Chapter 3

Interval velocity estimation

3.1 OVERVIEW

In the last chapter I formulated the velocity estimation problem as the maximization of beam stacks' semblance at the traveltimes and surface locations predicted by the velocity model. In this chapter I present an optimization procedure for solving the estimation problem. This procedure employs a conjugate-gradient and a Gauss-Newton algorithm. These algorithms, as all optimization algorithms based on the derivatives of the objective function with respect to the model parameters, require the evaluation of a linear operator that relates velocity perturbations to the consequent perturbations in the modeled data. The adjoint of this linear operator is the back-projection operator of the tomographic estimation, and it is applied to evaluate the gradient of the objective function with respect to the model.

The velocity estimation presented in this chapter was successfully tested with synthetic data modeled with use of a finite-difference program. In this test a velocity anomaly was estimated from the reflections off a dipping bed.

3.2 THE TOMOGRAPHIC BACK-PROJECTION OPERATOR

The optimization problem of maximizing beam stacks' semblance can be solved with an iterative procedure. At each step of the procedure the velocity model is moved along a direction that causes an increase of the objective function. Directions of increase can be computed by an evaluation of the derivatives of the objective function with respect to the model. The computation of these derivatives requires the evaluation of a linear operator $\mathbf{G}$ that relates model perturbations to the consequent perturbations in the modeled data.

The modeling functions can be linearized at a fixed reflector position or, at a fixed offset and midpoint, or at a fixed event in the data. The linear operator that I use for the velocity estimation relates the perturbations in the beam stacks' kinematics to perturbations in velocity, at a fixed event in the data. Fitting the data at a fixed event is the most appropriate procedure for seismic data, which contain a discrete set of events.

A velocity perturbation causes movements in the reflectors correspondent to the events in the data. The derivation of the linear operator $\mathbf{G}$ must take into account the movements of the reflectors and thus it must consider the perturbations in the raypaths caused by velocity perturbations. The derivatives of the raypaths with respect to velocity are computed by use of the ray-tracing method presented in Appendix C.

3.2.1 The objective function and its gradient

The goal of the velocity-estimation procedure is to maximize the beam stacks' semblance at the traveltimes and surface location predicted by ray tracing. In equation (2.21) I defined the following objective function for this maximization problem

$$Q_t(\mathbf{m}) = \sum_{y}\sum_{h}\sum_{p_y}\sum_{p_h} \text{Beam}\left(t(y,h,p_y,p_h,\mathbf{m}),y,h,p_y,p_h\right) - (\mathbf{m}-\mathbf{m_0})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_0}). \tag{3.1}$$

The beam stacks' semblance is significantly different from zero only in the proximity of its peaks, which correspond to events in the data. In particular, semblance is different from zero only at a few modeled data-points $(t(y,h,p_y,p_h,\mathbf{m}),y,h,p_y,p_h)$; these points correspond to the *detected* events for the velocity model $\mathbf{m}$. Only these detected events contribute to the value of the objective function and to the evaluation of its gradient. Therefore the objective function can be conveniently rewritten as,

$$Q_t(\mathbf{m}) = \sum_{i} \mathrm{B}_i\left(t_i(\mathbf{m})\right) - (\mathbf{m}-\mathbf{m_0})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_0}), \tag{3.2}$$

where i is the index of the set I of the modeled data-points $(t_i(\mathbf{m}),y,h,p_y,p_h)$ that correspond to detected events in the data. The composite functions $\mathrm{B}_i\left(t_i(\mathbf{m})\right)$ are made of the *fitting* functions $\mathrm{B}_i(t_i)$, which have higher values the closer t_i is to the times of the respective semblance peaks, and the *modeling* functions $t_i(\mathbf{m})$, which depend on the result

of modeling by ray tracing.

To compute the gradient of the objective function with respect to the velocity model I apply the chain rule to the composite functions $\mathrm{B}_i\,(t_i(\mathbf{m}))$ and obtain the expression

$$\begin{aligned}\nabla Q_t &= \sum_i \frac{\partial \mathrm{B}_i\,(t_i(\mathbf{m}))}{\partial \mathbf{m}} - 2{\mathbf{C_m}}^{-1}(\mathbf{m}-\mathbf{m_0}) = \\ &\sum_i \frac{\partial t_i}{\partial \mathbf{m}}\frac{\partial \mathrm{B}_i}{\partial t_i} - 2{\mathbf{C_m}}^{-1}(\mathbf{m}-\mathbf{m_0}) = \\ &\mathbf{G}_t^T\,\mathbf{D}_t - 2{\mathbf{C_m}}^{-1}(\mathbf{m}-\mathbf{m_0}),\end{aligned} \tag{3.3}$$

where the partial derivatives are computed at fixed values of y, h, p_y, and p_h.

The vector $\mathbf{D}_t$ is easily computed from beam-stacked data with a finite-difference approximation of the first-derivative operator. The components of the vector $\mathbf{D}_t$ are proportional to the distances between the modeled traveltimes t_i and the traveltimes of the semblance peaks. They are thus equivalent to the residuals in a least-squares minimization. The matrix of the Frechet derivatives $\mathbf{G}_t$ is a linearization of the modeling functions and relates the perturbation $\Delta\mathbf{m}$ in the velocity model to the consequent perturbation Δt_i in traveltimes; that is, $\Delta t_i = \mathbf{G}_t\Delta\mathbf{m}$. The gradient is computed by the application of the transpose $\mathbf{G}_t^T$ to the vector of the derivatives $\mathbf{D}_t$. The matrix $\mathbf{G}_t^T$ is a back-projection operator because it back projects the *residuals* $\mathbf{D}_t$ into the velocity model.

The computation of the gradient by use of equation (3.3) implies a fitting of the modeled traveltimes to the data traveltimes, at constant offset, midpoint and ray parameters. This formulation for the gradient is the most straightforward, given the modeling algorithm that I presented in section 2.3.1, but it is not the only one possible, and actually it is not the best one. The major problem with fitting traveltimes is that varying the traveltime at constant offset and midpoint implies a change of the event in the data. Therefore a perturbation in the velocity model causes the perturbed modeled data-point $(t_i + \mathbf{G}_t\Delta\mathbf{m}, y, h, p_y, p_h)$ to correspond to an event in the data different from the event correspondent to the unperturbed data-point (t_i, y, h, p_y, p_h). This effect can cause instability in the maximization procedure when the events in the data, that is the semblance peaks in the beam-stacked data, are sparse. In such a case, there are many modeled data-points $t_i(\mathbf{m})$ that are unconstrained and thus the maximization algorithm can force these unconstrained modeled data-points to concentrate around the same semblance peak. The

undesired result of this concentration is to explain one event with many modeled data-points, whereas, the desired correspondence is a one-to-one. This problem can be solved by a change in the linearization of the modeling function used for computing the gradient. Instead of keeping fixed the midpoints and offsets of the modeled data-points, the events in the data are kept fixed. The new linearization preserves the desired one-to-one correspondence between modeled data-points and events in the beam-stacked data, because a velocity-model perturbation causes the modeled data-points to follow the events in the data.

The new gradient computation is easily accomplished after a transformation of the traveltime axis; this transformation is applied to the beam-stacked data and to the result of modeling. In a beam stack with offset ray parameter p_h, the events are, by definition, parallel to the line of the equation $t = p_h h$. To follow an event, the modeled data-points must move along dipping lines, with time dip equal to p_h. These dipping lines are lines at a constant transformed traveltime τ when the time axis is transformed accordingly to the relation

$$\tau = t - p_h h. \tag{3.4}$$

The events in the transformed data can be fitted with variations in the offset h at constant transformed traveltime τ. The beam-stacked data are transformed according to the relation

$$Beam\,(\tau, y, h, p_y, p_h) = \mathrm{Beam}\,(\tau + p_h h, y, h, p_y, p_h)\,, \tag{3.5}$$

and the modeling by ray tracing can be conveniently represented by the function

$$h = h(\tau, y, p_y, p_h, \mathbf{m}). \tag{3.6}$$

Figure 3.1 shows the semblance panels computed by beam stacking the CMP gather shown in Figure 2.1 before (left) and after (right) the data have been transformed according to the the coordinate transformation of equation (3.4). The events in the data have constant traveltime after the transformation. The two curves superimposed onto the beam stacks have been computed by ray tracing through the same velocity model, but for obtaining the curve on the right, the result of ray tracing has been transformed with equation (3.4).

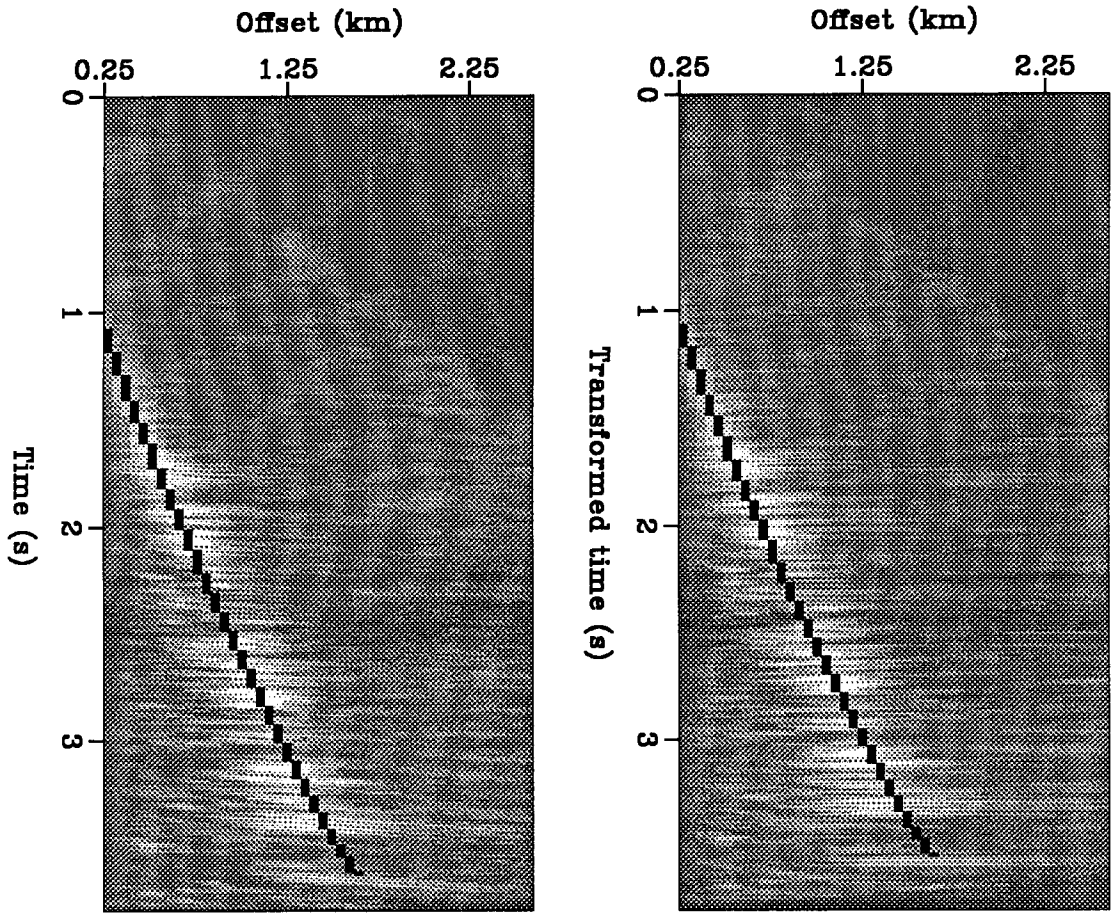

FIG. 3.1. Semblance panels computed by beam stacking the CMP gather shown in Figure 2.1 before (left) and after (right) the coordinate transformation of equation (3.4). The events in the data have constant traveltime after the transformation. Superimposed onto the beam stack are the traveltime-offset curves resulting from ray tracing.

In this new coordinate system the objective function in equation (3.1) becomes

$$Q_h(\mathbf{m}) = \sum_{\tau}\sum_{y}\sum_{p_y}\sum_{p_h} Beam\left(\tau, y, h(\tau, y, p_y, p_h, \mathbf{m}), p_y, p_h\right) - (\mathbf{m}-\mathbf{m_0})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_0}). \tag{3.7}$$

When only the detected events I are considered, this new objective function can be rewritten as

$$Q_h(\mathbf{m}) = \sum_{i} B_i\left(h_i(\mathbf{m})\right) - (\mathbf{m}-\mathbf{m_0})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_0}). \tag{3.8}$$

The gradient of this new objective function can be calculated as is the gradient in equation (3.3), by use of the following equation

$$\nabla Q_h = \sum_i \frac{\partial B_i\,(h_i(\mathbf{m}))}{\partial \mathbf{m}} - 2\mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_0}) =$$

$$\sum_i \frac{\partial h_i}{\partial \mathbf{m}} \frac{\partial B_i}{\partial h_i} - 2\mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_0}) =$$

$$\mathbf{G}_h^T\,\mathbf{D}_h - 2\mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_0}), \tag{3.9}$$

where the partial derivatives are now computed at fixed values of τ, y, p_y, and p_h.

As in the previous formulation the vector $\mathbf{D}_h$ is equivalent to the vector of residuals and the matrix $\mathbf{G}_h^T$ is the back-projection operator. The problem of evaluating this back-projection operator will be addressed in the next section.

The objective function $Q_t(\mathbf{m})$ expressed in equation (3.2) and the objective function $Q_h(\mathbf{m})$ expressed in equation (3.8) are clearly equivalent, but the two gradients ∇Q_t and ∇Q_h are different because the detected events are a discrete mesh in the data space. There would be no difference between the two gradients if continuous integrals were substituted for the discrete summations in the expressions of the gradients. The application of the gradient of equation (3.9) yield a more robust estimation process than the application of the gradient of equation (3.3) because of the discrete nature of seismic events.

An additional advantage of using the objective function of equation (3.8) instead of equation (3.2) is that beam-stacks' offset are more sensitive to velocity perturbations than are beam stacks' traveltimes, as I discussed in section 2.4. Therefore the inversion of the linear operator $\mathbf{G}_h$ is more robust of the inversion of $\mathbf{G}_t$. Noise in the measurements of the semblance derivative vector $\mathbf{D}_h$ will have less effect on the search direction than noise does in the measurements of $\mathbf{D}_t$.

3.2.2 Computation of the back-projection operator

To compute the gradient of the objective function with respect to the velocity model, the vector of semblance derivatives $\mathbf{D}_h$ is back projected into the model space by use of the linear operator $\mathbf{G}_h^T$. In this section I derive the basic formula for computing the matrix of the Frechet derivatives $\mathbf{G}_h$.

The matrix $\mathbf{G}_h$ is the linearization of the modeling function $h(\tau, y, p_y, p_h, \mathbf{m})$ at fixed

events in the data. More precisely, it relates velocity perturbations to perturbations in beam stacks' offsets, at constant transformed traveltimes and midpoints. A perturbation in the velocity model causes perturbations in the raypaths and consequently it causes movements of the reflectors correspondent to the events in the data. Because of these reflectors' movements, the raypaths' perturbations must be taken into account when the derivatives of offset at fixed events are being calculated. The need for considering the perturbations in the raypaths distinguishes the beam-stack back-projection operator from other tomographic back-projection operators (Bishop et al, 1985; Stork, 1988) and make it similar to the back-projection operator used by Sword (1987) for inverting CDR data. In classical tomography, traveltimes are computed by use of two-point ray tracing; because the Fermat principle can be invoked raypaths' perturbations can be neglected. The derivatives of the raypaths with respect to the velocity model are computed with the ray-tracing method presented in Appendix C.

In section 2.3.1 I presented an algorithm for modeling the beam-stacked data by combining the results of an initial-value ray tracing. Rays are traced from the shot position and the receiver position until they meet at the reflector $R(x,z,\theta)$. At fixed ray parameters p_y and p_h, the result of modeling is the manifold $h(\tau,y,\mathbf{m})$ defined in the data space; each point of this manifold corresponds to a reflector $R(x,z,\theta)$. Instead of starting from an assigned mesh of surface locations, combining the results of ray tracing can start from an assigned mesh of reflector positions. For each reflector position (x,z) the sets of down-going and up-going rays are interpolated and combined to generate the two rays meeting at (x,z). The transformed traveltime, the midpoint, and the offset correspondent to each reflector position, can then be easily computed from the traveltimes and the starting surface locations of the interpolated rays. If necessary the dip θ at the reflector position can also be easily estimated.

Therefore the modeling establishes a mapping, function of the velocity model $\mathbf{m}$, from the reflector space (x,z) into the data space (τ,y,h). The mapping can be expressed by the triplet of functions $[\tau(x,z,\mathbf{m}),y(x,z,\mathbf{m}),h(x,z,\mathbf{m})]$. These functions can be combined for evaluating the offset as a function of traveltime and midpoint; the composite function is $h\left(\tau(x,z,\mathbf{m}),y(x,z,\mathbf{m}),\mathbf{m}\right)$. Differentiating this composite function with respect to the velocity model, by use of the chain rule, I obtain

$$\left.\frac{\delta h}{\delta \mathbf{m}}\right|_{(x,z)} = \left.\frac{\partial h}{\partial \tau}\right|_{(y,\mathbf{m})} \left.\frac{\delta \tau}{\delta \mathbf{m}}\right|_{(x,z)} + \left.\frac{\partial h}{\partial y}\right|_{(\tau,\mathbf{m})} \left.\frac{\delta y}{\delta \mathbf{m}}\right|_{(x,z)} + \left.\frac{\partial h}{\partial \mathbf{m}}\right|_{(y,\tau)} . \qquad (3.10)$$

The only unknowns in this equation are the Frechet derivatives $\partial h/\partial \mathbf{m}$. The partial derivatives $\partial h/\partial \tau$ and $\partial h/\partial y$ can be evaluated at a constant velocity model, but with a varying reflector position, on the manifold $h(\tau, y)$ defined by ray tracing. The partial derivatives $\delta h/\delta \mathbf{m}$, $\delta \tau/\delta \mathbf{m}$, and $\delta y/\delta \mathbf{m}$ can be computed at constant reflector position (x, z), by use of the ray tracing presented in Appendix C. Equation (3.10) can thus be solved for $\partial h/\partial \mathbf{m}$ and rewritten as

$$\left.\frac{\partial h}{\partial \mathbf{m}}\right|_{(y,\tau)} = \left.\frac{\delta h}{\delta \mathbf{m}}\right|_{(x,z)} - \left.\frac{\partial h}{\partial \tau}\right|_{(y,\mathbf{m})} \left.\frac{\delta \tau}{\delta \mathbf{m}}\right|_{(x,z)} - \left.\frac{\partial h}{\partial y}\right|_{(\tau,\mathbf{m})} \left.\frac{\delta y}{\delta \mathbf{m}}\right|_{(x,z)} . \tag{3.11}$$

This equation allows us to evaluate the elements of the back-projection matrix $\mathbf{G}_h^T$; these elements are the combination of terms that depend on raypaths perturbations ($\delta h/\delta \mathbf{m}$, $\delta \tau/\delta \mathbf{m}$, and $\delta y/\delta \mathbf{m}$), with terms that are related to the perturbations in the modeling function caused by reflectors' movement ($\partial h/\partial \tau$ and $\partial h/\partial y$). In Appendix E equation (3.11) is rederived in way that directly relates the reflectors' movements to the derivatives used to compute $\partial h/\partial \mathbf{m}$. That Appendix also shows how the reflectors' movements are determined by imposing the constraint of keeping constant the transformed traveltime and the midpoint; that is, by constraining the modeled data-points to follow an event in the data.

Figure 3.2 and Figure 3.4 show examples of the back-projection operator $\mathbf{G}_h^T$ computed by use of equation (3.11). Each row of the matrix $\mathbf{G}_h^T$ corresponds to a velocity model parameter, while each column corresponds to an event in the data. The figures show a column of $\mathbf{G}_h^T$; that is, they show the amplitudes of the operator for a given event in the data as a function of the location of the model perturbation. The velocity model is parametrized in slowness with B-spline functions (Appendix D). The operator appears to be smooth because the figures show the operator resampled on a grid much denser than the original B-spline parametrization.

Figure 3.2 shows the amplitude of a column of $\mathbf{G}_h^T$ when the background velocity is constant and equal to 2 km/s. The midpoint ray parameter is zero and the offset ray parameter is .06 s/km. The operator is non-zero in a band around the down-going and the up-going rays. Above the rays the operator is positive (light in Figure 3.2) and below it is negative (dark in Figure 3.2). A slowness anomaly above the ray makes the ray travel more horizontally, and therefore causes an increase in the offset; below the ray an anomaly makes the ray travel more vertically and thus causes a decrease in the offset. These effects are in agreement with the offset perturbations caused by a velocity anomaly

in the beam stacks of the synthetic data presented in section 2.4. The amplitudes of the operator decrease with depth because the effects of ray bending on beam stacks' offset increase with the distance traveled by the bent ray.

Figure 3.3 shows the cross-section of Figure 3.2 taken at a constant depth of 180 m. The amplitudes of the operator are antisymmetric around the rays and they look like the amplitudes of a smoothed "first derivative" operator.

Figure 3.4 shows the amplitude of a column of $\mathbf{G}_h^T$ when the background velocity has a constant lateral gradient of 1 s^{-1}, with velocity increasing from left to right. The midpoint ray parameter is zero and the offset ray parameter is .06 s/km. The lateral gradient in velocity causes ray bending and consequently the operator bends with the rays. The dip of the reflector is caused by the ray bending. Figure 3.5 shows the cross-section of Figure 3.4 taken at constant depth of 180 m.

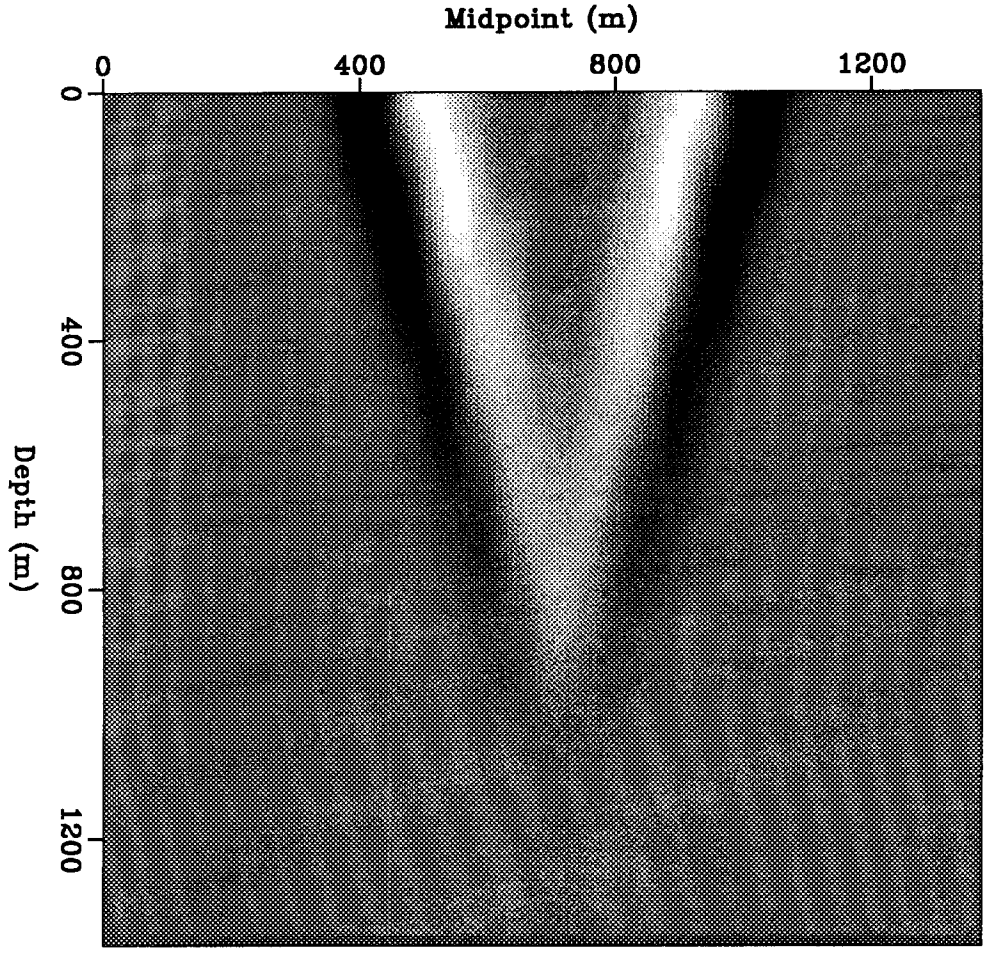

FIG. 3.2. The back-projection operator $\mathbf{G}_h^T$ for one event in the data as a function of the location of the slowness anomaly. The midpoint ray parameter is zero and the offset ray parameter is .06 s/km. The operator is spatially localized around the rays. Light areas indicate positive amplitude and dark areas negative amplitude.

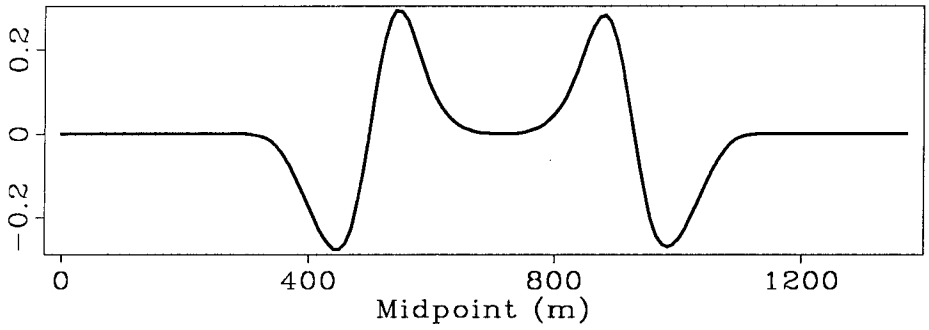

FIG. 3.3. A cross-section of the back-projection operator shown above taken at a depth of 180 m. The operator is antisymmetric around the rays and looks like a smoothed *first derivative* operator.

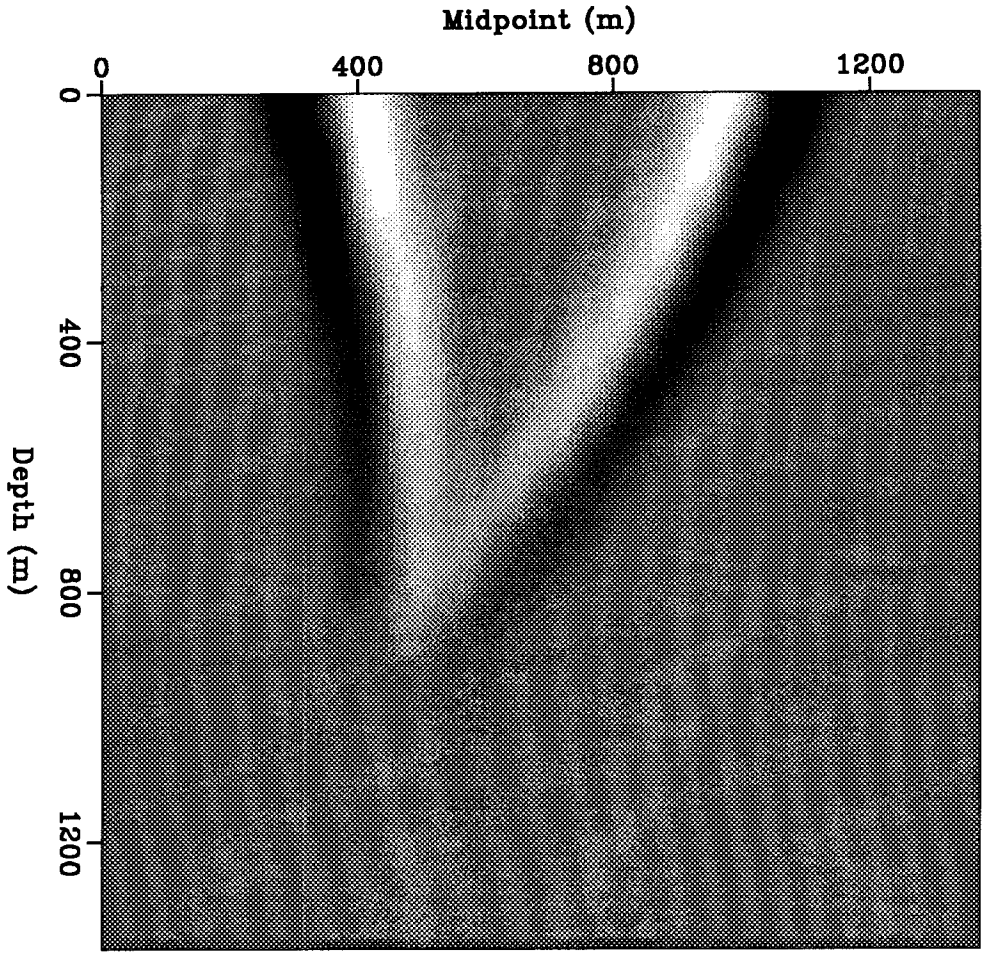

FIG. 3.4. The back-projection operator $\mathbf{G}_h^T$ for one event in the data when the background velocity has a constant lateral gradient of 1 s^{-1}. The midpoint ray parameter is zero and the offset ray parameter is .06 s/km. The dip of the reflector is the effect of ray bending caused by the lateral gradient in velocity.

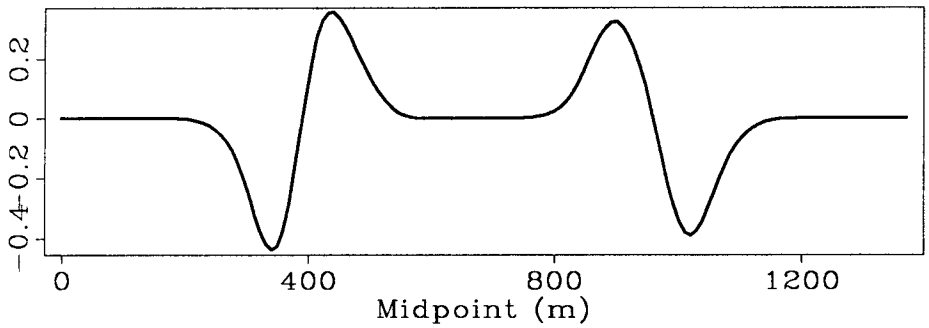

FIG. 3.5. A cross-section of the back-projection operator shown above taken at a depth of 180 m.

3.3 ALGORITHMS FOR SOLVING THE OPTIMIZATION PROBLEM

The maximization of the objective function of equation (3.8) is a hard optimization problem because it presents difficulties of global convergence as well as of local convergence. The objective function has many local maxima and it is highly non-quadratic. At the beginning of the optimization procedure, when the velocity model is distant from the true model, the first priority is to converge towards the true model and to avoid the local maxima. The non-quadratic behavior of the objective function is mostly determined by the low-wavenumber components of the velocity model. Therefore to start the estimation, I apply a conjugate-gradient algorithm that converges to a robust estimate of the low-wavenumber components of the velocity model. When the low-wavenumber components of the solution are close to the true model the objective function is well approximated by a quadratic. This locally quadratic problem is not well conditioned because some high-wavenumber components of the model are poorly determined, as in all tomographic estimation. At this stage of the estimation procedure I apply a Gauss-Newton algorithm that exploits information on the second derivatives of the objective function, for rapidly converging to the best estimates of the velocity model that is consistent with the beam-stacked data and the a priori assumptions.

3.3.1 Global convergence–Conjugate gradient algorithm

The objective function [equation (3.8)] is the sum of the composite functions $B_i(h_i(\mathbf{m}))$, which are composed of the modeling functions $h_i(\mathbf{m})$ and the fitting functions $B_i(h)$. The objective function would be quadratic if the modeling functions were linear and the fitting functions were parabolic. These conditions do not hold in field data applications and therefore the objective function is non-quadratic and it has many local maxima.

The primary cause of the non-quadratic behavior of the objective function is the non-parabolicity of the fitting functions $B_i(h)$; that is, beam stacks' semblance as a function of offset. Semblance usually has isolated peaks and it is not significantly different from zero in between the peaks. In the neighborhoods of the peaks, semblance can be well approximated with parabolas, but away from the peaks it is not parabolic at all. Semblance peaks correspond to events in the data; an event is *detected* when its offset is approximately predicted by the current velocity model. Until the most of the events are detected, the fitting

functions are non-parabolic, the objective function is not-quadratic, and the convergence of the optimization process is slow. The same problem is aggravated by the non-linearity of the modeling functions $h_i(\mathbf{m})$. The modeling functions are evaluated from the results of ray tracing (section 2.3.1). Ray bending and the reflectors' movements make the modeling functions non-linear. The effects of ray bending and reflectors' movements are taken into account in the linearization of the modeling function represented by the linear operator $\mathbf{G}_h$ (section 3.2.2), but the simple linear model is not valid when the velocity perturbations are large.

The non-quadratic behavior objective function, which causes a slow convergence of the optimization process, is mostly determined by the low-wavenumber components of the velocity model. These low-wavenumber components of the model strongly influence ray tracing, and therefore they determine the detection of the events and the non-linearity of the modeling functions. The convergence can be improved if the estimation is started from the low-wavenumber components of the velocity model. Furthermore, the low-wavenumber components of the model can be parametrized with only few B-spline functions. Reducing the number of model parameters has the double advantage of reducing the number of iterations needed for the solution to converge and of decreasing the cost of each iteration.

Starting the estimation from the low-wavenumber components of the model, and slowly increasing its bandwidth, is also an heuristic solution to the problem of local maxima caused by multiples (Toldi, 1985). The majority of the semblance peaks correspond to the primary reflections, but others correspond to multiples and artifacts. Multiples and artifacts cause local maxima in the objective function that should be avoided by the optimization process. Multiples influence the estimation process only if they have been detected by the current velocity model. Primary and multiples can be usually detected by the same interval velocity model only if the model is rapidly changing. The idea is to constraint the solution to be stiff until the estimation has detected the primaries; the constraints are softened when the current solution cannot be influenced by the multiples.

A robust optimization algorithm, with strong global-convergence properties, must be used for solving such a non-quadratic problem. Among the optimization algorithms based on the derivatives of the objective function, the ones utilizing the first derivatives have the best global-convergence properties. I choose a version of the conjugate-gradient algorithm derived by Polak-Ribiere (Luenberger, 1984) that is particularly efficient in solving non-quadratic problems. The scheme of the algorithm that I use for starting the estimation

process is thus

Set initial parametrization and starting model $\mathbf{m_0} = \bar{\mathbf{m}}$.

Set $k = 0$.

Compute by ray tracing $\mathbf{h}_0 = \mathbf{h}(\mathbf{m_0})$ and $\mathbf{G}_{h,0}(\mathbf{m_0})$.

Compute by finite difference $\mathbf{D}_{h,0}(\mathbf{h}_0)$.

Compute the gradient $\mathbf{g}_0 = \nabla Q_h = \mathbf{G}_{h,0}^T \mathbf{D}_{h,0} - 2\mathbf{C_m}^{-1}(\mathbf{m_0} - \bar{\mathbf{m}})$.

Set the search direction $\delta\mathbf{m_0} = \mathbf{g}_0$.

Find α_0 that maximizes $Q_h\left(\mathbf{h}(\mathbf{m_0} + \alpha_0 \delta\mathbf{m_0})\right)$.

Update the model $\mathbf{m}_1 = \mathbf{m_0} + \alpha_0 \delta\mathbf{m_0}$.

Set $k = k + 1$.

Compute by ray tracing $\mathbf{h}_k = \mathbf{h}(\mathbf{m}_k)$ and $\mathbf{G}_{h,k}(\mathbf{m}_k)$.

Compute by finite difference $\mathbf{D}_{h,k}(\mathbf{h}_k)$.

Compute the gradient $\mathbf{g}_k = \nabla Q_h = \mathbf{G}_{h,k}^T \mathbf{D}_{h,k} - 2\mathbf{C_m}^{-1}(\mathbf{m}_k - \bar{\mathbf{m}})$.

Compute the search direction $\delta\mathbf{m}_k = \mathbf{g}_k + \frac{(\mathbf{g}_k - \mathbf{g}_{k-1})^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \delta\mathbf{m}_{k-1}$.

Find α_k that maximizes $Q_h\left(\mathbf{h}(\mathbf{m}_k + \alpha_k \delta\mathbf{m}_k)\right)$.

Update the model $\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \delta\mathbf{m}_k$.

Check for convergence.

Increase the bandwidth of the parametrization.

Set $\mathbf{m_0} = \mathbf{m}_{k+1}$.

Check for convergence.

The low-wavenumber components of the velocity model usually converge close to the true model after few iterations of the previous algorithm. At this stage, the perturbations of the low-wavenumber components of the model are rather small and the modeling functions can be considered linear, because the first-order effects of the ray bending and the reflector's movement were taken into account when the linear operator $\mathbf{G}_h$ was evaluated. If the modeling functions are linear, the operator $\mathbf{G}_h$ is independent of the model, and the

offset perturbations $\Delta\mathbf{h}$, caused by the model perturbations $\Delta\mathbf{m}$, are computed by use of the simple linear relation $\Delta\mathbf{h} = \mathbf{G}_h\Delta\mathbf{m}$. The advantage of this linearization is a saving of the great computational effort that in the previous algorithm is taken by the ray tracing necessary at each iteration for modeling and evaluating the linear operator $\mathbf{G}_h$. (The matrix $\mathbf{G}_h$ can be stored on disk and used for many iterations of the conjugate-gradient algorithm. The matrix $\mathbf{G}_h$ is usually small enough that is practical to store it on disk because the model has been parsimoniously parametrized with B-spline functions. The matrix would be much larger if the model were less efficiently parametrized.)

The following algorithm can thus be substituted for the inner loop of the previous algorithm.

Set $k = k + 1$.

Compute $\mathbf{h}_k = \mathbf{h}_{k-1} + \alpha_{k-1}\mathbf{G}_{h,0}\delta\mathbf{m}_{k-1}$.

Compute by finite difference $\mathbf{D}_{h,k}(\mathbf{h}_k)$.

Compute the gradient $\mathbf{g}_k = \nabla Q_h = \mathbf{G}_{h,0}^T\mathbf{D}_{h,k} - 2{\mathbf{C}_\mathbf{m}}^{-1}(\mathbf{m}_k - \bar{\mathbf{m}})$.

Compute the search direction $\delta\mathbf{m}_k = \mathbf{g}_k + \frac{(\mathbf{g}_k - \mathbf{g}_{k-1})^T\mathbf{g}_k}{\mathbf{g}_{k-1}^T\mathbf{g}_{k-1}}\delta\mathbf{m}_{k-1}$.

Find α_k that maximizes $Q_h\left(\mathbf{h}_k + \alpha_k\mathbf{G}_{h,0}\delta\mathbf{m}_k\right)$.

Update the model $\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k\delta\mathbf{m}_k$.

Check for convergence.

The line search is an important part of these algorithms. For accurately estimating the step size, there are many methods that require a limited number of function evaluations. I used an iterative line-search algorithm that fits a quadratic function to the objective function and evaluates the objective function at the maximum of this quadratic curve (Luenberger, 1984).

The algorithms presented in this section could be used until complete convergence of the optimization process. But, when the current solution is sufficiently close to the true velocity model, the efficiency of the procedure can be improved by using the Gauss-Newton algorithm presented in the next section.

3.3.2 Local convergence–Gauss-Newton algorithm

The fitting functions can be well approximated by parabolas in the proximity of the semblance peaks. Therefore the optimization problem is approximately quadratic when the low-wavenumber components of the model have been determined and all the primaries have been detected. This locally quadratic problem is ill-conditioned, because some high-wavenumber components of the model are poorly determined. In any tomographic estimation the high-wavenumber components of the model are less determined than the low-wavenumber components because they have a smaller effect on ray-tracing and the ray coverage is limited. Some of the high-wavenumber components are completely undetermined by the data, but others can be recovered if the estimation process is allowed to converge fully. Because the problem is approximately quadratic, the convergence of the optimization procedure can be substantially improved when the information on the second derivative of the objective function is used.

The Hessian of the objective function can be derived from the gradient in equation (3.9):

$$\begin{aligned}\mathbf{H}(\mathbf{m}) = \frac{\partial^2 Q_h}{\partial \mathbf{m}^2} &= \sum_i \frac{\partial^2 B_i\left(h_i(\mathbf{m})\right)}{\partial \mathbf{m}^2} - 2\mathbf{C_m}^{-1} = \\ &\sum_i \frac{\partial^2 h_i}{\partial \mathbf{m}^2}\frac{\partial B_i}{\partial h_i} + \sum_i \frac{\partial h_i}{\partial \mathbf{m}}\frac{\partial^2 B_i}{\partial h_i^2}\frac{\partial h_i}{\partial \mathbf{m}} - 2\mathbf{C_m}^{-1} = \\ &\mathbf{S}\mathbf{D}_h + \mathbf{G}_h^T\,\mathbf{D}_2\mathbf{G}_h - 2\mathbf{C_m}^{-1},\end{aligned} \tag{3.12}$$

where $\mathbf{D}_2$ is a diagonal matrix containing the second derivatives of semblance with respect to offset.

The first term in the Hessian depends on the second derivatives of ray tracing with respect to the velocity model. I drop this term because it is too expensive to compute. Dropping this term implies the approximation of ray tracing with a linear function, which is a sensible approximation, as I discussed above. Another reason for neglecting the first term is that it also depends on $\mathbf{D}_1$, that decreases as the velocity model approaches the true velocity model.

The approximate Hessian can be used in a classical quasi-Newton step (Gill et al., 1981) and thus we find the search direction by solving the following linear system:

$$\left[\mathbf{G}_h^T(-\mathbf{D}_2)\mathbf{G}_h + 2\mathbf{C_m}^{-1}\right]\delta\mathbf{m} = \mathbf{G}_h^T\mathbf{D}_h - 2\mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_0}). \tag{3.13}$$

The solution of this system is equivalent to the least-squares solution of the system

$$\begin{pmatrix} \sqrt{-\mathbf{D}_2}\mathbf{G}_h \\ \sqrt{2\mathbf{C}_\mathbf{m}^{-1}} \end{pmatrix} \delta\mathbf{m} \simeq \begin{pmatrix} (\sqrt{-\mathbf{D}_2})^{-1}\mathbf{D}_h \\ -\sqrt{2\mathbf{C}_\mathbf{m}^{-1}}(\mathbf{m}-\mathbf{m}_0) \end{pmatrix}, \tag{3.14}$$

that can be efficiently solved using a conjugate-gradient algorithm such as LSQR (Paige and Saunders, 1982).

The scheme of the Gauss-Newton method that I used is

Set starting model $\mathbf{m}_0$.
Set k = 0.
Compute by ray tracing $\mathbf{h}_0 = \mathbf{h}(\mathbf{m}_0)$ and $\mathbf{G}_{h,0}(\mathbf{m}_0)$.

- Compute by finite difference $\mathbf{D}_{h,k}(\mathbf{h}_k)$ and $\mathbf{D}_{2,k}(\mathbf{h}_k)$.
- Compute the search direction by least−squares
  $$\delta\mathbf{m}_k = \left[\mathbf{G}_{h,0}^T(-\mathbf{D}_{2,k})\mathbf{G}_{h,0} + 2\mathbf{C}_\mathbf{m}^{-1}\right]^{-1}\left[\mathbf{G}_{h,0}^T\mathbf{D}_{h,k} - 2\mathbf{C}_\mathbf{m}^{-1}(\mathbf{m}_k - \bar{\mathbf{m}})\right].$$
- Find α_k that maximizes $Q_h\left(\mathbf{h}_k + \alpha_k\mathbf{G}_{h,0}\delta\mathbf{m}_k\right)$.
- Update the model $\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k\delta\mathbf{m}_k$.
- Check for convergence.
- Set $k = k+1$.
- Compute $\mathbf{h}_k = \mathbf{h}_{k-1} + \alpha_{k-1}\mathbf{G}_{h,0}\delta\mathbf{m}_{k-1}$.

The approximation of the Hessian used for computing the search direction [equation (3.13)] is similar to the approximation that is made in the development of the Gauss-Newton algorithm for solving a non-linear least-squares problem (Gill et al., 1981); therefore in the following discussion I will refer to this algorithm as the Gauss-Newton algorithm. The computation of the search direction using the system of equation (3.14) can also be seen as a weighted dumped least-squares inversion of the forward modeling operator $\mathbf{G}_h$. The weights are determined by the semblance second derivatives $\mathbf{D}_2$. The least-squares inverse is applied to the vector of residuals $\mathbf{D}_2^{-1}\mathbf{D}_h$, which are equal to the distances of the modeled data-points from the semblance peaks.

The second derivatives $\mathbf{D}_2$ estimated from the data might be not reliable enough to be used in equation (3.14). In this case the second derivatives can be set to be inversely proportional to the depth of the reflectors, because the width of the peaks in the beam stacks is proportional to the width of the Fresnel zones of the reflections.

3.3.3 Synthetic example

I tested the estimation method on the synthetic data-set that I presented in section 2.4. The data were modeled by a finite-difference program assuming a constant slowness background of .4 s/km and a circular slowness anomaly with minimum slowness of .357 s/km. The slowness model is estimated with the reflections from a single dipping reflector. The reflector is below the anomaly and is dipping at 20°. Figure 3.6 shows the slowness function assumed for modeling the data.

The data were beam-stacked according to six offset ray parameters p_h: from p_h equal to .04 s/km to p_h equal to .095 s/km. Along the midpoint direction the data were decomposed according to seven midpoint ray parameters p_y: from p_y equal to -.333 s/km to p_y equal to $-.273$ s/km. The beam-stacked data were transformed according to the time transformation defined by equation 3.4 and smoothed along the time and midpoint axes with a Gaussian window. The slowness model was parametrized with B-spline functions, using one basis function every 70 m in the vertical direction and one every 150 m in the horizontal direction. The nature and the size of the problem did not require that the parametrization of the model be changed during the estimation process.

To start the slowness estimation, I applied the conjugate-gradient algorithm presented in section 3.3.1. The starting model was the constant background slowness of .4 s/km. Figure 3.7 shows the solution produced by 2 iterations of the conjugate-gradient algorithm; the intensity scale in this figure is the same as in the figure showing the true model (Figure 3.6). After 2 iterations the anomaly has been roughly localized in the horizontal direction but it is smeared in the direction perpendicular to the reflector. In this direction the model is poorly resolved because of the ray coverage of the anomaly is limited in angle. However, the anomaly can be better resolved if all the information contained in the data is exploited. At this stage of the estimation the objective function is approximately quadratic because the model is close to the true model and the predicted offsets are close to the offsets of the beam stacks' peaks. I thus switched to the Gauss-Newton algorithm described in section 3.3.2. The solution produced by 2 iterations of this algorithm is shown

in Figure 3.8. Figure 3.9 shows a horizontal cross section of the slowness model shown in Figure 3.8 (dashed line) compared with the true slowness (solid line). The cross sections were taken at the center of the slowness anomaly; that is, at a depth of 700 m.

The estimated anomaly is fairly well localized, although it is still smeared in the direction perpendicular to the reflector and it has two negative side-lobes. These limitations in the resolutions are caused by the null space in the relation between the model and the data (Stork, 1988; Fowler, 1988). The null space is reduced because I used multiple ray parameters to describe the non-hyperbolic moveouts in the data, but it is not empty because of the limited ray coverage of the anomaly. The ray coverage depends on the cable length (1500 m) and on the reflectors' geometry. The resolution of the estimation would improve if there were more than one reflector and more than one reflectors' dip, as in the real data example presented in the next chapter.

3.4 CONCLUSIONS

The velocity model can be estimated from beam-stacked data by the solution of the optimization problem formalized in Chapter 2. The estimation procedure requires the evaluation of a linear operator that relates perturbations in the velocity model to the consequent variations in the beam stacks' kinematics. I thus derived an operator that can be used for computing the changes in the beam stacks' offsets of a fixed event in the data, caused by a given velocity perturbation. Consistently following the same event in the data is necessary for the stability of the estimation process, because seismic data contain a discrete set of events.

The optimization problem for estimating velocity is non-quadratic and ill-conditioned. In this chapter I presented an optimization procedure for solving this hard problem. The procedure starts with a conjugate-gradient algorithm that converges close to the true model and then it switches to a a Gauss-Newton algorithm that resolves the less determined components of the model.

The proposed algorithms have been successful in the estimation of a velocity anomaly from synthetic data. Although the velocity model is poorly determined by the data my tomographic estimation has well focused the velocity anomaly.

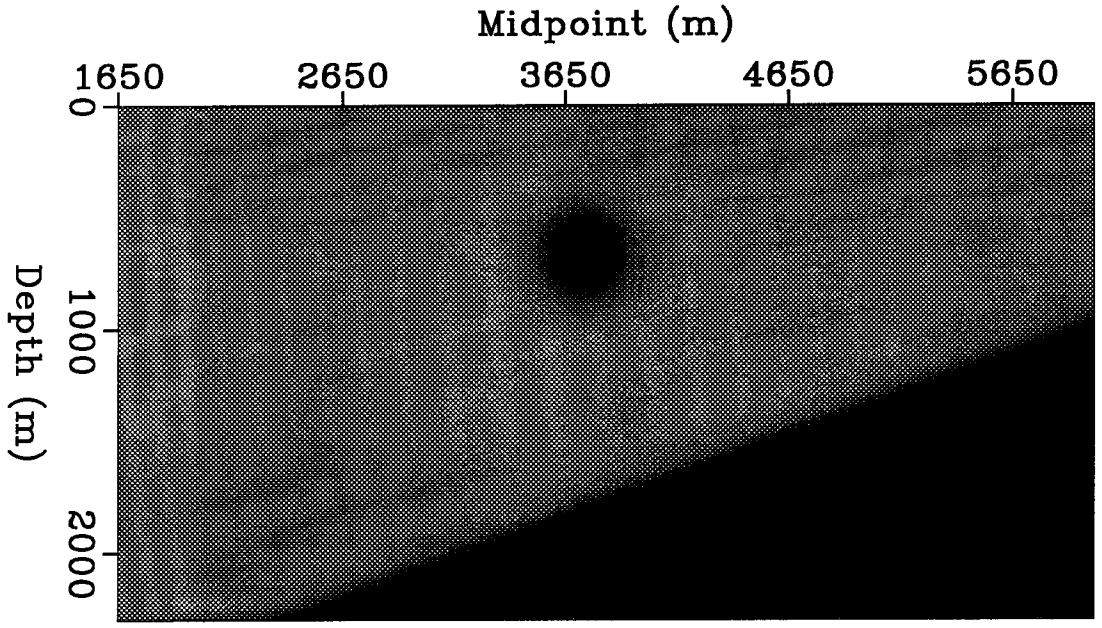

FIG. 3.6. The slowness model used to model the synthetic data. The background slowness is .4 s/km and the circular slowness anomaly is a Gaussian function with minimum slowness of .357 s/km. The dipping reflector has a dip angle of 20°.

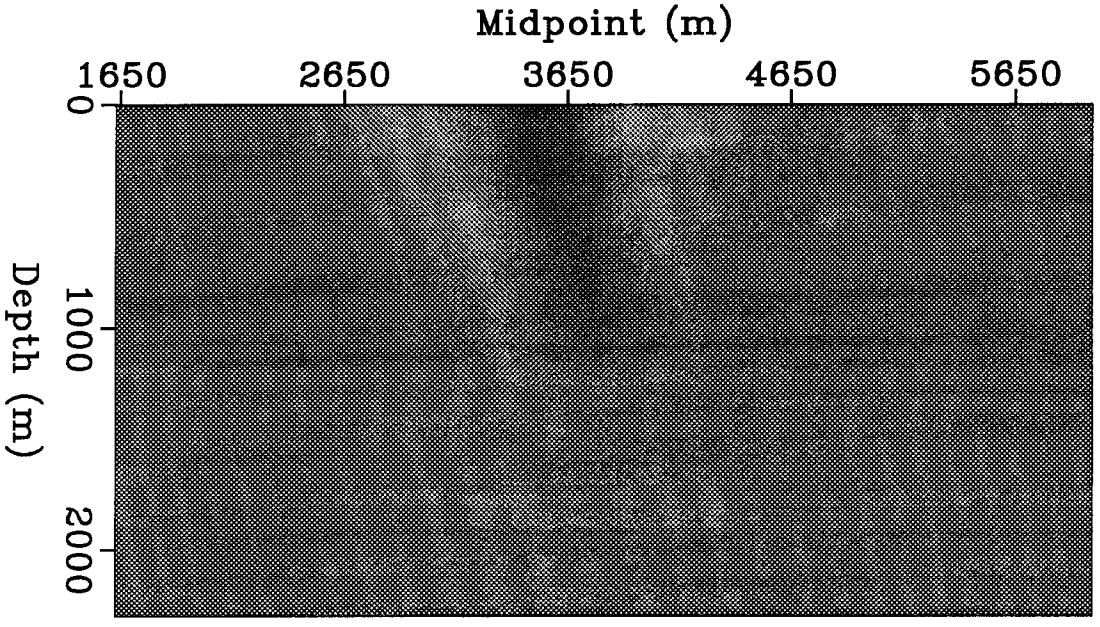

FIG. 3.7. The slowness model after 2 iterations of the conjugate-gradient algorithm presented in section 3.3.1; the intensity scale in this figure is the same as in the figure showing the true model (above). The anomaly has been roughly localized in the horizontal direction but it is smeared in the direction perpendicular to the reflector.

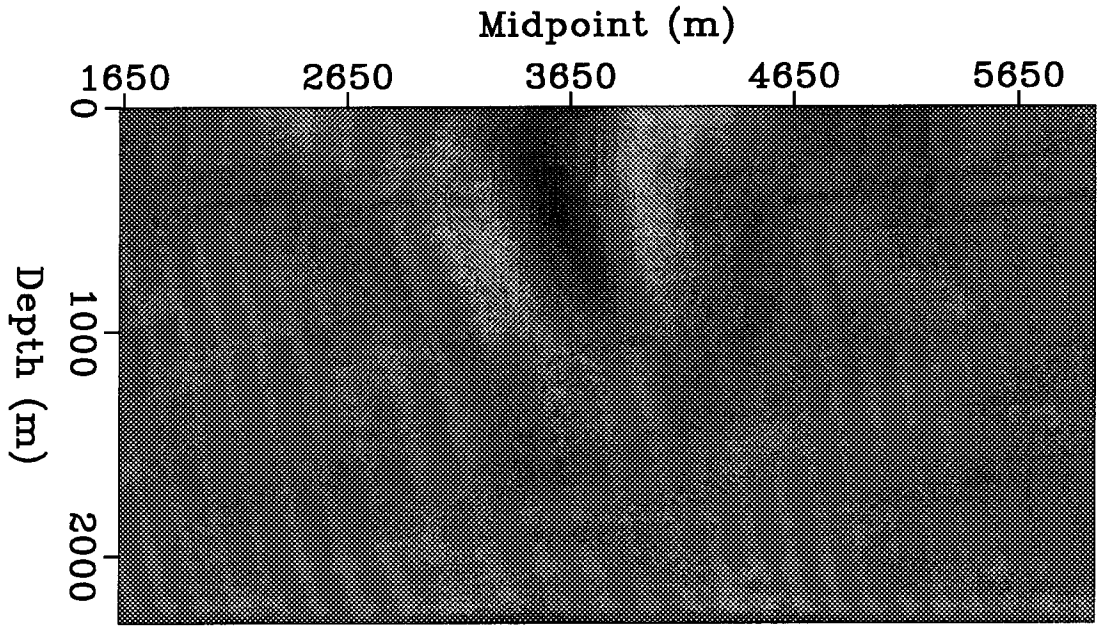

FIG. 3.8. The slowness model after 2 iterations of the Gauss-Newton algorithm presented in section 3.3.2. The estimated anomaly is fairly well localized, although it is still smeared in the direction perpendicular to the reflector and it has two negative side lobes because of the limited ray coverage of the anomaly. The result would improve if the cable length were longer (1500 m) and there were more than one reflector.

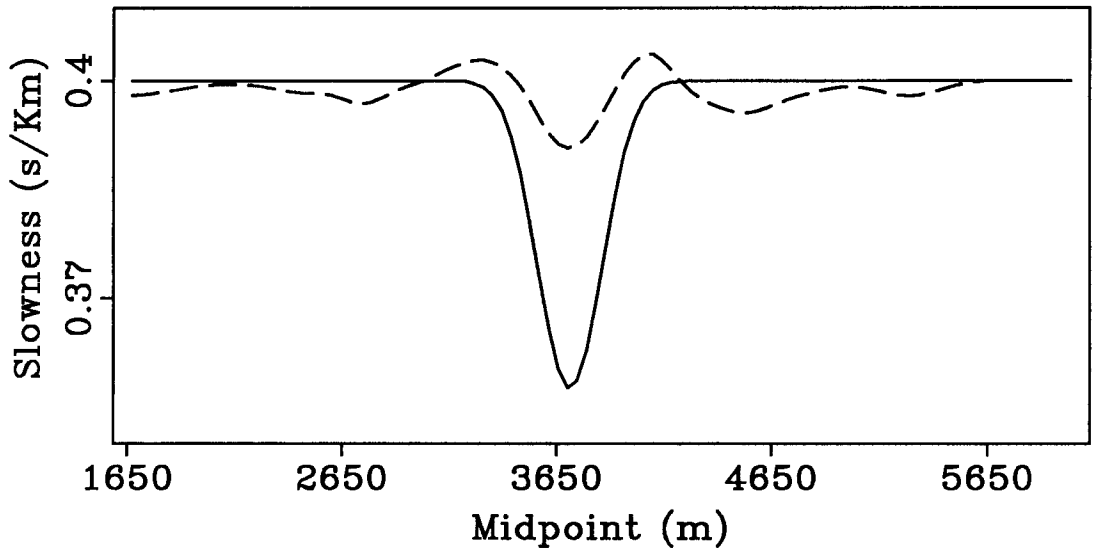

FIG. 3.9. A cross section of the slowness model shown in Figure 3.8 (dashed line) compared with the true model (solid line). The cross sections were taken at the depth of the center of the slowness anomaly; that is at a depth of 700 m.