Chapter 5

Velocity estimation

5.1 INTRODUCTION

So far, I have discussed only the forward problem: how events are migrated after they have been identified in the structural interpretation. The migration images separate subsets of the data, the constant-offset sections. Although each constant-offset section represents a different collection of seismic experiments, they all sample the same subsurface area. Therefore, barring discrepancies due to limited coverage, noise problems, shadow zones, AVO effects, etc., the migrated constant-offset sections, which are images of the subsurface, should be identical to each other. In reality they hardly ever are, because the velocity model needed in the migration of the sections is unknown; the migrated sections are therefore incorrect.

The above observation serves as the basis for the inverse problem: how to find the correct velocity model to use in migrating the events. The inversion is formulated as an iterative optimization process, in which success in the optimization is measured by a function that quantifies discrepancies between migrated events in different constant-offset sections. This objective function is calculated after events have been converted to the zero-offset-time domain, where residual moveout is better determined than in the depth domain. These converted events are equivalent to events in NMO+DMO-corrected data, provided that the DMO operator properly handles lateral velocity variations.

In finding the correct velocity model, the optimization method uses the derivatives of the objective function with respect to the model parameters. These derivatives form a linear operator that describes how perturbations in the velocity model change the positions of migrated reflectors. As in standard traveltime tomography, the calculation of this

linear operator involves the computing of derivatives of traveltime with respect to the model parameters. However, as discussed in Chapter 1, there are some differences between traveltime tomography and the velocity-estimation method presented here. An important difference is that I incorporate in the linear operator reflector movement as a function of velocity. This requires additional gradient calculations, some of which I have already discussed in Chapter 4.

I start this chapter by describing how I parametrize the velocity model. I then discuss the objective function, after which I examine in detail the calculation of its gradient. Next, I present the optimization scheme itself. Finally, I discuss geological constraints and the problem of finding a structural-velocity model.

5.2 MODEL PARAMETRIZATION

It is well-known that velocity determination by moveout or traveltime analysis gives only the low-wavenumber component of the velocity field (Claerbout, 1985, Fig.1.4-3). When a ray travels through several structures in the Earth, its traveltime is an integral measure of the velocities in the structures that the ray encounters. Unraveling the exact velocity of a structure from traveltime observations is therefore possible only when the structure is penetrated by a large number of rays over a wide range of angles. In reflection seismology, where most of the rays travel nearly vertically, such a situation is highly unlikely.

In view of this limitation, I choose to parametrize the velocity model by two-dimensional cubic spline functions that are intrinsically smooth. Also, spline functions can be described by few parameters, so that the inverse problem is well determined. The actual variable determined in the optimization is slowness, the reciprocal of velocity, because the inverse problem is more easily linearized for slowness than it is for velocity. Thus, the 2-D slowness model s is described as

$$s(x,z) = \sum_i \sum_j c_{ij}\, f_i(x)\, g_j(z), \tag{5.1}$$

with f_i and g_j the spline functions at the ith spline cell in the x-direction and the jth spline cell in the z-direction, respectively. c_{ij} are the spline coefficients that form the model parameter vector $\mathbf{m}$. I use a method described by Inoue (1986) in the calculation of spline coeffients for a two-dimensional function.

The initial velocity model is determined from NMO-velocity analysis, geological information, well logs, or any other source of a priori information. Note that some of this information, in particular knowledge coming from well logs, may justify specifying velocity contrasts that cannot be described by smooth spline functions. I will come back to this issue later (section 5.6); for now I assume that no such information is available.

5.2.1 Parametrizing horizons?

Many tomographic methods try to invert for an additional variable apart from velocity: the depth of interfaces in the velocity model (Bishop et al., 1985; Stork and Clayton, 1987). I choose not to parametrize both velocity and depth in the optimization. Instead, I obtain the reflector depths from the migrated data, and use them only in the computation of the backprojection operator. Of course, if the migration velocity model is incorrect, the depths of the reflectors will be wrong, which is why many tomographic inversions parametrize reflector depths in the first place: to correct for errors made in the gradient calculation by assuming a fixed, wrong reflector position. However, by taking the velocity dependency of the reflector positions into account in the gradient calculation, I do not make this type of error, and I do not have to include the horizon locations in the inversion.

Avoiding the use of reflector depth as a model parameter in the optimization has some advantages. First, as I discussed above, specifying horizons and velocity contrasts without any information other than seismic traveltimes may be unwarranted. Second, both velocity and depth of a structure control the position of a reflector after migration. This velocity-depth ambiguity can lead to instabilities and nonconvergence of the solution in the optimization (Stork, 1988). Finally, velocity and depth are two different physical parameters that need to be scaled differently in the optimization. The scaling factors are hard to determine in advance.

5.3 OBJECTIVE FUNCTION

If the migration-velocity model is not correct, the same reflector will appear at different depth positions for different constant-offset sections (Figures 2.5 and 2.7). In principle, one wants to determine differences between the true and migrated depths of the reflector at each offset and surface location, and translate these depth perturbations into velocity perturbations in the inversion method. At each surface location x_r, these depth perturbations

δz_r describe a residual-moveout curve as a function of offset:

$$\delta z_r(x_r, h; \mathbf{m}) \;=\; z_{\text{true}}(x_r) \;-\; z_r(x_r, h; \mathbf{m}), \tag{5.2}$$

where z_{true} is the true reflector depth as a function of surface location, and z_r the migrated reflector depth as a function of surface location, offset, and migration velocity. Unfortunately, the true position of the reflector is not known, and depth perturbations cannot be accurately determined.

5.3.1 Pseudo-depth perturbations after NMO and DMO

A solution to this problem is to model zero-offset events for the migrated reflectors in each constant-offset section. Modeling is done with the operator $\mathcal{M}$ of section 4.3 (equation (4.10)):

$$\mathbf{d_0}(y_r, h;\ \mathbf{m}) \;=\; \mathcal{M}(y_r, h = 0;\ \mathbf{m})(\mathbf{r}(y_r, h;\ \mathbf{m})). \tag{5.3}$$

$\mathbf{d_0}$ are the modeled zero-offset events in the constant-offset sections, and y_r the midpoints of the modeled events (as before, the subscript r denotes that y corresponds to a specific point on the reflector). $\mathbf{d_0}$ depends on offset through the offset dependency of the migrated horizons $\mathbf{r}$. Note that the horizons also depend on velocity: they are found after migration with the velocity model $\mathbf{m}$. I will come back to this in the next section.

In short notation, equation (5.3) is written as

$$\mathbf{d_0}(\mathbf{m}) \;=\; \mathcal{M}_{0,\mathbf{m}}(\mathbf{r}(\mathbf{m})), \tag{5.4}$$

where $\mathcal{M}_{0,\mathbf{m}}$ is the zero-offset modeling operator. Thus, a depth point $\mathbf{r}$ on a reflector in a constant-offset section has two corresponding points in two other domains: a data point $\mathbf{d}$ in the unmigrated time domain, and a *pseudo-depth point* $\mathbf{d_0}$ in the pseudo-depth or zero-offset-time domain. Throughout the rest of this chapter, I will regularly refer to these points, which I assume to be known for each depth point considered in the optimization.

Because the modeled events in the zero-offset section correspond to true zero-offset events (as long as the same velocity is used in migration and modeling), they can be used as a reference for modeled events in other sections. The reflector perturbations are thus better determined in zero-offset time than they are in depth. Another advantage of

converting events to pseudo-depth is that the reference events in the zero-offset section do not move as velocity changes; thus the gradient calculation concentrates on the offset behavior of the events, the part that provides the most velocity information.

Perturbations in zero-offset time τ of the events are now determined at common-midpoint locations y_r:

$$\delta\tau_r(y_r, h; \mathbf{m}) \;=\; \tau_{\text{true}}(y_r) \;-\; \tau_r(y_r, h; \mathbf{m}), \tag{5.5}$$

with $\tau_r(y_r, h; \mathbf{m})$ the time of modeled zero-offset events $\mathbf{d_0}$, and $\tau_{\text{true}}(y_r) = \tau_r(y_r, h = 0)$ the true zero-offset time of an event. Note that I ignore the stepout component of $\mathbf{d_0}$ (recall equation (2.4): $\mathbf{d} = (t, p_y)$); I am only interested in perturbations in pseudo-depth, the time component of $\mathbf{d_0}$.

The result of this cascaded process (migration and zero-offset modeling) is exactly the same as the one that would be obtained from interpreting the events in constant-offset sections after NMO and DMO has been applied to the data (Bolondi et al., 1982; Deregowski, 1986). Of course, like the modeling operator $\mathcal{M}$, the DMO operator has to be able to handle velocity variations in both the depth and lateral directions. Popovici and Biondi (1989) define such a general DMO operator as the same cascade of depth migration and zero-offset modeling.

5.3.2 Minimizing the objective function by least squares

The goal of the optimization is to minimize perturbations in moveout along offset. A common choice for the objective function J to be minimized is the Euclidean or L_2 norm of the perturbations,

$$J(\mathbf{m}) \;=\; \sum_{y_r,h} \Big|\delta\tau_r(y_r, h; \mathbf{m})\Big|^2 = \sum_{y_r,h} \Big|\tau_{\text{true}}(y_r) \;-\; \tau_r(y_r, h; \mathbf{m})\Big|^2. \tag{5.6}$$

The problem of finding a solution that minimizes this L_2-objective function is called the *least-squares* problem, and its solution is named the least-squares solution. Although least-squares methods are widely applied, some different approaches to the inverse problem have been proposed. Claerbout and Muir (1973) advocate the use of the L_1 norm, which leads to methods of linear programming. Linear programming methods are, however, not efficient enough for application in large-scale inversions. From here on, I therefore limit

myself to the L_2 norm.

A first step towards finding the least-squares solution is to linearize the modeled events with respect to the model parameters. Assuming small model perturbations $\delta\mathbf{m}$, the objective function can be evaluated for a perturbed model $\mathbf{m} + \delta\mathbf{m}$ if the problem is linearized around the current solution $\mathbf{m}$:

$$\begin{aligned} J(\mathbf{m}+\delta\mathbf{m}) &= \sum_{y_r,h} \left| \tau_{\text{true}}(y_r) - \tau_r(y_r,h;\mathbf{m}+\delta\mathbf{m}) \right|^2 \\ &= \sum_{y_r,h} \left| \tau_{\text{true}}(y_r) - \tau_r(y_r,h;\mathbf{m}) - \frac{\partial \tau_r}{\partial \mathbf{m}}(y_r,h;\mathbf{m})\delta\mathbf{m} \right|^2 \qquad (5.7) \\ &= \left\| \delta\tau - \mathbf{G}\delta\mathbf{m} \right\|^2. \end{aligned}$$

$\delta\tau$ is a vector consisting of the pseudo-depth perturbations $\delta\tau_r(y_r,h;\mathbf{m})$, $r = 1, N$, with N the number of reflector points. $\|\cdot\|$ denotes the vector norm. $\mathbf{G}$ is a $N \times M$ matrix, with M the number of model parameters. The elements of $\mathbf{G}$ are the first-order derivatives of pseudo-depth with respect to the model parameters m_c:

$$G_{rc} = \frac{\partial \tau_r}{\partial m_c}, \qquad r = 1, N; \quad c = 1, M. \qquad (5.8)$$

(A row of the matrix is calculated for each reflector point, hence the index r. To be consistent, the index for each column is denoted by c.) The gradient operator $\mathbf{G}$ is called backprojection operator because it projects perturbations in pseudo-depth back onto the model.

To find the least-squares solution $\delta\mathbf{m}$ that minimizes J, I solve $\partial J/\partial\mathbf{m} = 0$, which yields the normal equations

$$\mathbf{G}^T\mathbf{G}\delta\mathbf{m} = \mathbf{G}^T\delta\tau. \qquad (5.9)$$

Damped least squares

In practice, the above system is often underdetermined, and the least-squares problem usually must be damped to yield a meaningful solution. This damping can be done by

computing the least-squares solution of a modified system of equations

$$\begin{pmatrix} \mathbf{G} \\ \mathbf{D} \end{pmatrix} \delta\mathbf{m} = \begin{pmatrix} \delta\tau \\ 0 \end{pmatrix}, \tag{5.10}$$

where $\mathbf{D}$ is a damping matrix. The least-squares solution of this system minimizes

$$\left\| \delta\tau - \mathbf{G}\delta\mathbf{m} \right\|^2 + \left\| \mathbf{D}\delta\mathbf{m} \right\|^2 \tag{5.11}$$

(see Tarantola, 1984). $\mathbf{D}$ adds a penalty term to the objective function, and can thus be used to constrain the model according to a priori knowledge. $\mathbf{D}$ is often called the model-covariance matrix.

5.4 BACKPROJECTION OPERATOR

As I noted before, pseudo-depth modeling depends on the velocity model through the modeling operator $M_{0,\mathbf{m}}$, and on the position of the reflector point $\mathbf{r}$ (equation (5.4)), which itself is a function of velocity. To find the correct backprojection operator, the gradient calculation must honor the velocity dependence of the depth point:

$$\frac{\partial \mathbf{d_0}}{\partial \mathbf{m}} = \left. \frac{\partial M_0}{\partial \mathbf{m}} \right|_{(\mathbf{r},\mathbf{m})} + \left. \frac{\partial M_0}{\partial \mathbf{r}} \right|_{(\mathbf{r},\mathbf{m})} \cdot \frac{\partial \mathbf{r}}{\partial \mathbf{m}}, \tag{5.12}$$

or, when the equation is written out for pseudo-depth, the time of the modeled zero-offset event,

$$\frac{\partial \tau_r}{\partial \mathbf{m}} = \left. \frac{\partial \tau_r}{\partial \mathbf{m}} \right|_{(\mathbf{r},\mathbf{m})} + \left. \frac{\partial \tau_r}{\partial x} \right|_{(\mathbf{r},\mathbf{m})} \frac{\partial x_r}{\partial \mathbf{m}} + \left. \frac{\partial \tau_r}{\partial z} \right|_{(\mathbf{r},\mathbf{m})} \frac{\partial z_r}{\partial \mathbf{m}}. \tag{5.13}$$

In the above equation I have replaced the modeling operator $M_{0,\mathbf{m}}$ in equation (5.12) by $\tau_r = \tau_r(x_r, z_r; \mathbf{m})$ (equation (4.9), where I again ignore the stepout component of $\mathbf{d_0}$).

Thus, the backprojection operator consists not only of derivatives of pseudo-depth with respect to the model parameters for a fixed reflector position (the first term in the right-hand side of the above equation), but it also has two terms that describe the changes in pseudo-depth due to reflector movement, which in turn is caused by changes in velocity. This is an important difference with traditional tomography, in which reflector positions

normally are assumed to be fixed. The inclusion of reflector movement in the gradient calculations is crucial in reflection tomography; without it, the backprojection operator would be less accurate.

Equation (5.13) gives one row (for one depth point) of the backprojection operator

$$\mathbf{G}_r = \mathbf{A}_r + \mathbf{B}_r\mathbf{M}_r, \tag{5.14}$$

with

$$\mathbf{A}_r = \left.\frac{\partial \tau_r}{\partial \mathbf{m}}\right|_{(\mathbf{r},\mathbf{m})}; \quad \mathbf{B}_r = \begin{pmatrix} \frac{\partial \tau_r}{\partial x} & \frac{\partial \tau_r}{\partial z} \end{pmatrix}; \quad \mathbf{M}_r = \begin{pmatrix} \frac{\partial x_r}{\partial \mathbf{m}} \\ \\ \frac{\partial z_r}{\partial \mathbf{m}} \end{pmatrix}, \tag{5.15}$$

which requires the calculation of three different submatrices $\mathbf{A}_r, \mathbf{B}_r$, and $\mathbf{M}_r$. Matrix $\mathbf{A}_r$ describes how pseudo-depth varies as a function of the model parameters at fixed reflector position. Matrices $\mathbf{B}_r$ and $\mathbf{M}_r$ represent the effects of reflector movement: $\mathbf{M}_r$ is the linear residual-event-migration operator discussed in section 4.7; it relates changes in the model parameters to changes in reflector position. The latter changes are then translated by $\mathbf{B}_r$ into changes in pseudo-depth.

Note that the elements of $\mathbf{G}$ are zero for the zero-offset section: at zero offset, the elements of matrix $\mathbf{A}$ are canceled by the elements of $\mathbf{BM}$. In other words, pseudo-depths in the zero-offset section corresponds to true zero-offset times in the data, and are not affected by changes in the model.

5.4.1 Computing the backprojection operator

The calculation of the full backprojection operator $\mathbf{G}$ requires, in addition to the calculation of matrix $\mathbf{M}$ (which I discussed in section 4.7), the computation of matrices $\mathbf{A}$ and $\mathbf{B}$. However, these matrices consist of the same type of derivatives as the ones needed in the computation of $\mathbf{M}$. The only difference is that the derivatives for $\mathbf{A}$ and $\mathbf{B}$ describe changes in the pseudo-depth point $\mathbf{d_0}$, instead of changes in the data point $\mathbf{d}$. Furthermore, these matrices contain only derivatives of zero-offset time; no stepout derivatives are needed. Thus, a row of the matrix $\mathbf{B}$, consisting of $\partial\tau_r/\partial x$ and $\partial\tau_r/\partial z$, and a row of matrix $\mathbf{A}$, consisting of $\partial\tau_r/\partial\mathbf{m}$, are calculated as discussed in Appendix A, in which source

and geophone position are identical and given by the midpoint position of the modeled pseudo-depth point $\mathbf{d_0}$.

To summarize, for each depth point, the calculation of $\mathbf{G}$ involves the following steps:

1. Calculate derivatives of $\mathbf{d_0}$ with respect to model parameters; this gives a row of the matrix $\mathbf{A}$.
2. Calculate derivatives of $\mathbf{d_0}$ with respect to reflector movement; this gives a row of the matrix $\mathbf{B}$.
3. Repeat step 1 and 2, but now for $\mathbf{d}$, and construct $\mathbf{M}_r$, a submatrix of $\mathbf{M}$. This involves taking the inverse of a 2×2 matrix (equation (4.21)).
4. Compute a row of $\mathbf{G}$ with: $\mathbf{G}_r = \mathbf{A}_r + \mathbf{B}_r \mathbf{M}_r$.

The above procedure is repeated for every depth point on all the reflectors in every constant-offset section; this gives the full matrix $\mathbf{G}$.

5.4.2 Verifying the backprojection operator

To verify the backprojection operator, I use it to predict perturbations in τ_r. I first model constant-offset reflection events for a dipping reflector with a velocity model $\mathbf{m}$, and then I migrate them with an incorrect model $\mathbf{m} + \delta\mathbf{m}$. Next, I calculate the gradient operator $\mathbf{G}$ for these wrongly migrated events, and, by calculating $\mathbf{G}\delta\mathbf{m}$, I predict perturbations in pseudo-depth. I then compare these forward-predicted perturbations to the true ones.

The unperturbed model is a constant-velocity model with a velocity of 2 km/s; the perturbed one has a velocity that is 10% too low (1.8 km/s). Figure 5.1a shows the dipping reflector in migrated constant-offset sections after migration with the incorrect model. As can be seen in the figure, the reflector is imaged incorrectly: it is curved, and its position changes with offset. Figure 5.1b shows the migrated constant-offset reflectors converted to pseudo-depth by zero-offset modeling. The modeled event in the zero-offset section corresponds to the true zero-offset event (time τ_{true}), which serves as a reference for the events in the other constant-offset sections (time τ_r). The resulting pseudo-depth perturbations $\delta\tau = \tau_{\text{true}} - \tau_r$ are displayed for two midpoints in Figure 5.2 (solid lines). I now calculate the backprojection operator for all depth points in the different constant-offset sections, and then apply it to the model perturbations $\delta\mathbf{m}$ to predict perturbations $\delta\tau = \mathbf{G}\delta\mathbf{m}$ (dashed lines in Figure 5.2).

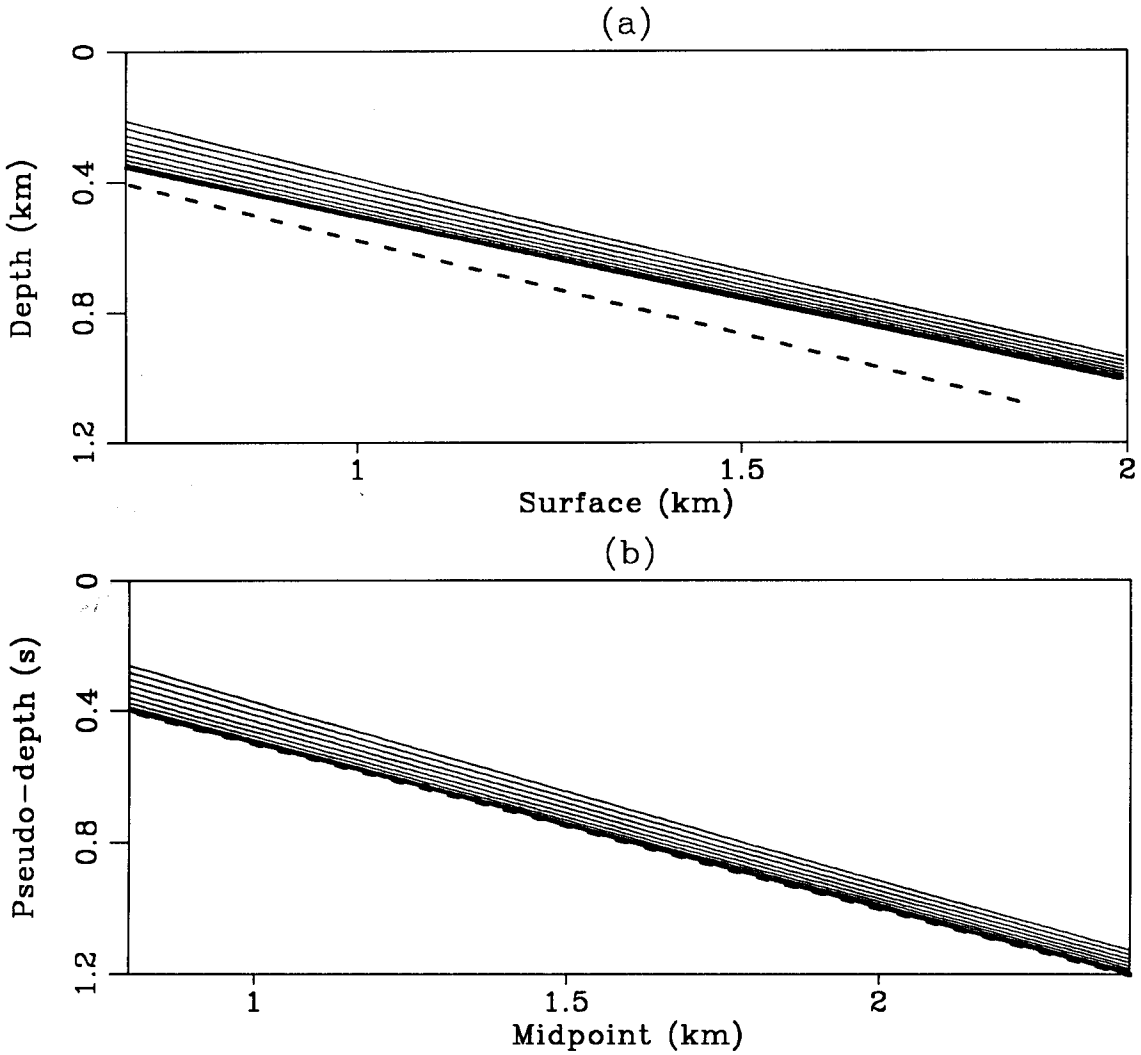

FIG. 5.1. Dipping reflector (true dip 30°) migrated with an incorrect velocity model (velocity 1.8 km/s; the true velocity is 2 km/s). Figure (a) shows the true reflector (dashed line) and the incorrectly migrated reflector in different constant-offset sections, with offsets ranging from 0 to 1.8 km (solid lines). The zero-offset reflector is shown by a fat line. Figure (b) shows the migrated constant-offset sections in pseudo-depth, after zero-offset modeling. The fat line denotes the zero-offset reflector in pseudo-depth, which is identical to the true zero-offset reflection event (dashed line).

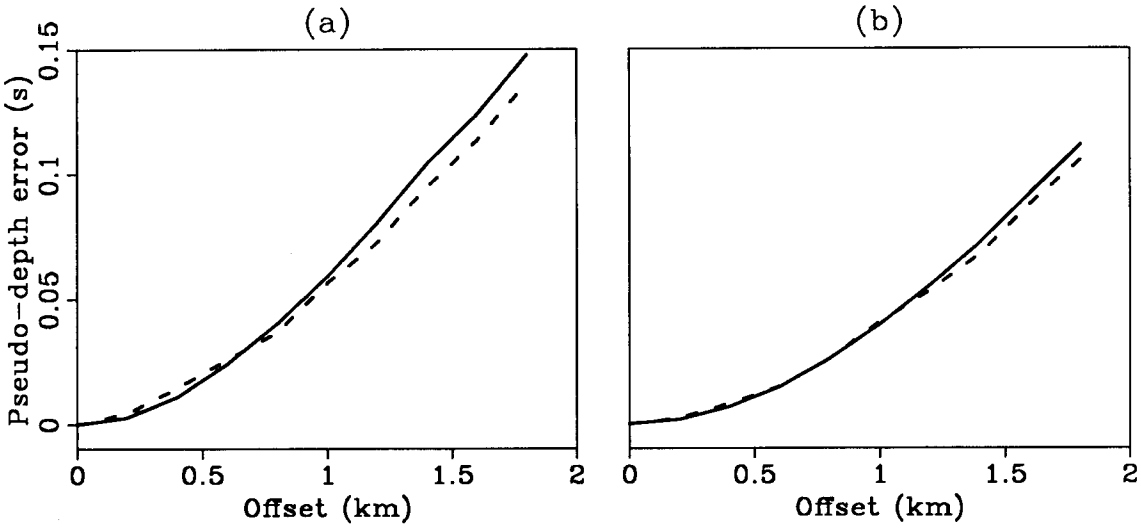

FIG. 5.2. Pseudo-depth perturbations for the dipping reflector of Figure 5.1 at midpoints (a) 1 km and (b) 2 km (solid lines). The dashed lines show the forward-predicted perturbations $\mathbf{G}\delta\mathbf{m}$, where $\mathbf{G}$ is calculated for a 16×16 spline model.

These predicted perturbations closely resemble the true ones, even though the backprojection operator is calculated for mispositioned reflector points. Furthermore, the reflector positions need not be guessed or estimated: they follow naturally from the migration results. These two aspects are significant advantages over traditional traveltime tomography; there the backprojection operator is often incorrectly calculated for wrong reflector positions, which are either guessed or estimated together with velocity.

5.5 OPTIMIZATION SCHEME

The backprojection operator described above is used in an optimization scheme that iteratively updates the velocity model until the pseudo-depth perturbations are minimized. The algorithm consists of several loops and is displayed in Figure 5.3.

The input to the inversion is a set of reflectors in the migrated constant-offset sections, an initial velocity model $\mathbf{m}$ (the same velocity model that was used for migrating the data), and a damping matrix $\mathbf{D}$ (see section 5.6). Before the optimization is started, the unmigrated events $\mathbf{d}$ are reconstructed with equation (4.11) for each depth point $\mathbf{r}$ on the reflectors. These modeled events are used throughout the inversion to residually migrate the reflectors. Furthermore, this modeling gives the source and geophone positions of

Input:

1. Migrated reflectors in constant-offset sections $\mathbf{r}$.
2. Initial migration-velocity model $\mathbf{m}$.
3. Geological constraints: damping matrix $\mathbf{D}$.

For each surface location S {
Calculate and store finite-difference traveltime map.
}

For each depth point $\mathbf{r}$ {
Model data point: $\mathbf{d} = \mathcal{M}_{\mathbf{m}}(\mathbf{r})$.
}

Nonlinear outer loop: *do* {
For each depth point $\mathbf{r}$ {
Model pseudo-depth point: $\mathbf{d_0} = \mathcal{M}_{0,\mathbf{m}}(\mathbf{r})$.
Determine pseudo-depth perturbation $\delta\tau_{\mathbf{r}}$.
Calculate row of gradient operator $\mathbf{G}_{\mathbf{r}}$.
}

Linear inner loop: *do* {
Invert for $\delta\mathbf{m}$: $\delta\mathbf{m} = (\mathbf{G}^T\mathbf{G} + \mathbf{D}^T\mathbf{D})^{-1}\mathbf{G}^T\delta\tau$.
Update model: $\mathbf{m} = \mathbf{m} + \delta\mathbf{m}$.
} *until* converge in $\mathbf{m}$.

For each surface location S {
Calculate and store finite-difference traveltime map.
}

For each depth point $\mathbf{r}$ {
Update reflector position: $\mathbf{r} = \mathcal{M}_{\mathbf{m}}^{-1}(\mathbf{d})$.
}
} *until* converge in $\mathbf{r}$.

Output:

1. Updated reflectors $\mathbf{r}$.
2. Updated velocity model $\mathbf{m}$.

FIG. 5.3. Optimization algorithm.

the data point belonging to each depth point, information that is needed in the gradient calculation. The modeling uses finite-difference traveltime maps calculated at regularly spaced surface locations. These maps are stored and used during the optimization by the various migration and modeling operators.

The algorithm consists of two major loops: a linear inner loop and a nonlinear outer one. In the inner loop, model perturbations are determined by an inversion of the linear gradient operator. The perturbations in the model are found by solving the damped system of equations (5.10) by least squares:

$$\delta\mathbf{m} = \left(\mathbf{G}^T\mathbf{G} + \mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{G}^T\delta\tau. \tag{5.16}$$

The inverse matrix in the above equation is not explicitly calculated; instead, a conjugate-gradient method is used to find the least-squares solution. I use the conjugate-gradient routine LSQR of Paige and Saunders (1982). New traveltime maps are computed for the updated velocity model in the outer loop of the algorithm, after which reflector positions are adjusted with the nonlinear residual-migration operator (equation (4.15)). The updated reflectors are used in the recalculation of the pseudo-depth perturbations $\delta\tau$ and the gradient operator $\mathbf{G}$. The inner and outer loops are run until the solution converges in $\mathbf{m}$ or $\mathbf{r}$, respectively.

5.6 STRUCTURAL VELOCITIES

As I discussed in section 5.2, only the low-wavenumber component of the velocity model can be determined from event moveout. However, a smooth velocity model may limit the interpretation of the migrated data: an interpreter may prefer a structural-velocity model that can be related to geological features in the seismic image. In regions with salt structures, in particular, finding a structural model and locating salt boundaries can be important for oil exploration (Larner, 1987).

Given that moveout analysis can determine only smooth velocity models, information about structural velocities has to come from other sources. If available, well logs probably provide the most important source of information. In most cases well logs allow the determination of reflector depths and velocities of major structures. Furthermore, they reveal the lithology of sediments in the region, knowledge that also proves useful in interpreting seismic data away from the well.

If no well logs are available, the seismic image itself provides a good start in deriving a structural model. Although structural boundaries may be mispositioned, the image is generally focused enough so that major structures can be identified. For example, salt structures are normally easy to recognize on the migrated image, and the velocity of salt formations in certain sedimentary basins is often known. The seismic data may also show different wave types that can further constrain the velocity model. The latter constraint is illustrated in the field-data example of the next chapter.

Finally, general geological knowledge can help shape the structural model. Examples are constraints on fault dips, section balancing, basin reconstruction, etc. Although promising, the quantitative specification of these constraints is still in its infancy. That is, most of the techniques employed in this field determine structural models by trial-and-error: they forward model the deformation of an assumed structural model in geological time, rather than invert this structural model from the current geology. The treatment of these methods is beyond the scope of this thesis; I refer the reader to the work of Suppe (1983; 1985).

5.6.1 Constraining the optimization

The input of these constraints into the optimization is more of an art than a science. Although a priori knowledge about the structural model can theoretically be entered in the damping or model-covariance matrix **D** (Tarantola, 1984), the exact choice of damping parameters is often elusive. In practice, the only damping matrices that are well-defined either constrain individual velocities or their derivatives (Toldi, 1985; Sword, 1987). This limitation is not very satisfactory when there is complex geology, for which one wants to specify constraints on both structure and velocity. My solution to this problem is to combine the optimization with an interactive analysis of the data and the velocity model. In other words, as long as the quality and acquisition of the geophysical data leave the velocity-inversion problem underdetermined, and given that geological constraints cannot be easily quantified, it is better to trust a trained interpreter than a poorly understood damping matrix.

The interactive analysis consists mostly of specifying a valid input model. Although the velocity analysis determines smooth velocity perturbations, the interpreter may specify a structural input model based on an initial inversion with a smooth velocity model. (An example of this analysis is given in the next chapter.)

5.7 SUMMARY

The velocity analysis is formulated as an iterative optimization process, in which the objective is to minimize discrepancies between migrated events in the different constant-offset sections. The optimization algorithm consists of two loops: a linear inner loop and a nonlinear outer one. In the inner loop, model perturbations are determined from the minimization of the objective function by a conjugate-gradient algorithm. In the outer loop, reflector positions are updated with the nonlinear residual-migration operator of section 4.5.

The gradient operator incorporates reflector movement and ray-bending effects through the linear residual-event-migration operator discussed in the previous chapter. The inclusion of these effects is an important advantage over traveltime tomography, which normally assumes fixed reflectors and rays. Furthermore, the gradient computation does not require an elaborate ray-tracing scheme, the traditional method for calculating derivative operators in tomography. These two advantages make the velocity estimation robust and well-suited for structurally-complex areas, where errors in structural velocities and reflector positions may be large.