# Structural-velocity analysis using migrated seismic data

*Jos van Trier*


## ABSTRACT

In structurally complex regions, an interactive interpretation of the migrated image is useful in determining the structural-velocity model. After the structural features are incorporated in the model, the velocities inside the structures can be determined by a gradient optimization scheme, where the objective is maximization of semblance in the migrated image. The semblance is calculated for reflection events in the prestack migrated data that correspond to reflectors picked in the interpretation. The gradients needed for the optimization are calculated by ray tracing and by determining semblance derivatives from the prestack migrated data.


## INTRODUCTION

Depth migration is necessary in structurally complex areas: both to get a well-focused image of the subsurface, and to accurately estimate the reflector depth. Since depth migration requires a 2-dimensional velocity model, more sophisticated velocity-estimation methods than traditional NMO are needed in these areas. Several 2-dimensional velocity analysis methods that deal with this problem have been introduced in the last couple of years (Al-Yahya, 1987, Fowler, 1988, Biondi, 1988, and Etgen, 1988). All these methods use the complete prestack dataset in a gradient-optimization scheme, and find a velocity model that maximizes some energy norm in the data. Fully automated, they are well-suited for areas with smooth lateral-velocity variations and long sequences of continuous reflectors.

When there are only a limited number of strong, continuous events in the data, and when there are large velocity contrasts between structures (such as in regions with salt intrusions and complex faulting), fully automated velocity-analysis methods may be hard to use. Not only will gradient calculations be expensive if gradients

have to be determined for each possible reflector point, but, if data is included that does not provide consistent velocity information, the gradient calculation can be inaccurate. For these applications, one would like to concentrate on strong events in the prestack data, and interactively constrain the velocity model to incorporate structural features observed in the migrated or stacked image. Several authors have recently introduced such event-driven structural-velocity estimation methods (Stork and Clayton, 1987; Landa et al., 1988; Verprat et al., 1988). However, these methods either require picking of events in the prestack data, which is cumbersome and not robust in the case of noisy data, or they do not use gradients in the optimization, making them inefficient and too expensive to apply to complicated models.

Here I present a velocity-analysis method that falls in the latter, event-driven category, but I calculate gradients that can be used in a conjugate-gradient optimization. The objective of the optimization is to maximize the semblance in the migrated image, rather than to minimize the difference between modeled and picked events. The semblance is calculated along stacking trajectories in the prestack migrated data, the shapes of which change during the optimization. The only picking involved in the method is the picking of structural boundaries in the migrated image, which takes place during the parametrization of the structural model. The parametrization is guided by geological constraints and is the same as the one I described last year (Van Trier, 1988a).

I will start with a brief overview of the method, then I will discuss the gradient calculations and optimization scheme in detail. The optimization is demonstrated with a simple example. Finally, I will discuss how to apply the method to migrated data that have been converted to the zero-offset-time domain.

## OVERVIEW

In SEP-57 (Van Trier, 1988a) I described an optimization method to determine seismic velocities in a structurally complex region. The method uses structural boundaries picked from a seismic image to parametrize a structural velocity model. The image is found by migrating the data with a smooth velocity model, where the smooth model is the result of a migration-velocity analysis. The velocities inside the structures are then modeled by 2-dimensional spline functions, the coefficients of which are determined by a tomographic inversion of traveltimes. The model parametrization allows for the explicit input of velocities and boundary locations, wherever dictated by geological or other information, such as well logs. Additional geological constraints can be applied during the inversion by damping the optimization with a "geological" damping matrix.

As presented in SEP-57, the method relied heavily on the results of the migration-velocity analysis. First, it assumed that cumulative traveltimes in the seismic data could be accurately reconstructed from the smooth model. Second, the image was

considered to be accurate enough to give a reasonable estimate of the position of the reflectors.

While I still use the same parametrization for the structural model, and the idea remains to concentrate on the events in the prestack data that correspond to the picked reflectors in the migrated image, I have modified the method to eliminate the above two assumptions. I now calculate the gradient of the objective function directly from the migrated prestack data, and I take into account reflector movement as a function of migration velocity. The objective function is the semblance in the migrated image. Semblance is calculated by stacking the prestack migrated data along a trajectory, the shape of which depends on the migration velocity. The goal of the optimization is maximization of the semblance, which is achieved by changing the migration-velocity model such that the trajectory follows the maximum energy in the prestack migrated data.

The gradient calculation consists of two parts. The first part determines how the semblance of the migrated image changes if the semblance stacking curve through the prestack data is changed. In the second part, ray tracing is used to determine how the trajectory moves when migration-velocity model parameters are perturbed. This calculation is similar to—but more complicated than—the one used in traditional tomography, where the derivatives of traveltimes with respect to model parameters are calculated. The complication arises because not only does the traveltime change with velocity, but so does the reflector point. Since the depth point is a function of the migration velocity, Fermat's principle can no longer be applied, and ray-bending effects have to be taken into account. Another complication comes from the fact that I am using migrated data instead of surface data. Extra calculations are required to convert derivatives from the unmigrated to the migrated data space. The combination of the semblance component and the ray-traced component gives the final gradient, which determines how the semblance changes if a model parameter is changed.

In the gradient calculation I have taken care not to make assumptions about the velocity model or to approximate any behavior of the gradient with its behavior in the constant velocity case. This is to ensure that the method will work in structurally complex regions where raypaths may be complicated. It will make the gradient calculation more complex and expensive than the one used by Fowler (1988) and Etgen (1988), for example, but since gradients are only calculated for a limited number of reflectors (instead of for every depth point) the overall cost of the optimization is greatly reduced. The computation time is in the order of minutes on the Convex C-1, and may approach interactive speeds on more powerful computers.

The results that are shown in this paper use depth-migrated data to calculate the gradient, but I will discuss how to apply the calculations to migrated data that have been converted to pseudo-depth or zero-offset-time domains. The advantage of working in these latter domains is that the zero-offset component of the prestack events does not migrate, meaning that the gradient calculation focuses on

the non-zero-offset behavior of the events—the part that provides the most velocity information.

## COMPARISON WITH TOMOGRAPHY

One might ask why I go through the trouble of including reflector movement in the gradient calculations, and apply the optimization to migrated data, rather than use a more traditional tomographic approach. In standard tomography, traveltime derivatives with respect to the model parameters are calculated for fixed boundaries, and unmigrated data are used in the optimization, either by picking traveltimes or by calculating semblance along traveltime curves in the prestack data.

I will now demonstrate some of the problems with this traditional approach for the simple example of a flat reflector in a constant velocity medium. The medium has a velocity of 2 km/s, the depth of the reflector is 1 km. If the reflection event is migrated with the wrong velocity, the reflector depth will be incorrect, and residual traveltime differences will exist between modeled and measured traveltimes. Now suppose that the velocity is estimated from the residuals by a tomographic inversion, where traveltime gradients are determined by ray tracing from the (mispositioned) reflector. For a constant velocity medium, the traveltime gradients and residuals can be calculated analytically, and the (constant-)velocity perturbation can be estimated in the least-squares sense by a standard least-squares method. The velocity model is then updated, the event remigrated, and the inversion repeated for newly calculated gradients and residuals.

Figure 1 shows the result of such an iterative inversion, with the initial migration velocity 2.5 km/s. I assume that the position of the reflector at each iteration is estimated by migrating the zero-offset section with the current velocity. As can be seen from Figure 1-1, the optimization converges slowly to the solution; since the gradients are calculated for the wrong reflector position, they are not able to predict correctly the traveltime residuals. The convergence can be improved considerably by including reflector depth as parameter and solving a two parameter problem (Figure 1-2), where the gradient of traveltime with respect to reflector depth is also calculated analytically. However, again the gradients are incorrect as long as the reflector depth is wrong, and several iterations are needed before the solution converges. When more than one reflector is concerned, the optimization can end up in a local minimum because of velocity-depth ambiguities (Stork, 1988). Furthermore, because depth and velocity are different physical parameters, one must be careful about scaling them correctly in the optimization; incorrect scaling might lead to unexpected results.

Figure 1-3 shows the result of inverting perturbations in the migrated data using the gradient operator that I present in this paper. The data are depth-migrated and then converted to the zero-offset-time domain by a simple scaling. Since the gradient calculations take into account the reflector movement as a function of velocity,

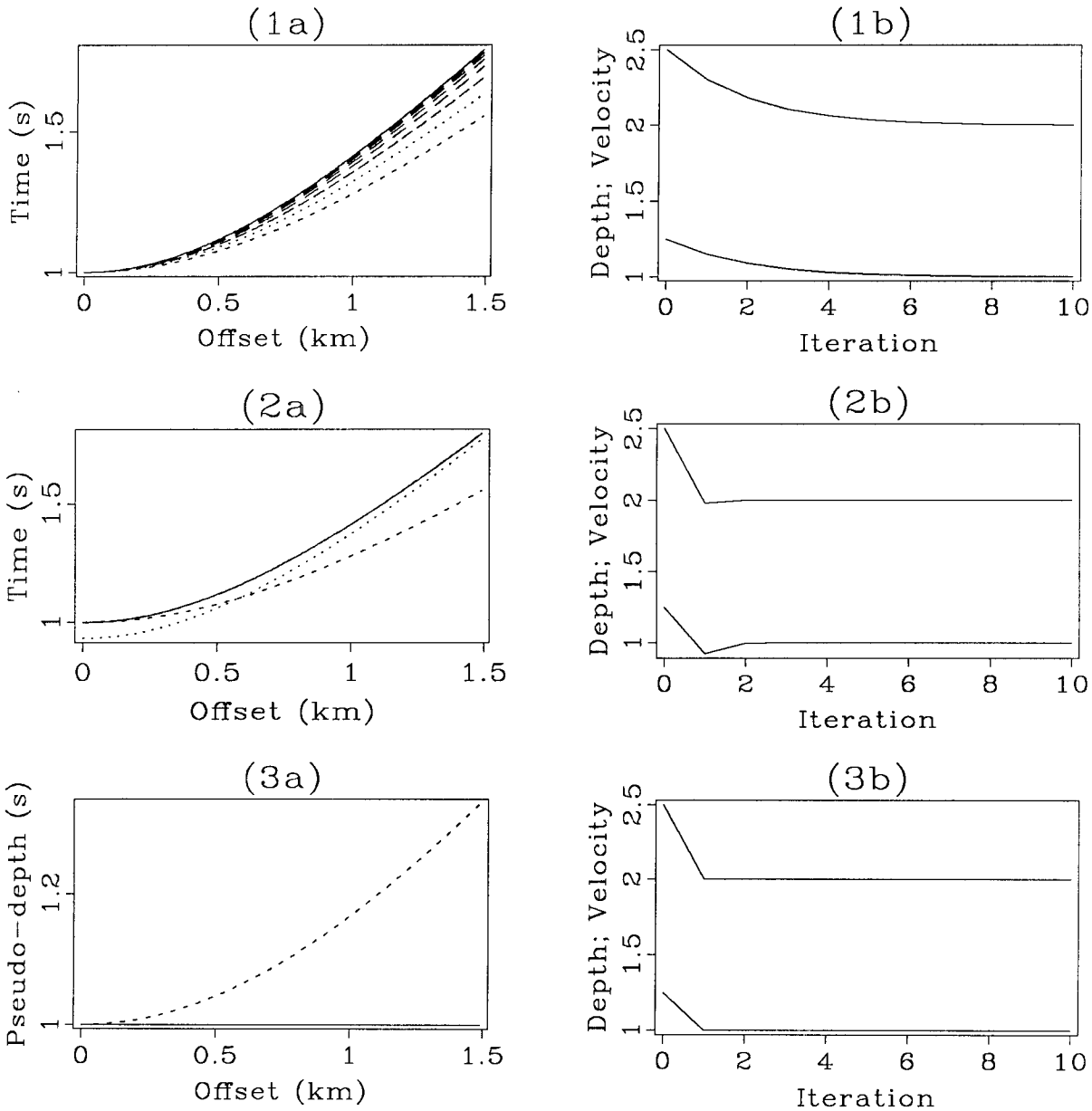


FIG. 1. Comparing the optimization presented in this paper with traveltime inversion for a constant-velocity model with a single reflector. (1): results of traveltime inversion with one parameter, velocity. The left plot shows traveltime-vs.-offset curves at each iteration in the inversion, with the solid line representing traveltime for the true model, and the dashed lines traveltime curves after each iteration. The right plot shows how velocity (in km/s) changes during iterations (upper curve). The lower curve in the same plot displays changes in the depth (in km) of the reflector, where depths are calculated from zero-offset traveltime and current velocity. (2): traveltime inversion with two parameters, velocity and depth. The plots are the same as in (1), but now the lower curve in Figure b represents inverted, instead of migrated depth. (3): pseudo-depth inversion with one parameter, velocity. The left plot shows the initial pseudo-depth curve after migration (dashed line), and the true depth (solid line). The right plot is the same as before.

they correctly predict perturbations in the migrated data, and fast convergence is achieved without having to add reflector positions as parameters in the inversion.

Of course, the above example is rather artificial in the sense that all the gradients and residuals are calculated analytically. The accuracy of numerically calculated gradients is limited to a linear region around the current model, and by including reflector movements in the calculations, this region may be smaller than in standard tomography. Also, one may have good reasons for using the unmigrated data, which are not contaminated by migration artifacts. On the other hand, since the migrated data correspond more closely to the geology, and are thus easier to interpret, analysis of migrated data may be preferable in structurally complex regions.

## GRADIENT CALCULATION

The objective function in the optimization is the semblance in the migrated image, calculated along a certain trajectory in the prestack migrated data. For now I assume that the prestack migrated data are a function of $x_m$ and $z_m$, the coordinates of the depth point, and $h$, the dimension in the offset direction—which corresponds to the difference between $x_m$ and shot location in the case of shot profile migration. Later I will describe how to calculate the gradient for depth-migrated data that have been converted to zero-offset time.

Initially, the stacking trajectory is flat along the offset direction as in normal semblance calculations, but it gets perturbed during linear iterations in the optimization. The objective function thus becomes a function of the stacking trajectory:

$$J(\mathbf{S}) \;=\; \frac{\left(\sum_{\mathbf{S}} d(\mathbf{S})\right)^2}{\sum_{\mathbf{S}} d^2(\mathbf{S})}, \tag{1}$$

where $\mathbf{S}$ is the trajectory in the prestack migrated data:

$$\mathbf{S} = (x_m(h), z_m(h), h), \qquad h = 0, h_{max}. \tag{2}$$

The gradient of the objective function can now be written as:

$$\frac{\partial J}{\partial \mathbf{m}} \;=\; \frac{\partial J}{\partial \mathbf{S}} \cdot \frac{\partial \mathbf{S}}{\partial \mathbf{m}} \;=\; \frac{\partial J}{\partial x_m}\,\frac{\partial x_m}{\partial \mathbf{m}} + \frac{\partial J}{\partial z_m}\,\frac{\partial z_m}{\partial \mathbf{m}} + \frac{\partial J}{\partial h}\,\frac{\partial h}{\partial \mathbf{m}}, \tag{3}$$

or, in matrix notation:

$$\frac{\partial J}{\partial \mathbf{m}} \;=\; \mathbf{D_1}\,\mathbf{M}, \tag{4}$$

with:

$$\mathbf{D_1} \;=\; \left(\frac{\partial J}{\partial x_m} \quad \frac{\partial J}{\partial z_m} \quad \frac{\partial J}{\partial h}\right), \quad \mathbf{M} \;=\; \begin{pmatrix} \dfrac{\partial x_m}{\partial \mathbf{m}} \\ \dfrac{\partial z_m}{\partial \mathbf{m}} \\ \dfrac{\partial h}{\partial \mathbf{m}} \end{pmatrix} \tag{5}$$

with $\mathbf{m}$ the model vector, which consists of the spline coefficients in the parametrization I discussed last year (Van Trier, 1988a). The first component of the gradient, $\mathbf{D_1}$, the first-order semblance derivatives, represents the interaction of the optimization with the data; the second component, $\mathbf{M}$, the derivatives with respect to $\mathbf{m}$, is calculated by ray tracing. I describe these two components separately in the next sections.

## Data component of the gradient

The data part of the gradient describes how the semblance changes if the stacking trajectory changes. The derivatives for this part are calculated by finite differences. For example, if we consider $\partial J/\partial z_m$:

$$\frac{\partial J}{\partial z_m}(\mathbf{S},h) \;=\; \frac{J(\mathbf{S}_{+\Delta z}(h)) \;-\; J(\mathbf{S}_{-\Delta z}(h))}{2\Delta z}, \tag{6}$$

where $\mathbf{S}_{+\Delta z}(h)$ is perturbed by $+\Delta z$ in the $z_m$ direction at offset $h$:

$$\mathbf{S}_{+\Delta z}(h) \;=\; \begin{cases} (x_m(h), z_m(h)+\Delta z, h) & \text{at offset } h, \\ \mathbf{S} & \text{elsewhere.} \end{cases} \tag{7}$$

Likewise, $\mathbf{S}_{-\Delta z}(h)$ is perturbed by $-\Delta z$ at offset $h$.

The gradient calculation is not very robust if the trajectory is just shifted at one offset, as described in the above equation. Therefore, the path is perturbed at several offsets, with the perturbations decreasing away from the offset under consideration. To visualize this, one can imagine the trajectory to be a rubber band. When we want to know how the semblance changes if the trajectory is perturbed at a certain offset, we pull the band at that offset, stretching it also at other offsets. We first pull it upwards, and then downwards, each time calculating the semblance along the perturbed trajectory. The difference of the two semblance values divided by the amount with which we pulled the band gives the derivative. Similarly, if we pull the band in the other directions, we can calculate $\partial J/\partial x_m$ and $\partial J/\partial h$. To stabilize the gradient calculations even more, a smoothed envelope of the data is used in the semblance determination, and the data are summed along a trajectory that has a width of more than one sample.

Note that the derivatives are a function of $h$, even though the semblance is calculated by stacking along the offset direction. This is because the semblance is a function of the stacking trajectory, and the trajectory can change as a function of offset.

## Ray-traced component of the gradient

After the boundaries have been picked from the migrated image, ray fans are traced from depth points on these boundaries to the surface. For a given ray fan (one reflector point), a set of points is found that have different offsets at the surface,

defining a curve in the prestack migrated data. Initially, this curve is a straight line (fixed $x_m$ and $z_m$, varying $h$). Semblance is calculated by stacking along this line. The previous section described how semblance changes with changes in the trajectory; this section discusses how the curve changes if the model is changed.

The raypaths and traveltimes along the rays are calculated by integrating a linear system of equations (see Van Trier, 1988b, and appendix). For the gradient calculation, two additional systems have to be integrated, one that calculates the derivatives of ray variables with respect to the model parameters, and one that describes the derivatives of ray variables with respect to the initial conditions of the ray (i.e., the position of the depth point, the dip of the reflector, and the take-off angle at the reflector). At the surface, these two sets of derivatives are combined to form the gradient operator **M**, which is needed in the gradient calculation of the objective function. I will now describe the procedure in detail.

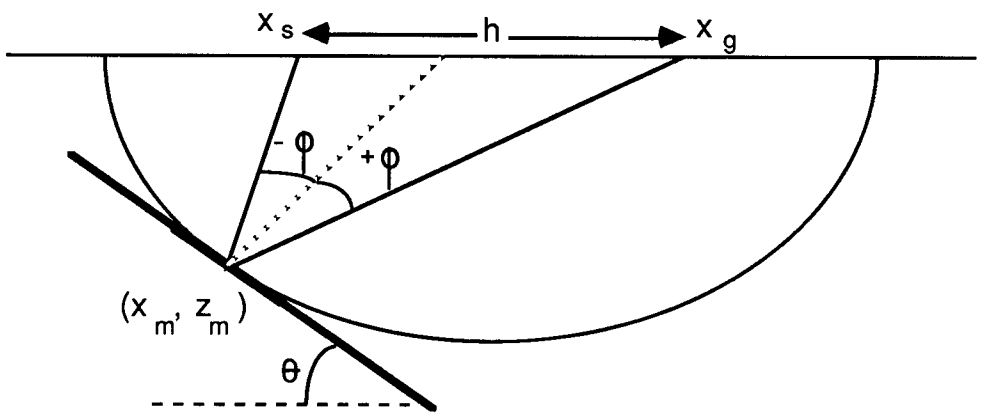


FIG. 2. Migration maps a data point with shot at $x_s$ and geophone at $x_g$ onto a semi-ellipse in depth. The shape of the ellipse is determined by the migration-velocity model. If the time dip of the data point is fixed, one depth point is found with a dip tangent to the ellipse.

Depth migration maps a point $(x_s, x_g, t\ )$ in the prestack data onto a migration ellipse in the subsurface (Figure 2), where $x_s$ denotes the shot position, $x_g$ the geophone position, and $t$ the traveltime. Assuming that the data point is part of a reflection event, the dip of the reflector can be determined from the stepout $p_h$ (time dip) of the event at the data point under consideration ($h$ denotes offset).

In the gradient calculation, the opposite mapping takes place: ray tracing tells us which data point $(x_s, x_g, t\ , p_h)$ in the original data corresponds to a point in the migrated data $(x_m, z_m, \phi, \theta)$. The position $(x_m, z_m)$ and dip $\theta$ of the reflector point are determined from the migrated image; the opening angle $\phi$ varies from some

minimum to some maximum angle. For each $\phi$, two rays are traced: a "shot" ray with opening angle -$\phi$, and a "geophone" ray with opening angle +$\phi$, (see Figure 2). The mapping can be written as:

$$\begin{pmatrix} x_s \\ x_g \\ t \\ p_h \end{pmatrix} = \begin{pmatrix} x_s(x_m, z_m, \phi, \theta; \mathbf{m}) \\ x_g(x_m, z_m, \phi, \theta; \mathbf{m}) \\ t\ (x_m, z_m, \phi, \theta; \mathbf{m}) \\ p_h(x_m, z_m, \phi, \theta; \mathbf{m}) \end{pmatrix}. \tag{8}$$

The mapping is fully general, that is, no assumptions are made about the velocity model or the migration. The only assumption is that a continuous reflector segment exists; diffraction events are not included in the gradient calculation.

If we perturb the migration-velocity model, we will image the same data point at a different position, and the reflector dip and opening angle will be different:

$$\begin{pmatrix} x_s(x_m, z_m, \phi, \theta; \mathbf{m}) \\ x_g(x_m, z_m, \phi, \theta; \mathbf{m}) \\ t\ (x_m, z_m, \phi, \theta; \mathbf{m}) \\ p_h(x_m, z_m, \phi, \theta; \mathbf{m}) \end{pmatrix} = \begin{pmatrix} x_s(x_m + \delta x_m, z_m + \delta z_m, \phi + \delta\phi, \theta + \delta\theta; \mathbf{m} + \delta\mathbf{m}) \\ x_g(x_m + \delta x_m, z_m + \delta z_m, \phi + \delta\phi, \theta + \delta\theta; \mathbf{m} + \delta\mathbf{m}) \\ t\ (x_m + \delta x_m, z_m + \delta z_m, \phi + \delta\phi, \theta + \delta\theta; \mathbf{m} + \delta\mathbf{m}) \\ p_h(x_m + \delta x_m, z_m + \delta z_m, \phi + \delta\phi, \theta + \delta\theta; \mathbf{m} + \delta\mathbf{m}) \end{pmatrix}. \tag{9}$$

Then, if the mapping function defined in equation (8) is locally linear, we can write:

$$\mathbf{G} \begin{pmatrix} \delta x_m \\ \delta z_m \\ \delta\phi \\ \delta\theta \end{pmatrix} = \mathbf{H}\ \delta\mathbf{m}, \tag{10}$$

with:

$$\mathbf{G} = -\begin{pmatrix} \frac{\partial x_s}{\partial x_m} & \frac{\partial x_s}{\partial z_m} & \frac{\partial x_s}{\partial \phi} & \frac{\partial x_s}{\partial \theta} \\ \frac{\partial x_g}{\partial x_m} & \frac{\partial x_g}{\partial z_m} & \frac{\partial x_g}{\partial \phi} & \frac{\partial x_g}{\partial \theta} \\ \frac{\partial t}{\partial x_m} & \frac{\partial t}{\partial z_m} & \frac{\partial t}{\partial \phi} & \frac{\partial t}{\partial \theta} \\ \frac{\partial p_h}{\partial x_m} & \frac{\partial p_h}{\partial z_m} & \frac{\partial p_h}{\partial \phi} & \frac{\partial p_h}{\partial \theta} \end{pmatrix}; \quad \mathbf{H} = \begin{pmatrix} \frac{\partial x_s}{\partial \mathbf{m}} \\ \frac{\partial x_g}{\partial \mathbf{m}} \\ \frac{\partial t}{\partial \mathbf{m}} \\ \frac{\partial p_h}{\partial \mathbf{m}} \end{pmatrix}. \tag{11}$$

$\mathbf{H}$ is a 4x$N$ matrix, with $N$ the number of model parameters. All the derivatives are taken for the unperturbed state. This means that the matrix $\mathbf{H}$ represents the derivatives with respect to the model for fixed depth point, dip, opening angle, and current velocity model. The same applies to the matrix $\mathbf{G}$. Both $\mathbf{G}$ and $\mathbf{H}$ are calculated by integrating a system of linear equations (see appendix).

By multiplying equation (10) with the inverse of $\mathbf{G}$, we find:

$$\begin{pmatrix} \delta x_m \\ \delta z_m \\ \delta\phi \\ \delta\theta \end{pmatrix} = \mathbf{G}^{-1}\ \mathbf{H}\ \delta\mathbf{m} = \mathbf{M}\ \delta\mathbf{m}, \tag{12}$$

with **M** the matrix of the derivatives of the depth-point variables with respect to the model parameters:

$$\mathbf{M} = \begin{pmatrix} \frac{\partial x_m}{\partial \mathrm{m}} \\ \frac{\partial z_m}{\partial \mathrm{m}} \\ \frac{\partial \phi}{\partial \mathrm{m}} \\ \frac{\partial \theta}{\partial \mathrm{m}} \end{pmatrix} = \mathbf{G}^{-1}\,\mathbf{H}. \tag{13}$$

These gradients describe how a point in the original data moves in the migrated data space if the velocity model is changed. Although the definition of **M** is slightly different than the one used in equation (5), they represent the same variables; the gradient of offset with respect to the model parameters is simply found by combining the derivatives of $\phi$ and $\theta$, and scaling them by factors that relate offset to angle:

$$\frac{\partial h}{\partial \mathrm{m}} = \frac{\partial h}{\partial \phi}\,\frac{\partial \phi}{\partial \mathrm{m}} + \frac{\partial h}{\partial \theta}\,\frac{\partial \theta}{\partial \mathrm{m}}. \tag{14}$$

The scale factors are determined from the ray-traced results.

## VISUALIZING THE GRADIENT OPERATOR

I will now illustrate the gradient calculations with a simple example. The gradients are calculated for a constant-velocity medium, a medium for which results can still be intuitively understood. Later I will show results for more complicated media. The velocity in the medium is 2.5 km/s, and the corresponding slowness function is represented by 2-D cubic splines, parametrized by 256 coefficients (16 in each direction).

The gradients are a function of 4 variables: depth and lateral position of the reflector point, reflector dip, and opening angle. Figure 3 shows the gradients of the unmigrated data variables with respect to the model parameters (matrix **H**) for fixed reflector position and dip, and two different opening angles. The pictures display intensity plots of the various gradient operators as a function of model parameter. The operators are smooth because they have been re-interpolated onto the grid of the model for display purposes. The pattern for $x_s$ can be explained as follows: if a spline coefficient slightly above the ray is increased, the ray will bend toward the center, increasing $x_s$; if a coefficient below the ray is increased, $x_s$ decreases. The same behavior can be seen in the gradient of $x_g$. For $t$, two effects have to be considered: a slowness effect and a raypath effect. If the slowness increases along the ray, the traveltime increases; however, since the ray bends towards the center of the fan, the raypath gets shorter—decreasing the traveltime. For wide opening angles, the gradient is stronger along the geophone branch of the raypath than along the shot branch (barely visible in the plot), because most of the changes in traveltime are caused by ray-bending effects along the nearly horizontal geophone

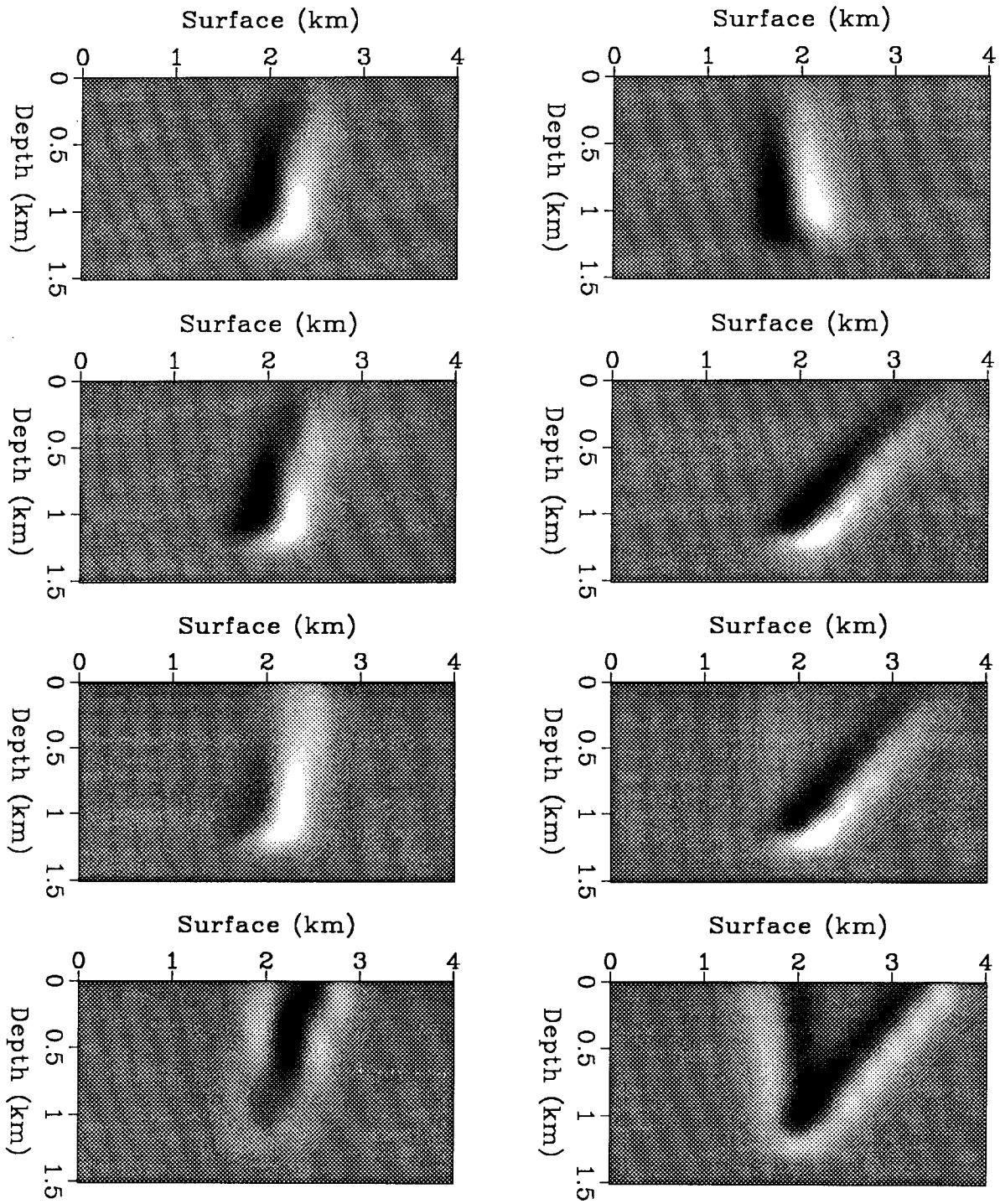


FIG. 3. Gradients of the surface variables with respect to the model parameters for fixed depth point and dip (20°). The figures show intensity plots of the gradients as a function of model parameter (negative values are dark, positive values are light). The operators are interpolated onto the model grid (the size of which is 400x150; the size of the spline grid is only 16x16). From top to bottom, the plots show the gradients of $x_s$, $x_g$, $t$, and $p_h$, respectively. The left side shows gradients for an opening angle of 0°, the gradients on the right side have an opening angle of 30°.

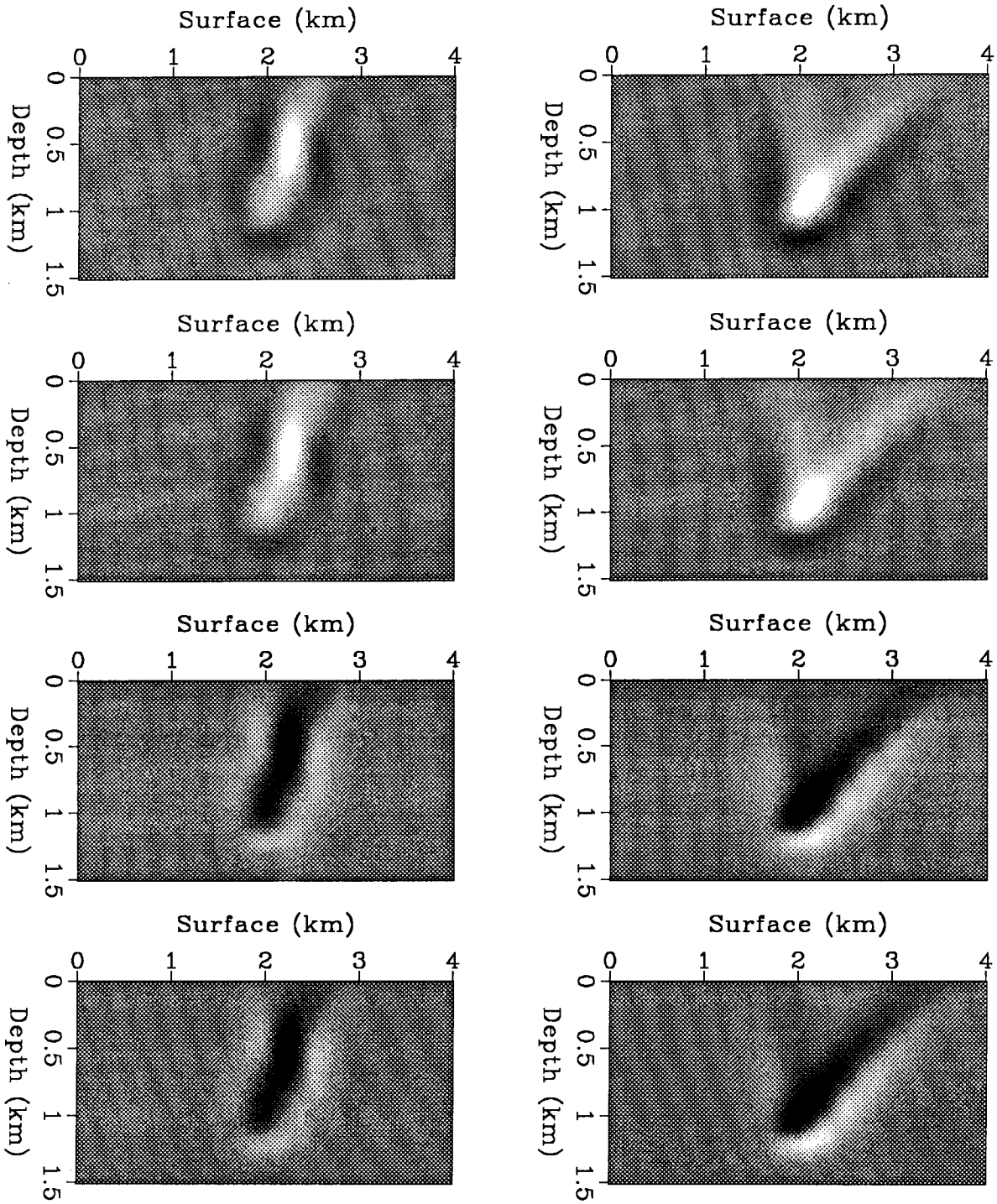


FIG. 4. Same as Figure 3, but now displaying the gradients of the subsurface variables. From top to bottom, the figure displays the gradients of $x_m$, $z_m$, $\phi$, and $\theta$, respectively.

ray. The gradient of $p_h$ displays similar raypath effects: if the slowness is increased at the inner side of the raypath, both branches of the ray will bend toward the center, and the time dip of the event will decrease.

The gradient operators for the migrated data are shown in Figure 4. They are determined by multiplying the gradients discussed above with the inverse of **G**. The calculation of **G** is described in the appendix; its inverse is found using singular-value decomposition. I will qualitatively explain the shape of the gradients with a schematic drawing (Figure 5); the next section discusses a quantitative verification of the gradient operator. Consider the sign of the gradients along the shot ray in the right plots of Figure 4. If slowness is increased in a spline cell above the ray, the depth and lateral position of the reflector point increase, whereas its opening angle and dip decrease. When a spline coefficient below the ray is increased, the opposite changes occur. Figure 5 shows raypaths of a single data point for three different migration models: the unperturbed model, and the two perturbed models described above. Since the slowness decreases, the length of the perturbed rays has to increase if the traveltime has to stay the same. rays. Likewise, in order for the rays to arrive at the same surface location, the reflector point has to move as shown in the plots. The plots show that the movement of the reflector point is indeed as predicted by the gradients.

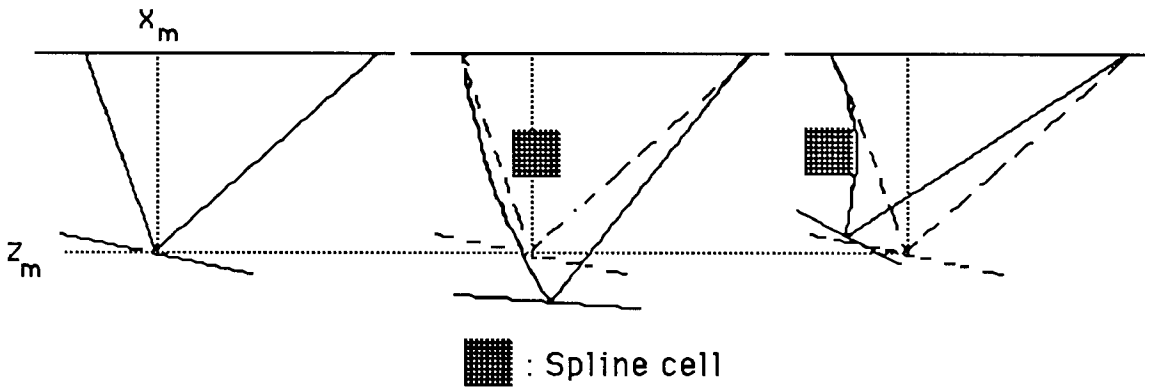


FIG. 5. Schematic explanation of the gradient operators in Figure 4. The left plot shows the raypaths for the unperturbed model. The middle figure shows how the reflector point moves if slowness is increased in a spline cell above the left ray. The dashed lines show the unperturbed rays for reference. The changes correspond to the signs in the gradients of Figure 5. If a spline coefficient below the ray is perturbed (right plot), the opposite effect occurs.

## PREDICTING DEPTH POINT ANOMALIES

To verify the operators, they are used to predict perturbations in $x_m$, $z_m$, $\phi$, and $\theta$ for a perturbed model. First, rays are traced in the unperturbed model from a certain depth point with a given dip. Then, the gradient operator **M** is used to predict perturbations in depth point, opening angle, and reflector dip for

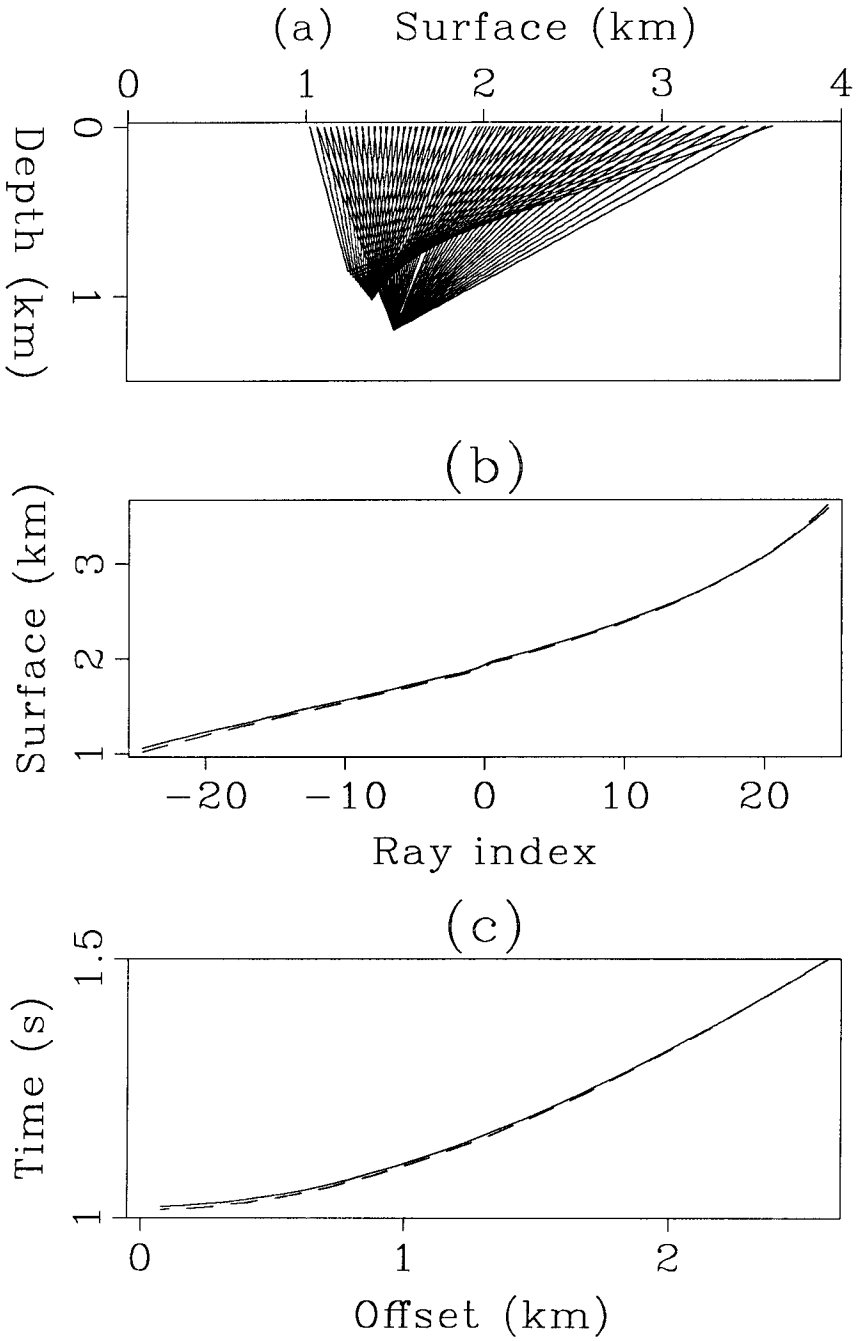


FIG. 6. Movement of reflector points as predicted by the gradient operators. Figure (a) shows a ray fan for a constant-velocity reference model (starting from one depth point) and one for a perturbed model, which has a 10 % decrease in velocity. The zero-offset rays are not plotted to show how the reflector dip (which is perpendicular to the white area of the left-out zero-offset ray) changes. The perturbed depth points, opening angles, and reflector dip are predicted by multiplying the gradients with the model perturbations. Figure (b) shows the surface locations as a function of ray index (negative for "shot" rays, positive for "geophone" rays) for unperturbed (solid line) and perturbed model (dashed line). Figure (c) does the same for traveltimes as a function of offset.

a certain model perturbation $\delta\mathbf{m}$, by calculating $\mathbf{M}\delta\mathbf{m}$. Finally, rays are traced for the perturbed model, and their surface locations and traveltimes are compared with those for the unperturbed model. If the gradients are correct, the surface positions and traveltimes should be the same, since the gradients predict how the depth points move for fixed data points as a function of migration velocity.

The unperturbed model is the same model as before, the perturbed one has a velocity that is 10 % too low. Figure 6a shows two ray fans, one for the unperturbed model (starting from one depth point), and one for the perturbed model. In the perturbed model, the depth points have been perturbed as predicted by the gradients, and the figure shows how the depth points are smeared out if the wrong migration velocity is used. Similarly, the reflector dip and opening angle are perturbed; the reflector dip perturbation is the same for all rays, as can be expected for a constant velocity medium. As can be seen in Figures 6b and c, the surface positions and traveltimes of the perturbed rays closely match those of the unperturbed ones.

## OPTIMIZATION SCHEME

The gradients described above are used in an optimization scheme that perturbs the velocity model such that semblance is maximized. The optimization consists of an iterative conjugate-gradient search method. At each iteration a line search along the gradient direction is carried out, which finds $\alpha$ that maximizes $J(\mathbf{m} + \alpha\, \partial J/\partial \mathbf{m})$. Semblance evaluations in the line search are performed by stacking along a perturbed trajectory, where perturbations in the trajectory are predicted as described in the previous section.

Several choices can be made with respect to the (non-)linearity of the gradient calculations in the optimization. In a fully non-linear scheme the data are remigrated after each model update, boundaries are repicked, and gradients are recalculated by ray tracing. Obviously, this is expensive, even when the migration and ray tracing can be restricted to few events in the prestack data. There are cheaper, linear alternatives for each of the three steps (migration, picking, and gradient calculation). For moderate model changes, residual migration of the prestack data is well-approximated by residual moveout of the migrated data, with the moveout defined by the perturbations in the stacking trajectory (residual migration "straightens" the stacking curve). As for the boundaries, they need not be repicked at each iteration if the initial migrated image is converted to a zero-offset section. An inexpensive event migration of this section determines the new boundary positions in the updated model. When the gradients are calculated in the zero-offset-time domain (see later section), the boundaries are picked only once from the zero-offset section, after which they stay fixed. Finally, it may not be necessary to recalculate the ray-traced component of the gradient if the non-linearity in the gradient operator mainly comes from the data component. At each iteration, only the semblance derivatives are recalculated, while the ray-traced gradient component is kept fixed.

I have not yet tested the different linearizations described above, so I cannot tell which ones work best, and how to incorporate them in the optimization scheme. I imagine the scheme to consist of several loops, with linear inner loops and non-linear outer loops. The optimization iterates in the inner loops until the model updates converge, after which gradients are recalculated non-linearly in the outer loops.

Conjugate gradients converges to a model close to the solution; however, near the solution the convergence rate can be improved if second derivatives of the objective function with respect to the model parameters (Hessian) are used. Gauss-Newton methods are examples of methods that use the Hessian, and Biondi (1988) shows that the Gauss-Newton solution of an optimization problem similar to the one described in this paper is equivalent to the solution of the following system:

$$\sqrt{\mathbf{D_2}}\,\mathbf{M}\,\delta\mathbf{m} \;=\; \left(\sqrt{\mathbf{D_2}}\right)^{-1}\mathbf{D_1} \tag{15}$$

with $\mathbf{D_2}$ the matrix of second-order semblance derivatives in the data. Following Biondi, I use constants for these derivatives, since they cannot be reliably estimated from the data. The system can be solved with standard techniques such as LSQR (Paige and Saunders, 1982).

## RESULTS: LINEAR OPTIMIZATION

I have tested the optimization for a simple example, which I already presented in SEP–59 (Van Trier, 1988c). Finite-difference data are calculated for a 3-layered medium with a flat and dipping reflector and a low-velocity anomaly at the top (see Figure 7). The data are migrated using the same velocity model, but without the anomaly. In SEP–59 I showed several slices through the migrated data, and I refer the reader to that report for more details; here I just display the zero-offset section (Figure 8).

The Gauss-Newton method described above is used to invert the migrated data. Boundaries are picked from the migrated image, and ray fans are traced from the boundaries to the surface for the calculation of the ray-traced components of the gradients. The maximum opening angle in the fans is determined by the maximum offset in the data. For the data part of the gradient operator, I have only used semblance derivatives with respect to depth. The derivatives with respect to offset had little influence on the optimization (offset perturbations are small because the anomaly is at the surface; rays can not accumulate large raypath distortions). The same applies to the derivatives with respect to surface location. In practice, it may not be desirable to use the latter, anyway, since they are distorted by lateral changes in the reflection coefficient along the reflector.

Figure 9 shows the result of an optimization with a single fan, calculated for one depth point on the flat reflector. The number of parameters in the optimization is 256, representing 16x16 spline coefficients of a 2-D spline function. The inversion has located the anomaly laterally, but its amplitude (Figure 9b) and depth are not

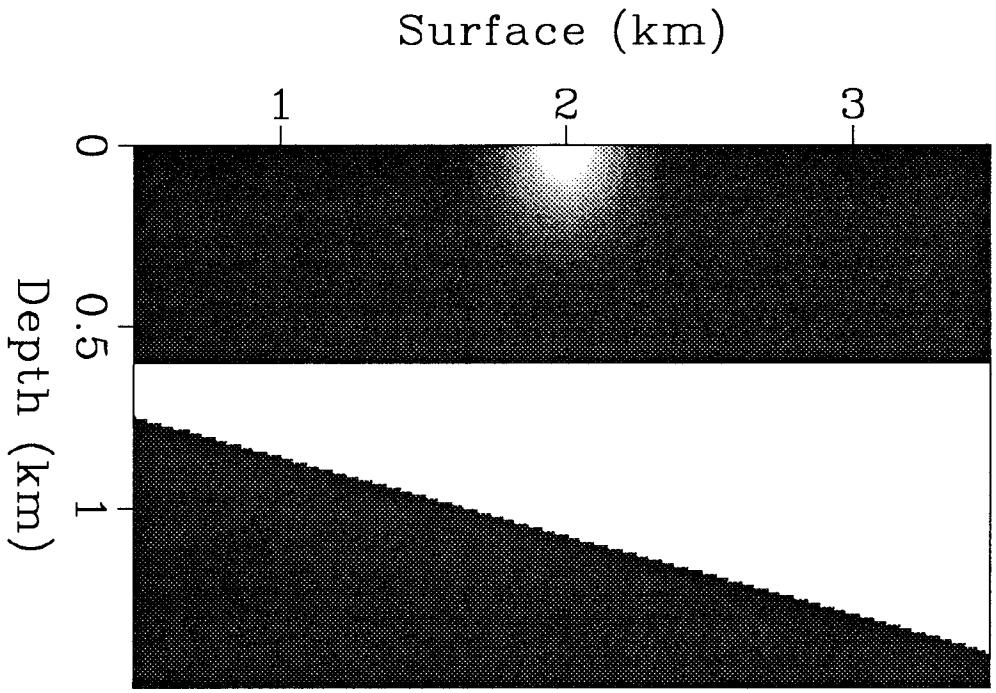


FIG. 7. Velocity model with anomaly at the surface. The velocity in the center of the anomaly is 2 km/s, the velocity in the top layer is 2.5 km/s, the one in the middle is 3 km/s.

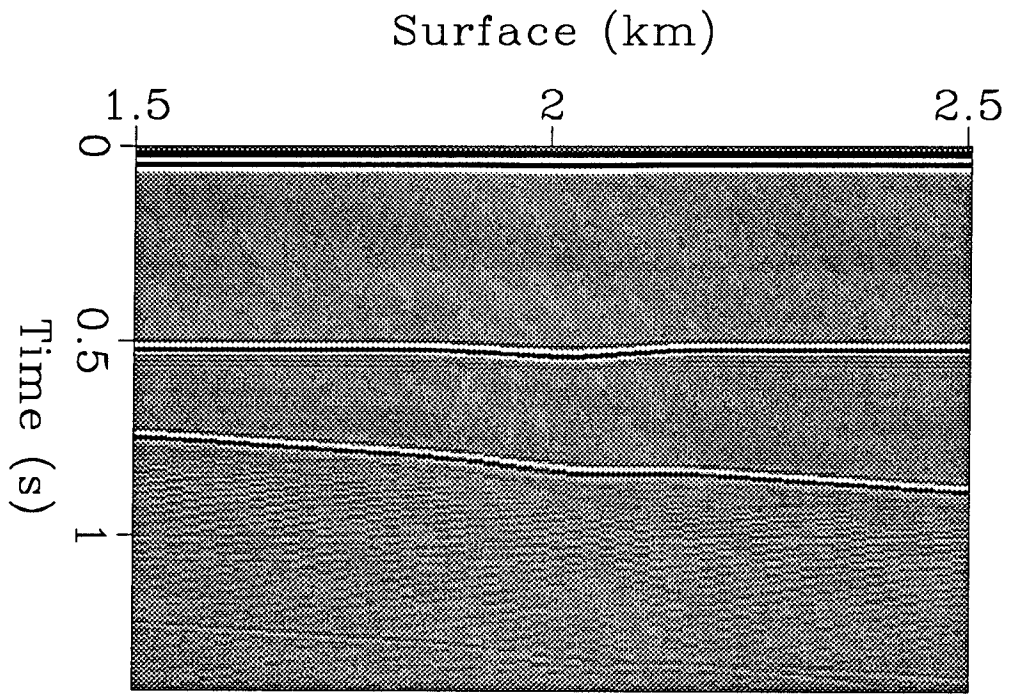


FIG. 8. Zero-offset section of data calculated by finite differences for the model of Figure 7.

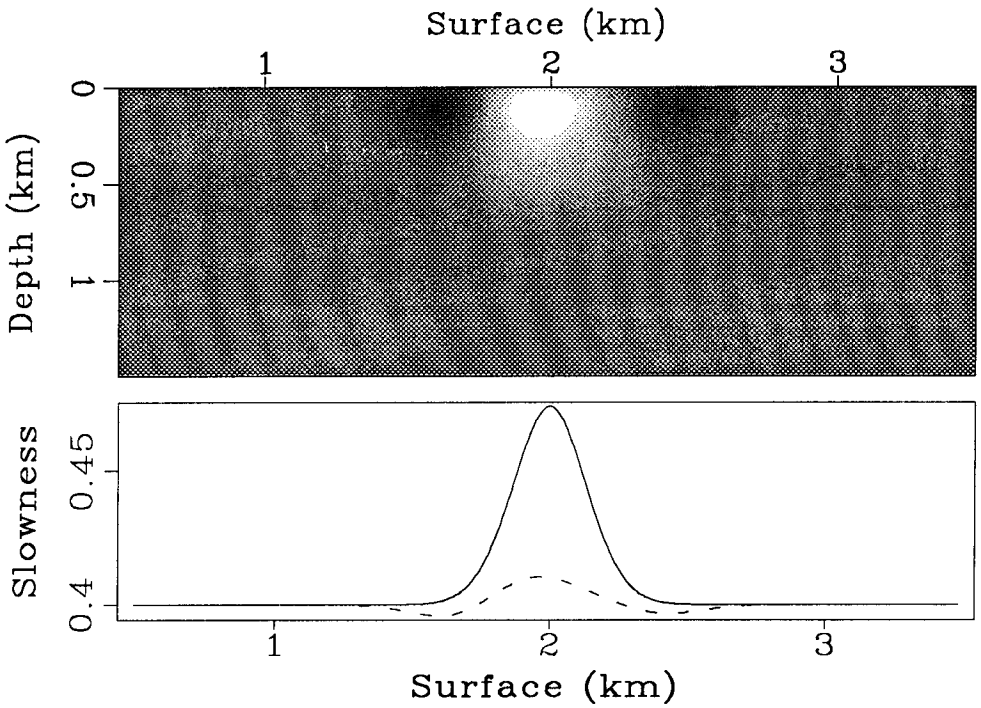


FIG. 9. Inversion result using a single fan, starting from a depth point on the flat reflector directly below the anomaly. Figure (b) shows a depth slice at a depth of 100 m, where the true anomaly is plotted as a solid line for reference. The slowness is plotted in s/km.

well resolved. The inverted anomaly has the side-lobes characteristic of tomography (Fowler, 1988; Stork, 1988), which are caused by the limited ray coverage in a single reflection experiment.

Figures 10 and 11 display results of inverting multiple fans for the flat and dipping reflector, respectively. The spacing of fans is 20 m, and they extend laterally from 1.5 to 2.5 km (the same geometry as the shots). With the increased ray coverage, the depth extent of the anomaly is better resolved, and its amplitude estimate is improved. As expected, the result for the flat reflector is better than for the dipping one, since the closer flat reflector illuminates the anomaly at higher opening angles. Also, the semblance derivatives of the data from the dipping reflector are contaminated by diffraction events caused by the step-like character of the dipping boundary in the discrete model. The plots still show some artifacts, most noticeably the two "arms" extending from the anomaly downward. They can be explained as end effects due to limited shot coverage. Depth points in the center of the survey are illuminated by many shots with the anomaly "visible" at different offsets for each shot. At the sides of the survey, the anomaly is only partly illuminated, and therefore not well-resolved in the inversion.

Inverting data for both the flat and dipping reflector simultaneously yields virtually the same result as the flat-reflector inversion, and I have not reproduced it

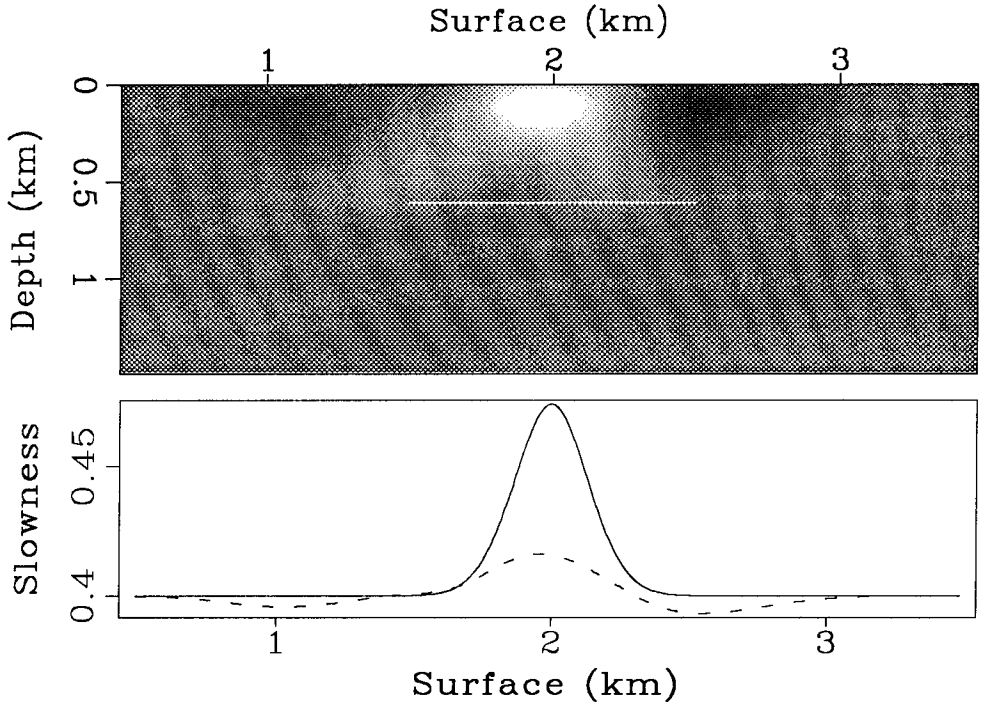


FIG. 10. Same as Figure 9, but for multiple fans and using migrated data from the flat bed only. The starting points of the fans extend laterally from 1.5 to 2.5 km, as is shown by the white line in the figure.

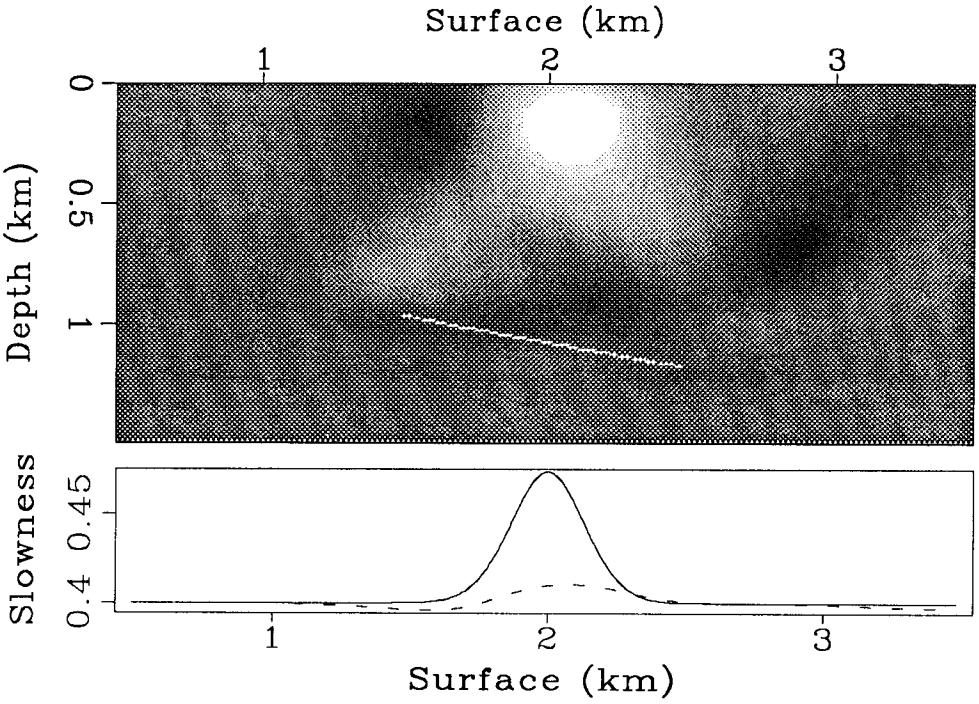


FIG. 11. Same as Figure 10, but for dipping-bed data only.

here. The flat reflector illuminates the anomaly at a wide range of angles; adding a reflector with a more limited ray coverage has little effect. A better way to improve the result would be to extent the shot coverage of the survey, or to do a non-linear iteration where rays are traced through the updated model.

## THE GRADIENT IN THE ZERO-OFFSET-TIME DOMAIN

So far, I have been assuming that the data have been depth migrated. The problem with depth migration is that not only the offset behavior of the events changes when the velocity changes, but the events also move up or down, depending on the velocity. This movement may have several unwanted effects on the optimization. First, if boundaries are picked from the depth-migrated image, they have to be readjusted at every iteration. Second, the vertical movement of the event has virtually no effect on the objective function (the semblance stays the same if both event and stacking trajectory move vertically as a function of migration velocity), and is therefore expected to be ill-resolved. Finally, since the linear operators can only predict a limited amount of reflector movement, more non-linear iterations are required if vertical reflector movements are included in the optimization.

The solution to these problems is to convert the migrated data from the depth domain to the zero-offset-time domain using the ray-traced results. In this domain, the zero-offset section always stays the same, and the optimization only has to invert for the non-zero-offset behavior of the events (see also Etgen, 1988).

Once the events in the zero-offset section have been mapped into boundaries in the depth section using event migration, the data transformation is nothing more than a simple scaling and re-ordering of the depth migrated data. Consider a constant-$x_m$ gather in the prestack migrated data. For each event in the gather, the aforementioned mapping tells us how to scale the depth axis to the zero-offset-time ($\tau_m$) axis, and to which midpoint $y_m$ the event belongs:

$$d_{zof}(\tau_m, y_m, h') \;=\; d_{zof}(t_0(x_m, z_m; \mathrm{m}),\; y_0(x_m, z_m; \mathrm{m}),\; h') \;=\; d(x_m,\; z_m,\; h), \quad (16)$$

with $t_0$ and $y_0$ the traveltime and midpoint of the zero-offset ray, respectively. I have made the distinction between $\tau_m$, $y_m$ and $t_0$, $y_0$ to stress that the first variables denote coordinate axes in the transformed data, whereas the second ones are properties of the zero-offset ray. As before, offset is not the same as the original offset in the reflection experiment, rather it corresponds to the opening angle at the reflector. There is a slight difference with the previously defined $h$ (the distance between $x_m$ and shot position), though: $h'$ is the distance between $y_m$ and the shot position. This means that the zero-offset section in the transformed migrated data is the same as the zero-offset section in the unmigrated data.

The gradient of $h'$ is determined from the gradients of $\phi$ and $\theta$ as in equation (14), the other ones are calculated by converting the depth-domain gradients. As I will show now, the conversion is not complicated; all the needed conversion factors are

already calculated. The expressions for the gradient of zero-offset time and midpoint are:

$$\frac{\partial \tau_m}{\partial \mathbf{m}} = \frac{\partial t_0}{\partial \mathbf{m}} + \frac{\partial t_0}{\partial x_m}\frac{\partial x_m}{\partial \mathbf{m}} + \frac{\partial t_0}{\partial z_m}\frac{\partial z_m}{\partial \mathbf{m}}; \tag{17}$$

$$\frac{\partial y_m}{\partial \mathbf{m}} = \frac{\partial y_0}{\partial \mathbf{m}} + \frac{\partial y_0}{\partial x_m}\frac{\partial x_m}{\partial \mathbf{m}} + \frac{\partial y_0}{\partial z_m}\frac{\partial z_m}{\partial \mathbf{m}}. \tag{18}$$

All the derivatives on the right-hand sides of the equations are already calculated: $\partial x_m/\partial \mathbf{m}$ and $\partial z_m/\partial \mathbf{m}$ are the same as before; the other derivatives are various zero-offset elements in the matrices $\mathbf{G}$ and $\mathbf{H}$.

The first part of the derivatives describes how, for a given depth section, the zero-offset section changes as a function of velocity; the second part relates reflector movement (due to changes in migration velocity) to changes in the zero-offset section. At zero offset, the two parts cancel each other, and the gradients are zero. At non-zero offset, the two parts cancel only when the correct migration velocity is used. The pseudo-depth inversion in Figure 1 uses the gradient defined in equation (17).

## CONCLUSIONS

I have presented a gradient calculation that can be used in a event-driven velocity-optimization method. Since the gradients include reflector movement as a function of velocity, structural boundaries do not have to be included as parameters in the estimation problem. The calculations do not make assumptions about the velocity model, and should be readily applicable in structurally complex areas.

The results that I have shown here deal with small velocity perturbations, where the linear gradient operators are expected to work well; I still have to test the method for structural models with large velocity anomalies. Furthermore, the semblance derivative operators have to be verified for field data.

## ACKNOWLEDGMENTS

I benefited a lot from discussions with Francis Collino during his short stay at Stanford. His initial help in developing the theory for the gradient calculations is greatly appreciated. I also thank Biondo Biondi, John Etgen and Marta Woodward for many interesting discussions on velocity analysis and migration. Finally, I thank Shell Internationale Petroleum Maatschappij for financial contributions.

## REFERENCES

Al-Yahya, K., 1987, Velocity analysis by profile migration: Ph.D. thesis, Stanford University.

Biondi, B., 1988, Interval velocity estimation from beam stacked data—an improved method: SEP–**57**, 99-116.

Červený, V., 1987, Ray tracing algorithms in three-dimensional laterally varying layered structures, *in* Nolet, G., Ed., Seismic Tomography: Riedel Publishing Company, 99–134.

Etgen, J., 1988, Prestack depth migration velocity analysis: linear theory revised: SEP–**59**, 121–139.

Fowler, P., 1988, Prestack migration velocity estimation: Ph.D. thesis, Stanford University.

Landa, E., Kosloff, D., Keydar, S., Koren Z., and Reshef, M., 1988, A method for determination of velocity and depth from seismic reflection data: Geophys. Prosp., **36**, 223–243.

Nolet, G., 1987, Seismic wave propagation and seismic tomography, *in* Nolet, G., Ed., Seismic Tomography: Riedel Publishing Company, 1–24.

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

Stork, C., 1988, Travel time tomography velocity analysis of seismic reflection data: Ph.D. thesis, Caltech.

Stork, C., and Clayton, R., 1987, Application of tomography to two data sets containing lateral velocity variation: Presented at the 57th Ann. Internat. Mtg., Soc. Explor. Geophys.

Van Trier, J., 1988a, Geological constraints in velocity inversion: SEP–**57**, 117–138.

Van Trier, J., 1988b, Vectorized initial-value ray tracing for arbitrary velocity models: SEP–**57**, 279–288.

Van Trier, J., 1988c, Velocity analysis of migrated seismic data after structural interpretation: SEP–**59**, 141–150.

Verprat, M., Perroud, H., Landa E., and Haas, A., 1988, Structural inversion by iterative prestack migration: Presented at the 58th Ann. Internat. Mtg., Soc. Explor. Geophys.

## APPENDIX

In this Appendix I first review the ray-tracing method I described in SEP-57, and then I discuss how the derivatives needed for the ray-traced component of the gradient are calculated.

### Ray tracing

The ray tracing method that I have implemented is based on the ray tracing system of first-order partial-differential equations, derived from the Eikonal equation by the method of characteristics (see Nolet, 1987 and Červený, 1987 for more

details):

$$\begin{cases} \dfrac{dx}{ds} &= \dfrac{p_x}{w(\mathbf{m})}, \\ \dfrac{dz}{ds} &= \dfrac{p_z}{w(\mathbf{m})}, \\ \dfrac{dp_x}{ds} &= \dfrac{\partial w(\mathbf{m})}{\partial x}, \\ \dfrac{dp_z}{ds} &= \dfrac{\partial w(\mathbf{m})}{\partial z}. \end{cases} \tag{19}$$

Here $w(\mathbf{m})$ is the slowness as a function of the model parameter vector $\mathbf{m}$, $(x,z) = (x(s),z(s))$ are the coordinates of the ray as a function of $s$, the arclength along the ray, and $(p_x(s),p_z(s))$ are the components of the slowness vector $\mathbf{p}$. The traveltime $t(s)$ along the ray is given by:

$$\frac{dt}{ds} = w(\mathbf{m}). \tag{20}$$

The slowness vector is perpendicular to the wavefront ($\mathbf{p} = \nabla t$), and must satisfy:

$$|\mathbf{p}|^2 = p_x^2 + p_z^2 = w^2. \tag{21}$$

The system of ray equations together with equation (20) can be solved by a standard numerical integration method. I use a fourth-order Runge-Kutta method.

To simplify notations, equations (19) and (20) can be combined into one equation if we organize the ray variables in a vector $\mathbf{x} = (x\ z\ p_x\ p_z\ t)^T$:

$$\begin{cases} \dfrac{d\mathbf{x}}{ds} &= \mathbf{a}(\mathbf{x},\mathbf{m}) \\ \mathbf{x}(s=0) &= \mathbf{x}_0(x_m,z_m,\phi,\theta,\mathbf{m}) \end{cases} \tag{22}$$

with:

$$\mathbf{a} = \left(\frac{p_x}{w(\mathbf{m})}\ \ \frac{p_z}{w(\mathbf{m})}\ \ \frac{\partial w(\mathbf{m})}{\partial x}\ \ \frac{\partial w(\mathbf{m})}{\partial z}\ \ w(\mathbf{m})\right)^T; \tag{23}$$

$$\mathbf{x}_0 = (x_m\ \ z_m\ \ p_{x,0}(\phi,\theta,\mathbf{m})\ \ p_{z,0}(\phi,\theta,\mathbf{m})\ \ 0)^T. \tag{24}$$

$(x_m,z_m)$ are the coordinates of the starting point of the ray, and $(p_{x,0},p_{z,0})$ are the components of its initial slowness vector, which depend on the slowness at the starting point and the take-off angle $\phi$. In my application, $(x_m,z_m)$ is the position of a migrated depth point, and the take-off angle is the angle with respect to the normal at the reflector. For example, if the dip at the reflector is given by $\theta$, the components of the initial slowness vector are as follows:

$$p_{x,0} = w(x_m,z_m;\mathbf{m})\ \sin(\theta+\phi); \tag{25}$$

$$p_{z,0} = w(x_m,z_m;\mathbf{m})\ \cos(\theta+\phi). \tag{26}$$

In a reflection experiment, a raypath consists of two branches: a ray traveling from the shot to the reflector, and a ray traveling from the reflector to the receiver. At the reflector the angle of incidence is equal to the angle of reflection, so we can calculate the two branches by solving the above system with two different initial condition vectors, one with $+\phi$ as opening angle, and one with $-\phi$. In the following discussion, I will denote the two branches by superscripts $S$ and $G$, for source and geophone, respectively.

## Derivatives with respect to the model parameters: matrix H

As in all tomographic problems, the gradient calculations need to determine how the ray variables change if the model is changed. However, traditional tomographic problems consider only one ray variable, the traveltime, and assume fixed raypaths. Since I do not use picked traveltimes, and allow for the movement of the reflector, I have to consider all ray variables and include raypath variations. Therefore, the complete system of ray equations (equation 22) has to be differentiated with respect to $\mathbf{m}$:

$$\begin{cases} \dfrac{d\left(\partial \mathbf{x}/\partial \mathbf{m}\right)}{ds} &= \dfrac{\partial \mathbf{a}}{\partial \mathbf{x}}\dfrac{\partial \mathbf{x}}{\partial \mathbf{m}} + \dfrac{\partial \mathbf{a}}{\partial \mathbf{m}} \\ \dfrac{\partial \mathbf{x}}{\partial \mathbf{m}}(s=0) &= \dfrac{\partial \mathbf{x}_0}{\partial \mathbf{m}} \end{cases} \quad \text{or:} \quad \begin{cases} \dfrac{d\mathbf{y}}{ds} &= \mathbf{A}\mathbf{y} + \mathbf{b} \\ \mathbf{y}(s=0) &= \mathbf{y}_0 \end{cases} \tag{27}$$

with:

$$\mathbf{A} = \frac{\partial \mathbf{a}}{\partial \mathbf{x}} = \begin{pmatrix} -\dfrac{p_x}{w^2}\dfrac{\partial w}{\partial x} & -\dfrac{p_x}{w^2}\dfrac{\partial w}{\partial z} & \dfrac{1}{w} & 0 & 0 \\ -\dfrac{p_z}{w^2}\dfrac{\partial w}{\partial x} & -\dfrac{p_z}{w^2}\dfrac{\partial w}{\partial z} & 0 & \dfrac{1}{w} & 0 \\ \dfrac{\partial^2 w}{\partial^2 x} & \dfrac{\partial^2 w}{\partial x \partial z} & 0 & 0 & 0 \\ \dfrac{\partial^2 w}{\partial x \partial z} & \dfrac{\partial^2 w}{\partial^2 z} & 0 & 0 & 0 \\ \dfrac{\partial w}{\partial x} & \dfrac{\partial w}{\partial z} & 0 & 0 & 0 \end{pmatrix}; \quad \mathbf{b} = \frac{\partial \mathbf{a}}{\partial \mathbf{m}} = \begin{pmatrix} -\dfrac{p_x}{w^2}\dfrac{\partial w}{\partial \mathbf{m}} \\ -\dfrac{p_z}{w^2}\dfrac{\partial w}{\partial \mathbf{m}} \\ \dfrac{\partial^2 w}{\partial \mathbf{m} \partial x} \\ \dfrac{\partial^2 w}{\partial \mathbf{m} \partial z} \\ \dfrac{\partial w}{\partial \mathbf{m}} \end{pmatrix}, \tag{28}$$

where $\mathbf{y} = \partial \mathbf{x}/\partial \mathbf{m}$ are the derivatives of the ray variables with respect to the model parameters.

System (27) is integrated along with the ray equations for both source and receiver branch of the raypath, and by selecting the appropriate components of $\mathbf{y}$

at the surface ($s = S$), we find the derivatives needed in equation (13):

$$\begin{pmatrix} \frac{\partial x_s}{\partial \mathbf{m}} \\ \frac{\partial x_g}{\partial \mathbf{m}} \\ \frac{\partial t}{\partial \mathbf{m}} \\ \frac{\partial p_h}{\partial \mathbf{m}} \end{pmatrix} = \begin{pmatrix} y_1^S(s = S^S) \\ y_1^G(s = S^G) \\ y_5^S(s = S^S) + y_5^G(s = S^G) \\ -y_3^S(s = S^S) + y_3^G(s = S^G) \end{pmatrix}, \tag{29}$$

where I have used $p_h = p_{x,g} - p_{x,s}$ and $t = t_s + t_g$.

## Derivatives with respect to the depth-point variables: matrix G

For the matrix **G** I follow the same procedure, but instead of taking the derivative of the ray equations with respect to the model parameters, I take the derivative with respect to $x_m$, $z_m$, $\phi$, and $\theta$. As in equation (27), we can derive:

$$\begin{cases} \frac{d\left(\partial \mathbf{x}/\partial q\right)}{ds} = \frac{\partial \mathbf{a}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial q} \\ \frac{\partial \mathbf{x}}{\partial q}(s = 0) = \frac{\partial \mathbf{x}_0}{\partial q} \end{cases} \quad \text{or:} \quad \begin{cases} \frac{d\mathbf{z}_q}{ds} = \mathbf{A}\mathbf{z}_q \\ \mathbf{z}_q(s = 0) = \mathbf{z}_{q,0} \end{cases} \tag{30}$$

Here $\mathbf{z}_q$ is the derivative of the ray vector $\mathbf{x}$ with respect to $q$, where $q$ can be one of $x_m$, $z_m$, $\phi$ or $\theta$.

After integrating the above system, the elements of **G** are found by taking the appropriate $\mathbf{z}_q$ variables at the surface. For example, $G_{11} = \frac{\partial x_s}{\partial x_m}$ is:

$$\frac{\partial x_s}{\partial x_m} = z^S_{1,x_m}(s = S^S) = \frac{\partial x^S}{\partial x_m}(s = S^S). \tag{31}$$

## THE FIVE MOST IMPORTANT PROBLEMS IN GEOPHYSICS

### (informal survey of Stanford faculty and students October 1988)

- Fault processes *(suggested six times)*
- Driving forces of plate tectonics (5)
- Origin of the earth's magnetic field (4)
- Earthquake prediction (3)
- Global climate (3)
- Crustal rheology (3)
- Mountain building mechanism (3)
- Crustal fluids (2)
- Petroleum reservoirs (2)
- Planetary interiors (2)
- Cause of seismic reflections (2)
- Survey aliasing
- Lithospheric stress
- Deep scientific drilling
- Cause of crustal heterogeneities
- Predictability of geophysical phenomenom (chaos)
- Cretaceous—Tertiary boundary
- Laboratory verses geologic dimension scales
- Unification of geophysical datasets
- Symbolic computation
- Public education