell.

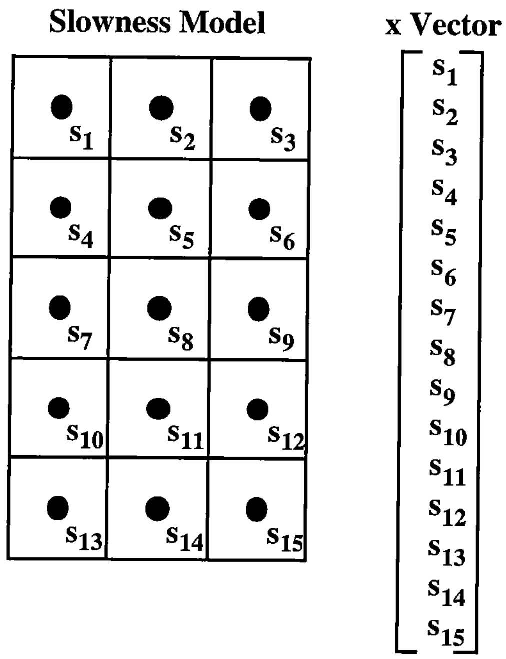

Figure 1: A simple parameterization of the slowness model and how it is stored as a vector. The slowness is defined for each node which is located in the center of the constant slowness cell.

Direct Arrival Traveltime Tomography

Direct arrival traveltime tomography is the most common method of processing cross-well data. Typically the only parameters being determined are the slowness (1/velocity) values of the model. By substituting **s** (slowness) for **x** we can rewrite Eqn. 1 as:

$$\mathbf{A}\,\mathbf{s} = \mathbf{t} \tag{2}$$

The number of elements in **s** equals the number of slowness elements (n) in the model and the number of elements in **t** (m) equals the number of traveltime observations.

Expressing Eqn. 2 in matrix form further clarifies the setup of the problem:

$$\begin{bmatrix} \frac{\partial t_1}{\partial s_1} & \frac{\partial t_1}{\partial s_2} & \frac{\partial t_1}{\partial s_3} & \cdots & \frac{\partial t_1}{\partial s_m} \\ \frac{\partial t_2}{\partial s_1} & \frac{\partial t_2}{\partial s_2} & \frac{\partial t_2}{\partial s_3} & \cdots & \frac{\partial t_2}{\partial s_m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial t_n}{\partial s_1} & \frac{\partial t_n}{\partial s_2} & \frac{\partial t_n}{\partial s_3} & \cdots & \frac{\partial t_n}{\partial s_m} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \cdots \\ s_m \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ \cdots \\ t_n \end{bmatrix} \tag{3}$$

From the system of equations shown in Eqn. 3 the linear equations corresponding to the various source-receiver traveltime observation can be more easily seen. The variable $\partial t_i / \partial s_j$ is the length of raypath i in cell j. The unknown s_j is the slowness of the j'th cell and t_i is the traveltime observation corresponding to the i'th raypath. Each single equation in the system of equations corresponds to a single traveltime observation. For example, the first raypath, seen in row 1 of the **A** matrix, defines the linear equation

$$\frac{\partial t_1}{\partial s_1} s_1 + \frac{\partial t_1}{\partial s_2} s_2 + \frac{\partial t_1}{\partial s_3} s_3 + \ldots + \frac{\partial t_1}{\partial s_m} s_m = t_1 \tag{4}$$

Now that the setup of direct arrival traveltime tomography has been defined I will describe how reflection traveltimes can be added to the inversion.

Combined Direct & Reflected Arrival Tomography

In my past implementations of CRTT I assume the locations of the reflectors are known so the only parameters solved for in the inversion are the model slowness values. The addition of the reflection traveltime data to the inversion in this case is very simple. No parameters are added to the **x** vector so **s** = **x** as in Eqns. 2 & 3. The only change is the addition of a row to **A** and **t** for each reflection traveltime and its associated raypath. Since the reflector location is known the only parameters that can influence the reflection traveltime are the slowness values of the cells through which the raypath passes. So, for each reflection traveltime, a linear equation in the form of Eqn. 4 can be added to the linear system of equations. Once the linear system is defined the inversion can be performed in the same manner as if only direct arrivals were being used.

To solve for model slowness and reflector depths simultaneously both **A** and **x** must be modified. Although I use a more general description of the reflectors in the CRTT program I use flat reflectors in this discussion to describe how Eqns. 3 & 4 are modified. Flat reflectors can be fully described by one parameter, depth. I use the notation Zr_j to signify the depth of reflector r_j. For each reflector that traveltime picks are available an additional parameter is added to **x**. Corresponding to each parameter added to **x**, an additional column is added to **A**. In Eqn.5 I show one way to modify the linear set of equations shown in Eqn. 3 to include reflection traveltimes and solutions for reflector depths.

The **A** matrix in Eqn. 5 has been assembled with rows related to direct arrival traveltimes placed in the top part and rows related to reflected arrival traveltimes placed in the bottom part. A new term has been introduced in the **A** matrix, $\partial t_i / \partial Zr_j$. This term is the derivative of the traveltime with respect to changes in the reflector depth. In Eqn. 5 I assume a parameterization where slowness and reflector depths are decoupled. For this reason $\partial t_i / \partial Zr_j$ is always zero for direct arrival raypaths. I also ignore raypaths which contain more than one reflection so there is only one non-zero reflector derivative for each reflected raypath.

$$\begin{array}{c} \text{Direct arrival} \\ \text{raypaths} \\ \\ \\ \\ \text{Reflected arrival} \\ \text{raypaths} \end{array} \quad \begin{bmatrix} \frac{\partial t_1}{\partial s_1} & \frac{\partial t_1}{\partial s_2} & \frac{\partial t_1}{\partial s_3} & \cdots & \frac{\partial t_1}{\partial s_m} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & 0 & \cdots & 0 \\ \frac{\partial t_{n1}}{\partial s_1} & \frac{\partial t_{n1}}{\partial s_2} & \frac{\partial t_{n1}}{\partial s_3} & \cdots & \frac{\partial t_{n1}}{\partial s_m} & 0 & \cdots & 0 \\ \frac{\partial t_{n2}}{\partial s_1} & \frac{\partial t_{n2}}{\partial s_2} & \frac{\partial t_{n2}}{\partial s_3} & \cdots & \frac{\partial t_{n2}}{\partial s_m} & \frac{\partial t_{n2}}{\partial Zr_1} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial t_n}{\partial s_1} & \frac{\partial t_n}{\partial s_2} & \frac{\partial t_n}{\partial s_3} & \cdots & \frac{\partial t_n}{\partial s_m} & 0 & \cdots & \frac{\partial t_n}{\partial Zr_r} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \cdots \\ s_m \\ Zr_1 \\ \cdots \\ Zr_r \end{bmatrix} = \begin{bmatrix} t_1 \\ \cdots \\ t_{n1} \\ t_{n2} \\ \cdots \\ t_n \end{bmatrix} \tag{5}$$

<u>Summary of Basic Theory</u>

In this section I have described the basic setup of a linear set of equations for direct arrival tomography and combined direct and reflected arrival tomography. The exact setup depends on the parameterization used in the inversion and on any regularization added to the system. In the next several sections I describe in detail the parameterization and regularization used in my inversion code and the approach I use for solving the non-linear inversion problem.

PARAMETERIZATION OF THE MODEL

A number of issues influence the choices made in parameterizing the CRTT inversion. One issue is accuracy. For example, describing velocity as a 1-D function of depth leads to inversion errors for data sets collected in areas where lateral velocity variations are present. Parameterizing the problem allowing velocity to vary in 2-D may reduce inversion errors but leads to the issue of problem size. Invariably, increasing the number of parameters used to define the model will increase the size of the inversion. Potentially this may lead to a problem which is overparameterized. In addition, a large inverse problem may exceed the capacity of the computer hardware in the areas of memory or speed.

In my current CRTT program I have focused most heavily on parameterizing the model to accommodate the features found in STP data sets and the limitations of STP computer hardware. The primary features I have attempted to accommodate in the current version of my traveltime inversion are:

- 2-D isotropic velocity variations
- Deviated source and receiver wells
- Dipping, non-linear, and/or discontinuous reflector interfaces

In later versions I hope to accommodate additional parameters such as anisotropy and depth statics.

Slowness

In my current version of CRTT I parameterize the model using constant slowness cells. I do this for two reasons: to maintain simplicity and to minimize memory requirements. Simplicity is maintained since the inverse problem is defined using the setup shown in Eqn. 5. The traveltime derivative, $\partial t_i / \partial s_j$, can be calculated in a straightforward manner using this parameterization since it is simply the length of the i'th ray in the j'th cell.

The primary reason I use the constant slowness cell parameterization is that the amount of RAM memory required to solve the inverse problem is minimized. This reduction in memory results from the way the **A** matrix is stored. Any one ray will only intersect a small number of cells in the slowness model. Rather than wasting space storing all the zero traveltime derivatives the **A** matrix is preserved in a compact form where only the non-zero traveltimes derivatives are stored. For example, a horizontal ray traveling from one side to the other through the Fig. 1 model would have 12 zero and 3 non-zero traveltime derivatives. Only the 3 non-zero derivatives (and their indices) need be saved.

Another useful way to parameterize the slowness model is by using nodes similar to those in Fig. 1 but using them to define the value of the slowness at the corners of a cell. By interpreting the slowness within the cell this parameterization yields a continuous function of slowness in 2-D. One advantage of a continuous function of slowness is that it is much more stable in which to trace rays. Unfortunately, the memory requirements using this parameterization are approximately double those using the constant slowness cell parameterization. This increase results from the more complex traveltime derivatives that result from this parameterization. Instead of the 3 non-zero derivatives saved in our previous example we would be required to save 6 traveltime derivatives. This is because the slowness at any point in the model is a function of at least 3 nodes.

While the constant slowness cell parameterization requires much less memory to store the **A** matrix it poses a problem of how to trace rays through it effectively. I explain later in this paper the details of the raytracing and traveltime calculations. It should be restated that the primary reason I use the constant slowness cell parameterization in the inverse problem is to reduce memory requirements. Currently I am solving problems in synthetic models of approximately 650 ft x 1000 ft using 10 foot cells. Using approximately 60,000 direct and reflected arrival traveltimes results in a problem that requires almost 100 Mb of memory. The almost negligible errors found in the results of the synthetic studies detailed in Part 2 of my next paper suggest that the constant slowness cell parameterization is sufficiently accurate for these problems. Doubling the size of these problems by using a more accurate model parameterization would make them too large to solve using STP computers.

Reflectors

In the basic theory section of this paper I outlined the setup of the linearized inverse problem. This setup included solutions for reflector location defining the reflectors using a single parameter, depth. Unfortunately, this parameterization limits our model to simple flat reflectors. In my CRTT inversion code I have implemented a more general description of the reflectors. Each reflector is defined at a constant interwell interval across the slowness model. The interval between defined reflector locations is a variable set by the user.

Between defined reflector locations the reflector is described using a cubic spline curve. The advantage of using the cubic spline curve is that the 1st derivative of depth with respect to interwell offset is continuous. This improves the ability of the raytracer to link source and receivers for reflected raypaths.

Discontinuous reflectors are handled as independent reflecting horizons. The starting estimate for each reflecting horizon where traveltime picks are obtained is flat, continuous, and ranges from one side of the slowness model to the other. The actual extent of the reflector is determined following the inversion and is based on the ray coverage along the reflector. This approach for handling a discontinuous reflector does not provide any information about how the discontinuous reflector segments may be associated with each other. As with surface seismic imaging, determining fault locations and throw would be done in the interpretation phase.

FORWARD MODELING — RAYTRACING & TRAVELTIMES

I use an initial value raytracer and solve the two-point problem using an iterative approach. One drawback of the discrete constant slowness cell parametrization of the model is that it suggests raytracing by launching rays and calculating changes in trajectory at cell boundaries using Snell's Law. The resulting rays tend to be undesirable since they can make sharp changes in trajectory at each cell boundary which is unphysical in light of the assumptions made in justifying the use of ray theory in the first place. Unfortunately, as mentioned previously, parameterizing the slowness model using gradients leads to inverse problems which are too large to solve. To overcome this dilemma I have adopted a hybrid parameterization of slowness. An alternative to this approach would have been to limit the number of reflectors used in any given inversion.

I implement my hybrid parametrization of slowness starting with a model where slowness values are defined at node points on an evenly spaced grid. This gridding scheme is identical to that shown in Fig. 1. The difference is in how the value of slowness between nodes is allowed to vary. For raytracing, where better results are obtained with a continuous slowness field, I interpolate the slownesses as needed using bilinear interpolation. To calculate traveltimes and the traveltime derivatives required to set up the inverse problem I assume each node defines the center of a constant slowness cell.

While this hybrid scheme might seem fatally inconsistent, it roughly parallels the philosophy used in straight ray tomography. In straight ray tomography the calculation of the raypaths and the discretization of the slowness field (for purposes of setting up the inverse problem) are completely decoupled. Likewise, in this hybrid parameterization I obtain the most reliable raypaths using the best parameterization and then, ignoring the origin of the raypaths, use them in a model parameterization ideal for the inverse problem. Results of synthetic tests are presented in the next paper which support my opinion that this hybrid approach of model parameterization is effective and sufficiently accurate to obtain good results.

Initial Value Raytracing

I use a modified version of the initial value raytracer described by Harris (1992). This raytracer calculates raypaths in smoothly varying heterogeneous media. Raytracing is performed in a piece-wise fashion using the Runge-Kutta method to solve the ordinary differential equations defining the raypath.

I have modified the original code described by Harris to obtain slightly better performance. The original version calculates the slowness gradient at each point along the raypath from interpolated slowness values. To improve speed and efficiency I calculate the slowness gradient in the x and z directions at each node point prior to raytracing. I then obtain the gradient when required by interpolating the precalculated gradients. Bilinear interpolation is used both to interpolate slowness values to calculate the gradients and to interpolate the gradients themselves.

Calculating Reflected Raypaths

Reflected raypaths are calculated using the same initial value raytracing code as direct raypaths. Raypaths are calculated first for all direct arrivals then for all reflected arrivals, each reflector in turn. To obtain a reflected raypath for a particular reflector the ray is initially launched in the same fashion as a direct ray. The ray is monitored as it is calculated step by step until it crosses the reflecting horizon.

Once the ray crosses the reflecting horizon I calculate the exact reflection point. The reflection point calculation uses an iterative approach. First points on the reflector are obtained using the cubic spline approximation which bracket the possible intersection point. The reflector is assumed to be linear between these points and an intersection point with the raypath is found. This intersection point is used to calculate a new point on the reflector which replaces one of the two previous reflector points so the possible reflection point remains bracketed. This routine is performed iteratively until the intersection point is found to a user-defined tolerance.

The intersection point of the ray and the reflector is used to calculate the local incidence angle of the ray on the reflector. This angle is computed using the local raypath trajectory and a derivative function of the cubic spline curve to determine the local reflector dip. From this information a reflected trajectory is computed and the ray is re-launched. The ray then proceeds until it intersects the well or leaves the model.

The Two-Point Problem

The two-point problem is solved using a standard iterative approach which is a derivation of a code described by Langan et al. (1985). Rays are traced by receiver gather from receiver to source. I do this partly for philosophical reasons, to parallel the acquisition method, and partly to aid the calculation of traveltime residuals.

The first step in solving the two-point problem is to shoot a reference fan for the receiver gather under consideration. The rays are shot so as to provide a relatively uniform coverage at the source well. For each source-receiver pair for which a raypath is desired the algorithm finds rays from the reference fan which bracket the pair. Attempts are made to connect the source and receiver using the reference fan data as a starting point and a derivative-based search algorithm. Rays are shot from receiver to source until a ray is "captured" within a user defined "capture tolerance" or until a user-specified number of iterations is reached and the process is aborted.

It is possible to find more than one pair of reference rays bracketing the source. This will occur when triplications are present. To discriminate between these various rays the user sets a flag indicating whether rays of minimum traveltime or minimum path length are desired. A minimum path length raypath corresponds to the "energetic" direct arrival ray. The minimum traveltime raypath corresponds to the "first arrrival" raypath which may be associated with a head wave. Raypaths for all possible source-receiver combinations are calculated and the desired path is stored.

Calculating Traveltimes and Traveltime Derivatives

Once the raypath is obtained its traveltime is calculated using the constant slowness cell parameterization. The traveltime of raypath i, t_i, is calculated using this equation,

$$t_i = \sum_{j=1}^{m} l_{ij} s_j \tag{6}$$

In Eqn. 6 l_{ij} is the length of the i'th ray in the j'th cell and s_j is the slowness of the j'th cell.

The traveltime derivative, $\partial t_i / \partial s_j$, for the constant slowness cell parameterization is simply the length of the ray in the cell, l_{ij} from Eqn. 6. I currently use an estimate of this which is accurate to 1/10 of a foot. To obtain this estimate I resample the ray in 1/10 ft increments and count the number of segment endpoints which fall in each cell. The advantage of this approximation is speed and simplicity.

Calculating Reflector Depth Derivatives

Traveltime derivatives with respect to the reflector depth parameters, $\partial t_i / \partial Zr_j$, are calculated within the raytracing process. To calculate these derivatives I use the procedure derived by Bishop et al. (1985) with a modification to handle both upgoing and downgoing reflections. The expression for the traveltime depth derivative of reflector parameter j when the reflection point, (x_R, z_R), is located (in offset) between $x(r_j)$ and $x(r_{j-1})$ is

$$\frac{\partial t_i}{\partial Zr_j} = 2s(x_R, z_R) \cos\beta \cos\theta \left(\frac{x_R - x(r_{j-1})}{x(r_j) - x(r_{j-1})} \right) \tag{7}$$

In this equation, $s(x_R, z_R)$ is the local slowness at the reflection point, β is the dip of the reflector from the horizontal, and θ is the local angle of incidence of the raypath on the reflector.

SOLVING THE INVERSE PROBLEM

To solve the tomographic inverse problem I use a continuation approach described by Bube and Langan (1994). The basis of this approach is to add regularization to the inverse problem in the form of smoothing penalty terms. The non-linear system of equations is then solved with penalty weights fixed using the Gauss-Newton (G-N) method. After the non-linear problem is solved the penalty weights are relaxed and the non-linear problem is solved again. This procedure is repeated until an optimum solution is reached. In this approach a "continuation step" refers to the solution of a non-linear problem through a series of linearized steps while holding the smoothing penalties constant. A "G-N step" is the solution of a single linearized system of equations within the larger continuation step.

There are several advantages to this approach. One is that it provides regularization to the inverse problem. A second, related advantage, is that the optimum weights for the regularization do not need to be known *a priori.* A range of weights is run during the inversion so that initial iterations are dominated by the smoothing terms and later inversions are dominated by the data. Although a wide variety of smoothing terms are possible I have found, like Bube and Langan (1994), that penalty terms for the horizontal and vertical first spatial derivatives of slowness work well. Additionally, in my inversion, I have found that I must add a smoothing penalty term for the second derivative of the reflector depth with respect to offset. In the next sections I will further discuss the details of these penalty terms.

Regularization — Slowness Field

In my crosswell inversion I use smoothing penalty terms for horizontal and vertical spatial derivatives of slowness in the same manner as Bube and Langan (1994). Typically I weight the horizontal derivatives about four times heavier than the vertical derivatives although this is a user-defined variable. This preferential weighting is consistent with the lower resolution of the typical crosswell inversion in the horizontal direction. Also, there is typically less lateral variation in slowness than vertical variation in many geologic settings.

Regularization — Reflector Geometries

Because I allow the possibility of reflectors which do not extend completely from one well to the other there must be regularization added to the solution of the reflector depths. The main reason regularization is required is because of the cubic spline parameterization of the reflectors. If, for example, a reflector only extends half way between the wells only about half of the reflection depth derivatives for that reflector will be non-zero. The reflector depth parameters for which the depth derivatives are always zero lie in the null space of the model. The least-squares solver used in my program will not provide any perturbations to those parameters.

In a CRTT inversion containing discontinuous reflectors there will exist adjacent reflector points where one is updated during the inversion and the other remain unmodified since it lies in the null space of the inversion. The unfortunate side effect of this is that the cubic spline solution across this discontinuity will vary wildly. Even worse, these wild variations will not be confined to the area of the discontinuity. Once a reflector geometry acquires these sharp oscillations it becomes difficult to link rays through this reflector.

To solve this problem I have added a smoothing penalty term to the inversion to minimize second derivative variations of the reflector depth with respect to horizontal offset. The effect of this penalty term on discontinuous reflectors is to preserve the slope of the reflector when no other information is present. Like the smoothing penalty terms applied to the slowness field, I decrease the weight of this reflector penalty term at the beginning of each continuation step. I have found in synthetic examples shown in my next paper that the weight of the reflector second derivative penalty term need not be very strong to maintain the stability of the reflector solutions.

Problem Setup

The least-squares solution of the linearized G-N step is the minimum of

$$\|\mathbf{A}\delta\mathbf{x} - \delta\mathbf{t}\|^2 + c_h\|\mathbf{D}_h\mathbf{x}\|^2 + c_v\|\mathbf{D}_v\mathbf{x}\|^2 + c_z\|\mathbf{D}_z\mathbf{x}\|^2 \tag{8}$$

In Eqn. 8, **A** is the matrix of derivatives of the traveltime with respect to each of the parameters of the model. In this case, **A** contains derivatives of traveltimes with respect to the slowness in each cell and derivatives of traveltime with respect to the depth of the reflectors. The vector $\delta\mathbf{t}$ contains the traveltime residuals, e.g. the differences between traveltimes calculated using the starting model $\mathbf{x}^k$ and traveltimes observed in the experiment:

$$\delta\mathbf{t} = \mathbf{t}(\mathbf{x}^k) - \mathbf{t}_{obs} \tag{9}$$

The Eqn. 9 the vector, $\delta\mathbf{x}$, is the perturbation to the starting model $\mathbf{x}^k$ required to minimize Eqn. 8. The result of the k'th G-N step is the $k+1$'th updated model, $\mathbf{x}^{k+1}$, where

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \delta\mathbf{x} \tag{10}$$

The remaining terms in Eqn. 8 are the smoothing penalty terms. The constants c_j are the weights of the various penalty terms relaxed prior to the beginning of each new continuations step. The matrix operators $\mathbf{D}_j$ are used to calculate the derivatives used in the smoothing penalties and are constant through the entire inversion.

The final version of the linearized system of equations solved in the CRTT inversion is

$$\begin{bmatrix} \cdots & \frac{\partial t_i}{\partial s_j} & \cdots & 0 \\ \cdots & \frac{\partial t_i}{\partial s_j} & \cdots & \frac{\partial t_i}{\partial Zr_k} \\ \cdots & \sqrt{c_h}\mathbf{D}_h & \cdots & 0 \\ \cdots & \sqrt{c_v}\mathbf{D}_v & \cdots & 0 \\ \cdots & 0 & \cdots & \sqrt{c_z}\mathbf{D}_z \end{bmatrix} \begin{bmatrix} \cdots \\ \delta s_m \\ \cdots \\ \delta Zr_r \end{bmatrix} = \begin{bmatrix} \delta t_{dir} \\ \delta t_{ref} \\ \sqrt{c_h}\mathbf{D}_h(s_m - C_{start}) \\ \sqrt{c_v}\mathbf{D}_v(s_m - C_{start}) \\ -\sqrt{c_z}\mathbf{D}_z Zr_r \end{bmatrix} \tag{11}$$

In Eqn. 11 the constant C_{start} provides the user the ability to drive the derivatives toward an input slowness model or towards flatness (when C_{start} is homogeneous). I typically initialize C_{start} to be homogeneous and equal the average slowness of the data set.

I have broken up the **A** matrix into 5 distinct parts in Eqn. 11. Each distinct part of **A** is designated by a row. The first row represents derivatives related to direct arrivals only. Notice in the first row that the derivatives of traveltime with respect to reflector depths are zero as you would expect for non-reflected raypaths. The second row represents all reflected arrival raypaths which have both slowness and depth derivatives. The next three rows are the smoothing penalty terms.

Finally, to keep the penalty terms properly balanced they are normalized by the Frobenius norm of subsets of the **A** matrix. The values c_h and c_v are calculated by taking the current penalty weight and scaling it by the square of the Frobenius norm of the traveltime derivatives with respect to slowness. The value c_z is scaled by the square of the Frobenius norm of the traveltime derivatives with respect to depth.

CONCLUSIONS

In this paper I present an approach to reflection and direct arrival traveltime tomography. This approach solves for a 2-D isotropic slowness model and reflector depths parameterized using cubic splines. The slowness model and reflector depths are calculated by minimizing the difference between both direct arrival traveltimes and reflected arrival traveltimes to selected reflectors, and traveltimes calculated tracing rays through the model.

The CRTT program described in this paper uses a continuation approach for solving the non-linear tomography problem. Smoothing penalty terms are added to the inversion providing regularization for both the slowness and reflector solutions. The inverse problem is solved in a series of continuation steps where the smoothing penalty terms are slowly relaxed. Each continuation step consists of a number of linearized Gauss-Newton steps solving the non-linear inverse problem while the weights of the smoothing penalty terms are held constant. Solutions of early continuation steps provide a smooth slowness model and nearly linear reflectors. In later solutions the slowness model is allowed more 2-D variations while the reflectors are allowed to model increasingly non-linear fluctuations.

In this paper I have provided the background theory of a combined direct and reflected arrival traveltime inversion and details of my implementation of this theory. In my next paper I provide the results of several synthetic studies and one field data study.

ACKNOWLEDGEMENTS

I would like to thank Ken Bube and Robert Langan for providing me with their traveltime inversion code and helpful guidance. I would also like to thank Jesse Costa for his invaluable assistance in providing me insights to the inversion process. I would finally like to thank Jerry Harris and the sponsors of the STP consortia for providing support for this work

REFERENCES

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

Bube, B. P., and Langan, R. T., 1995, Resolution of crosswell tomography with transmitted and reflected traveltimes: STP-**6**, Paper I.

Harris, J. M., 1992, Initial value raytracing in smoothly varying heterogeneous media: STP-**3**, Paper I.

Langan, R.T., Lerche, I., and Cutler, R.T., 1985, Tracing of rays through an heterogeneous media: Am accurate and efficient procedure: Geophysics, **50**, September.

Lazaratos, S. K., 1993, Cross-well reflection imaging: Ph.D. thesis, Stanford University.

Lazaratos, S. K., Harris, J. M., Rector, J. W., and Van Schaack, M. A., 1995, High-resolution crosswell imaging of a west Texas carbonate reservoir: Part 4—Reflection imaging: Geophysics, **60**, 702–711.

Marquardt, D. W., 1963, An algorithm for least-squares estimation of non-linear parameters: Soc. of Industr. Appl. Math., **11**, 431–441.

Van Schaack, M. A. and Lazaratos, S. K., 1993, An approach to interval velocity analysis for crosswell reflection imaging using reflection traveltimes: STP-**4**, Paper F.

Van Schaack, M. A., 1994, Integrated crosswell imaging: Reflection tomography—Synthetic examples: STP-**5**, Paper A.