## Chapter 4

# Residual event migration

## 4.1 INTRODUCTION

In Chapter 2 I discussed the need for structural interpretation of migrated seismic data in areas with complex geology. The interpreted data describe the depth of major reflectors in several constant-offset sections. However, because the migration-velocity model is unknown, these reflector depths are generally incorrect, and may vary from offset to offset. The goal of the velocity-estimation method is to update the velocity model iteratively until reflector depths match for all offsets. Re-migrating and re-interpreting the full data set at each iteration is computationally expensive and time-consuming. I therefore residually migrate only the picked horizons.

Horizon or map-migration methods have been used extensively in the migration of stacked sections (Hubral, 1977). These conventional methods use so-called normal rays to migrate events in the stacked data, which are assumed to be equivalent to reflection events recorded in a zero-offset experiment. For each event point, a normal ray is traced downward from the surface, where the take-off angle of the ray is determined by the stepout of the data point. The normal ray travels through the velocity model until its traveltime is half the arrival time of the data point; the endpoint of the ray gives the position of the migrated data point. A normal ray thus models the ray path of a reflection event in a zero-offset experiment, in which the shot ray (traveling from shot to reflector) is identical to the receiver ray (traveling from reflector to receiver).

For non-zero-offset data, the concept of normal rays is much more complicated: there are many combinations of take-off angles of source and geophone rays that yield the same stepout in the data. Consequently, many rays have to be traced downward through the model, before the shot-geophone ray pair is found that not only shares a depth point, but also has a combined traveltime equal to the arrival time of the data point. In complex velocity models, finding this ray pair can be complicated.

In this chapter I discuss an efficient method for migrating events in constant-offset sections. The method is based on the traveltime calculations described in the previous chapter; because it does not use rays, it avoids the problem described above. The event migration is fully general: no assumptions are made on the velocity model, or on the movement of reflectors. The only inaccuracies come from picking errors made in the interpretation, but because the computations are fast enough that migrations can be run interactively, possible picking mistakes are easily corrected.

As mentioned at the beginning of this section, the velocity-estimation method requires *residual* migration of events. Residual event migration starts with the result of a previous migration: the reflectors in constant-offset depth sections. I formulate residual migration as a two-stage process, in which I first recover reflection events by event modeling, and then migrate them with the new velocity model. The event modeling also uses the traveltime calculations of Chapter 3.

I begin this chapter by reviewing prestack depth migration, the major process I apply to the data before interpretation. Then, I discuss event modeling and migration. Finally, I describe residual event migration, first as a nonlinear operator, applicable to general changes in the velocity model, and then as a linear one, which linearly relates changes in the reflectors to model perturbations. This latter operator plays an important role in the next chapter, in which I discuss the inverse problem of determining model perturbations from discrepancies in the horizons. I illustrate the several operators with a field data example.

## 4.2 PRESTACK DEPTH MIGRATION

In Chapter 2 I introduced migration as a process that corrects for the wave propagation that often distorts the recorded seismic data. In the last decades, numerous migration methods have been developed (see Gardner, 1985, for a collection of papers). The different methods can roughly be divided into finite-difference, frequency-domain, and integral (Kirchhoff) techniques. Each technique has certain advantages and disadvantages with respect to computational speed and capability to handle irregular geometries or strong variations in velocity (Claerbout, 1985, p.38; Yilmaz, 1987, p.246).

I use a Kirchhoff depth-migration method. Depth-migration methods (Schultz and Sherwood, 1980) are well-suited for migrating data recorded in areas with complex structure and strong lateral velocity variation, as opposed to time-migration methods, which can be applied only in areas with mild lateral velocity variations (Hatton et al., 1981). My reasons for choosing a Kirchhoff method are twofold. First, Kirchhoff methods have the advantage that they allow partial imaging of the subsurface, which is useful if only windows in the data need to be migrated (see section 6.2). Second, the Green's functions used in Kirchhoff migration can efficiently be calculated with the finite-difference traveltime method.

Kirchhoff migration can readily be applied to different subsets, or gathers, of seismic data. Migration-velocity analysis methods often use shot-profile migration (Al-Yahya, 1987; Cox et al., 1988), mainly because shot-record migration can easily be applied to data recorded with irregular shot geometry. Nonetheless, I prefer migrating constant-offset sections because these sections are easy to interpret, as I discussed in Chapter 2, and because events recorded at different offsets remain separated after migration, simplifying the velocity analysis (shot-profile migration collapses events along the offset direction). (The latter point is also important if one wants to study amplitude-versus-offset (AVO) effects in the migrated data.)

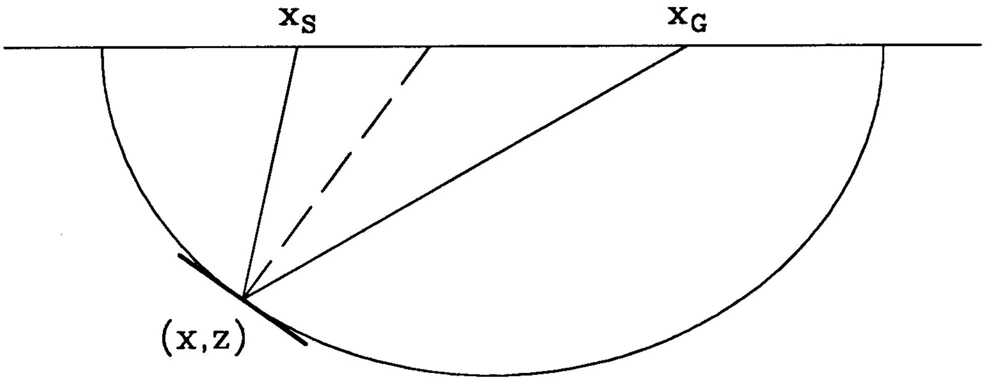

FIG. 4.1. Migration maps a data point recorded at time $t$ with shot at $x_S$ and geophone at $x_G$ onto a semi-ellipse in depth. The shape of the ellipse is determined by the migration-velocity model.

Assuming constant velocity, prestack depth migration maps a time sample recorded at time $t$ and shot-receiver pair $(x_S, x_G)$ onto a semi-ellipse in depth (Figure 4.1), where the foci of the ellipse are determined by shot and geophone position. The depth points $(x,z)$ on the ellipse satisfy

$$t_S(x,z) \ + \ t_G(x,z) \ = \ t(x_S, x_G), \tag{4.1}$$

with $t_S$ and $t_G$ the traveltimes from depth point to source and geophone, respectively. For nonconstant velocity, the ellipse becomes distorted; equation (4.1) then describes a general curve in the subsurface.

## 4.3 EVENT MODELING

Fermat's principle states that a ray will travel along a path for which the traveltime is stationary with respect to minor variations of the raypath. In most situations, this means that the raypath between two points is the least-time path, that is, of all the possible raypaths between the points, the ray will take the one requiring the least traveltime. Fermat's principle provides a straightforward way of modeling events with the traveltime calculations of Chapter 3. Suppose traveltime maps $t_S(x,z)$ are calculated for regularly spaced surface sources $S$, to yield a set of maps

$$t_S(x,z), \qquad S = i\Delta S \ \text{ for } i = i_{\min}, i_{\max}, \tag{4.2}$$

with $\Delta S$ the source spacing (Figure 4.2a).

Now consider a reflector whose position $(x,z)$ is determined by the parametric representation

$$(x(r), z(r)), \qquad r \in [0,1]. \tag{4.3}$$

The parameter $r$ distinguishes the different points on the reflector, and its range is conveniently defined as $[0,1]$. A schematic example of this reflector representation is shown in the right plot of Figure 4.2a. The traveltimes $t_{SG}$ of all possible rays traveling from source $S$ to geophone $G$ and reflecting at the reflector are simply found by the summing of shot and geophone traveltimes along the reflector:

$$t_{SG}(r) \ = \ t_S\big(x(r), z(r)\big) \ + t_G\big(x(r), z(r)\big), \qquad r \in [0,1]. \tag{4.4}$$

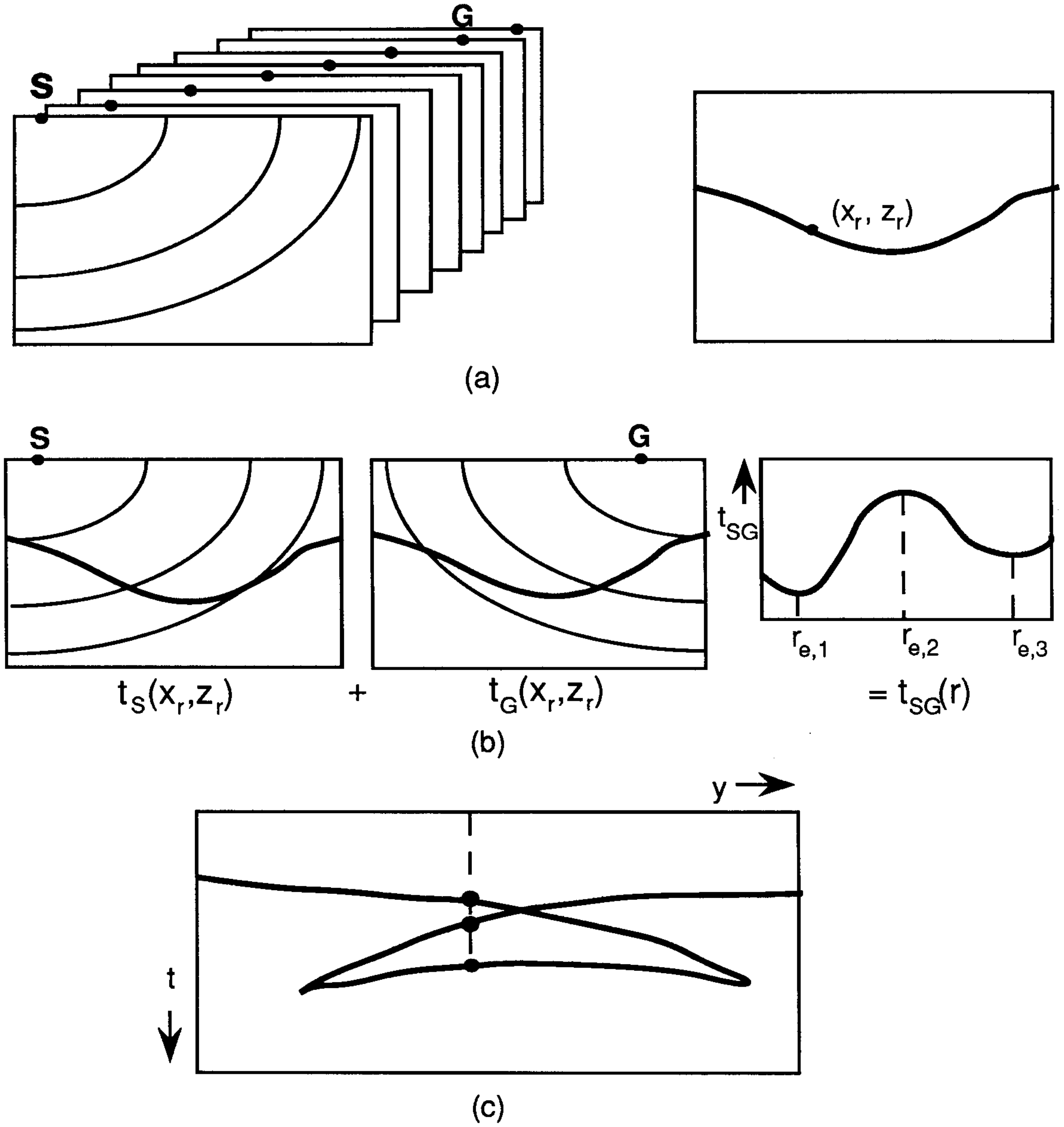

FIG. 4.2. Event modeling. (a) Input: a set of traveltime maps for a range of surface sources (dots in left plot), and a horizon, the coordinates of which are described by $(x(r), z(r))$ (right plot). (b) Fermat's principle: the left two plots show the traveltime maps for shot $S$ and geophone $G$, respectively, with the reflector superimposed on them. Summed traveltimes along the reflector form a curve $t_{SG}(r)$ (right plot). The stationary points $r_{e,1}$,$r_{e,2}$, and $r_{e,3}$ are the points where Fermat's principle is satisfied, and the traveltime at these points are the time of reflections recorded at $S$ and $G$. (c) Output of event modeling: a constant-offset section. The dashed line is drawn at the midpoint of $S$ and $G$, and the times of the three dots correspond to the traveltimes found in (b).

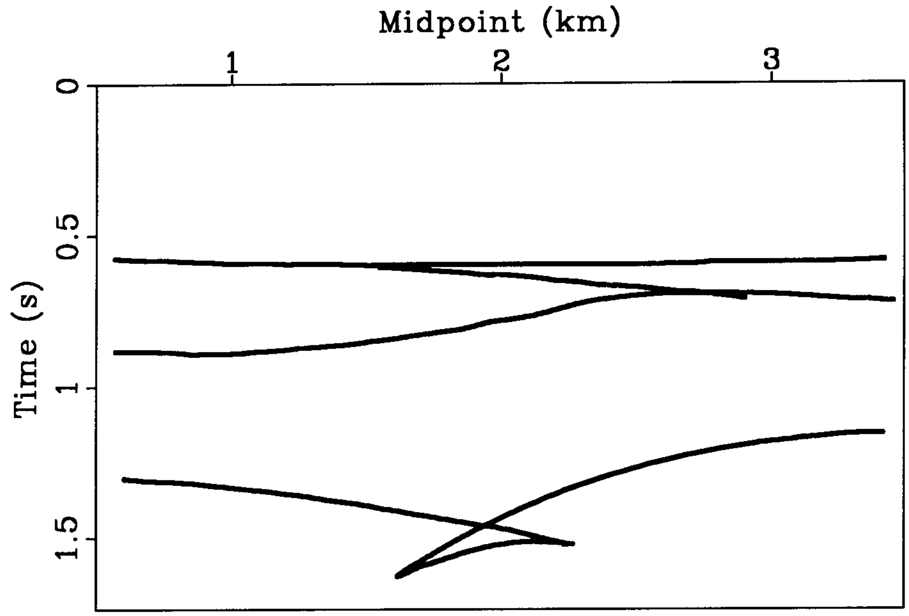

FIG. 4.3. Reflection events in a constant-offset section computed for reflectors in the model of Figure 3.1. The offset is 1 km.

Applying Fermat's principle, the time of the reflection event at source-receiver pair $SG$ is given by the stationary points

$$t_{SG}(r_e), \qquad \text{with } t'_{SG}(r_e) \;=\; 0. \tag{4.5}$$

$r_e$ determines the reflector point(s) $R$ where Snell's law is satisfied. Snell's law can be derived from Fermat's principle (Aki and Richards, 1980, Ch.4), and says that at the reflector the angle of the incident ray equals the angle of the reflected ray, where angles are measured with respect to the normal at the reflector. The coordinates of $R$ are

$$(x_r, z_r) \;=\; (x(r_e), z(r_e)). \tag{4.6}$$

The above procedure is repeated for several shot-geophone pairs to model the events in a complete survey, as is demonstrated in Figure 4.2c. Figure 4.3 shows an example of a thus modeled constant-offset section for the reflectors and velocity model of Figure 3.1.

As discussed in Chapter 2 (section 2.2.3), a reflection event in a constant-offset section is defined not only by its arrival time $t$, but also by its stepout $p_y$. Stepout $p_y$ of the event point is modeled as the sum of source and geophone stepout, $p_S$ and $p_G$,

$$p_y(x_S, x_G) \; = \; p_S(x_r, z_r) \; + \; p_G(x_r, z_r) \tag{4.7}$$

(Claerbout, 1985). $p_S$ and $p_G$ are easily calculated from the traveltime maps with a finite-difference approximation. For example,

$$p_S(x, z) \; \simeq \; \frac{t_{S+1}(x, z) - t_{S-1}(x, z)}{2\Delta S}. \tag{4.8}$$

$p_S$ is the stepout in traveltime for a ray traveling from a depth point $(x, z)$ to the source at $S$. $S - 1$ and $S + 1$ denote the shots to the left and right of $S$, respectively.

### 4.3.1 Triplications

Reflection events are often multi-valued for curved reflectors (triplications). The algorithm described above models triplications by finding more than one stationary point in equation (4.5), as is demonstrated in Figure 4.2b. The modeled constant-offset section in Figure 4.3 shows a triplication for the deep reflector.

Many applications—including the event-migration method described in the next section—require that consecutive data points in the reflection events correspond to neighboring depth points on the reflector. The above algorithm does not meet this requirement; it models event points as a function of recording geometry. The two ordering schemes can be quite different when events triplicate: rather than modeling the several branches of the reflection event in sequence, the algorithm finds points on separate branches for each particular shot-geophone pair. However, because depth points are known for each event point (equation (4.6)), one can easily reorder event points. I simply create an additional array with the same dimension as the one containing the reflectors points, and each time I find an event point, I store its traveltime in the array at the index of the corresponding depth point. "Gaps" in the array can be filled by interpolation.

### 4.3.2 Operator notation

For a particular shot-receiver pair, the event-modeling algorithm models *event* or *data points* $\mathbf{d} = (t, p_y)$ from *depth* or *reflector points*, $\mathbf{r} = (x_r, z_r)$, and can therefore be written as

$$(t,p_y) \;=\; \Big(\, t(x_r,z_r;\ \mathbf{m}),\ p_y(x_r,z_r;\ \mathbf{m}) \,\Big). \tag{4.9}$$

The modeled events depend of course on $\mathbf{m}$, the parameters that describe the velocity model. These parameters can be velocity values at discrete grid points in the model, or spline coefficients modeling a smooth velocity function. I discuss the model parametrization in the next chapter (section 5.2).

For easy reference, I denote the modeling algorithm by an operator $\mathcal{M}$, and write equation (4.9) as

$$\mathbf{d} = \mathcal{M}(x_S,x_G;\ \mathbf{m})(\mathbf{r}). \tag{4.10}$$

Apart from its aforementioned dependence on the velocity model, $\mathcal{M}$ is a function of the source and geophone positions, $x_S$ and $x_G$. Note that the depth point also depends on these positions: there is a depth point for every data point at a particular source-receiver pair. In general, the modeling of a data point $\mathbf{d}$ from a depth point $\mathbf{r}$ is unique: strong lateral velocity variation is needed for rays, originating from the same shot location and reflecting at two different reflector points, to arrive at the same geophone location with identical traveltime and stepout. From here on, I assume that $\mathcal{M}$ is a one-to-one operator. Even if it is not, in practice $\mathcal{M}$ can always be made unique by restricting the algorithm to find only one depth point. (Remember that $\mathcal{M}$ is not an analytical operator; it is just a convenient notation for a modeling algorithm.)

Depending on the choice of shot and geophone positions, equation (4.10) can model both shot gathers and constant-offset sections. Therefore (and for conciseness), I will drop the dependence on recording geometry in subsequent equations, and write the above equation as

$$\mathbf{d} = \mathcal{M}_{\mathbf{m}}(\mathbf{r}). \tag{4.11}$$

Here $\mathcal{M}_{\mathbf{m}}$ is a short notation for $\mathcal{M}(\mathbf{m})$.

## 4.4 EVENT MIGRATION

The inverse operator of event modeling is event migration, which maps a data point to a depth point. Similar to modeling, event migration involves the evaluation of arrival time and stepout of events from the traveltime maps. However, the computations are slightly more complicated than they are in modeling, because reflector positions are not known. Nonetheless, I show in this section that event migration can be done for approximately the same cost as modeling.

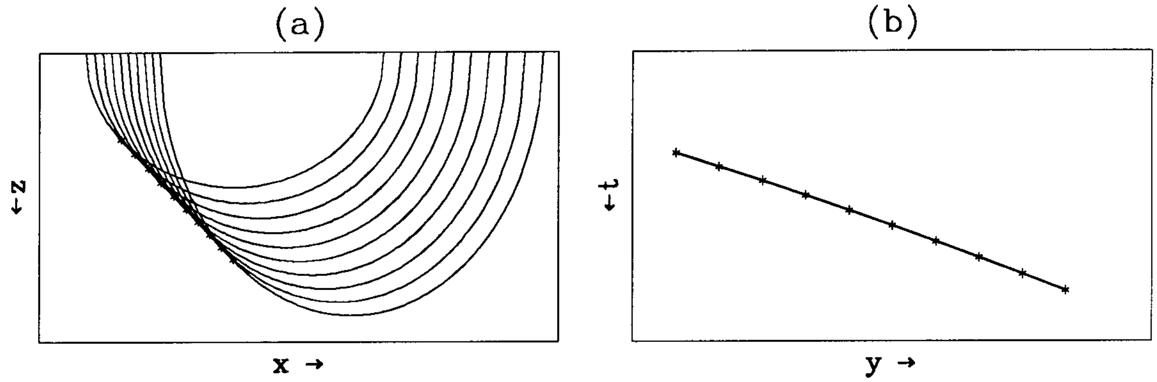

FIG. 4.4. Migration ellipses and reflection event for a dipping reflector. Figure a shows migration ellipses for data points on a constant-offset event (Figure b) reflecting off of a dipping reflector. Data points and corresponding depth points are denoted by asterisks. At each depth point, the dipping reflector segment (the fat line in Figure a) is tangent to the corresponding migration ellipse.

As displayed in Figure 4.1, migration maps a time sample recorded at a given shot-receiver pair onto an ellipse in the subsurface. In event migration, the extra stepout information constrains the mapping of an event sample to a point instead of an ellipse, as is illustrated in Figure 4.4. The figure shows migration ellipses for several data points in an unmigrated constant-offset section, where the data points are part of a reflection off of a dipping layer. The superposition of the migration ellipses shows constructive interference at a linear segment in the subsurface that coincides with the reflector. Note that if the wrong velocity were used in the migration, the reflector segment would have been wrongly positioned and curved.

Now consider the central data point in the reflection event, and assume that its stepout is known and equal to $p_y$. Event migration images this data point at a depth point in the middle of the reflector segment. Of all the possible depth points on the migration ellipse, this depth point is the only point for which the sum of stepouts at the source and geophone equals the observed stepout. In other words, given a data point with arrival time $t$ and stepout $p_y$, event migration selects one depth point $(x_r, z_r)$ on the ellipse where, in addition to equation 4.1, the following condition is met:

$$p_S(x_r, z_r) \ + \ p_G(x_r, z_r) \ = \ p_y(x_S, x_G). \tag{4.12}$$

The tangent to the migration ellipse at the depth point corresponds to the dip of the reflector segment in the migrated section.

The mapping of a data point $\mathbf{d}$ to a depth point $\mathbf{r}$ is again assumed to be unique, and I will write the mapping operator as $\mathcal{M}^{-1}$:

$$\mathbf{r} = \mathcal{M}^{-1}_{\mathbf{m}}(\mathbf{d}). \tag{4.13}$$

Just like $\mathcal{M}$, $\mathcal{M}^{-1}$ depends on $\mathbf{m}$, the velocity model, and on the source and geophone positions of the data point.

### 4.4.1 Algorithm

Equations (4.1) and (4.12) serve as the basis for the event-migration algorithm. As before, traveltime maps $t_S(x, z)$ are available for a range of source positions $S$ (equation (4.2)), and stepout is calculated with equation (4.8). Then, given a data point $\mathbf{d}$ at source-receiver pair $S - G$, a depth point must be found that meets criteria (4.1) and (4.12). Because the traveltime functions are discrete, these criteria are hardly ever met exactly, so instead I use

$$\begin{aligned} |t_S(x, z) \ + \ t_G(x, z) \ - \ t(x_S, x_G)| \quad &\leq \ \epsilon_t; \\ |p_S(x, z) \ + \ p_G(x, z) \ - \ p_y(x_S, x_G)| \quad &\leq \ \epsilon_p, \end{aligned} \tag{4.14}$$

where $\epsilon_t$ and $\epsilon_p$ are some appropriate small numbers. The choice of these parameters determines the resolution of the migration (see below).

In principle, one could search the whole subsurface model (all $(x, z)$-pairs) for the desired depth point, but this would be prohibitively expensive, and, as it turns out, unnecessary. Because a (continuous) reflection event is caused by a continuous reflector, the depth points of neighboring data points on an event must lie in each other's vicinity. Therefore only depth points around a previously found depth point have to be scanned in the verification of conditions (4.14). Only if this limited scan fails is a more elaborate search necessary, as is also the case for the first data point on the event.

The constants $\epsilon_t$ and $\epsilon_p$ are normally set to the errors expected to be made in picking traveltime and stepout, respectively. When no depth point is initially found, one can increase the values of $\epsilon_t$ and $\epsilon_p$, and thereby lower the resolution of the migration. Alternatively, the sampling interval in the traveltime functions can be decreased. The result is that more depth points will be scanned, and thus the chance of finding the desired depth point will increase. Of course, the latter approach increases the computational expense of the algorithm, and in general some trade-off must be made between cost and accuracy. Overall, the cost of migration is of same order as that of modeling: the computations are similar, and are done for approximately the same number of depth points (for modeling, equal to the number of points on the reflector; for migration, equal to the number of points around the previously found depth point).

## 4.5 RESIDUAL EVENT MIGRATION

After an initial event migration, an additional migration with a different velocity model is often necessary. For example, in the velocity-estimation method, the velocity model is updated at every iteration, and each time new reflector positions have to be determined for the updated velocity model. Also, data is interpreted after migration, so the input to the velocity analysis is a set of migrated reflectors as a function of depth, rather than events in time. Although reflection events are therefore not directly available, they can easily be modeled for the migrated reflectors, and then migrated as described above with the new migration model.

Residual event migration is thus simply defined as

$$\mathbf{r}' \;=\; \mathcal{M}^{-1}_{\mathbf{m}'}(\mathcal{M}_{\mathbf{m}}(\mathbf{r})), \tag{4.15}$$

with $\mathbf{r}'$ the residually migrated depth point, $\mathbf{m}'$ the new velocity model, and $\mathbf{m}$ and $\mathbf{r}$ the orignal velocity model and depth point, respectively.

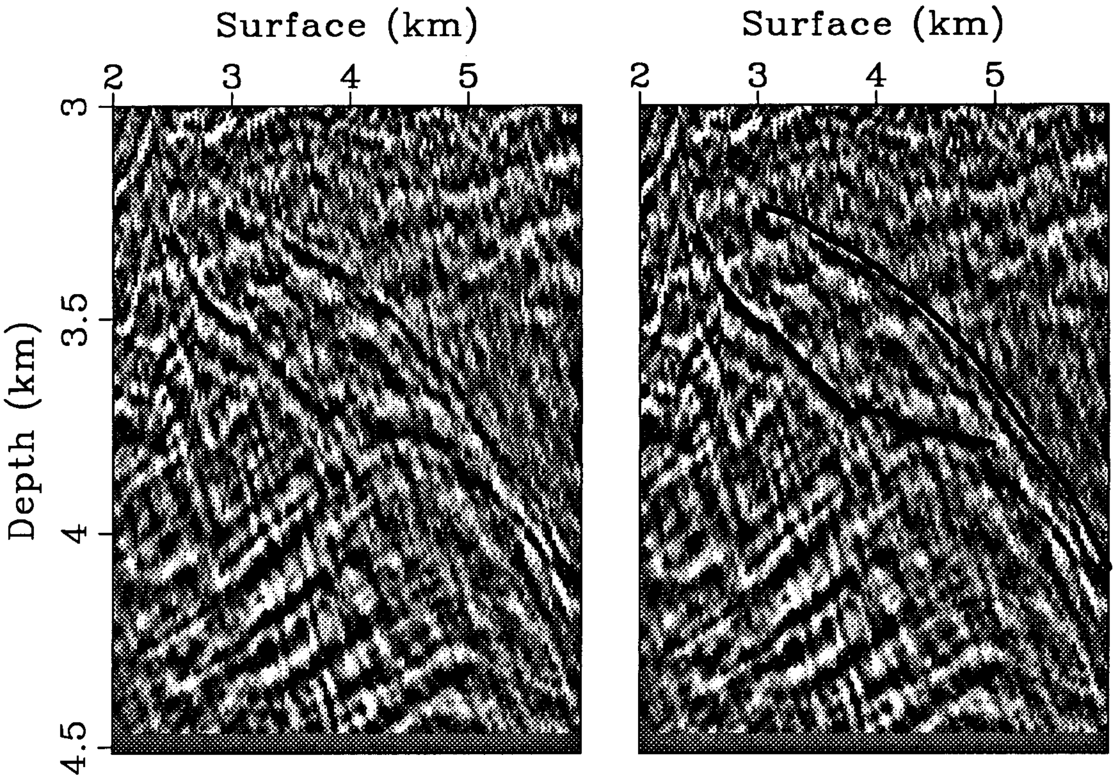

FIG. 4.5. Detail of constant-offset section (offset 1.75 km), migrated with a laterally invariant velocity model. The right figure plots the picked steeply dipping reflectors on top of the same image.

## 4.6 FIELD DATA EXAMPLE

To show an example of event modeling and migration, I residually migrate the two steeply dipping reflectors that I discussed in Chapter 2. Figure 4.5 shows the picked reflectors in a migrated constant-offset section at an offset of 1.75 km. The section is migrated with the laterally invariant velocity model shown in Figure 6.1. The unmigrated reflection events are first reconstructed by event modeling (Figure 4.6), and then migrated with a new migration-velocity model. The velocity model incorporates the high velocity of the salt body, and is discussed in more detail in Chapter 6 (see Figure 6.1). Figure 4.7 plots both the residually migrated reflectors and the fully remigrated data.

The modeled events match events in the unmigrated data, although it is debatable whether an interpretation based solely on the unmigrated data would find these events. The residually migrated reflectors, on the other hand, closely follow the reflectors imaged by a full remigration of the section with the new velocity model.

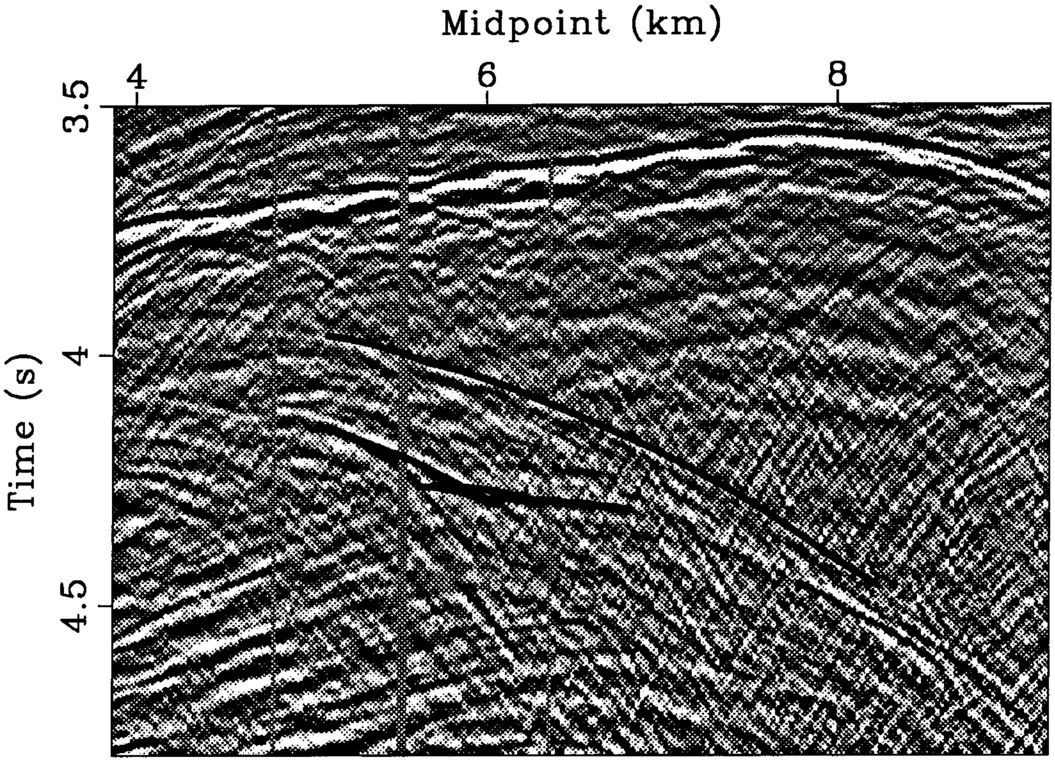

FIG. 4.6. Events modeled for the reflectors in Figure 4.5, displayed on top of the unmigrated data. Figure 2.6 shows the same data without the modeled events.

## 4.7 RESIDUAL EVENT MIGRATION AS A LINEAR OPERATOR

The residual-event-migration operator $\mathcal{M}_{m'}^{-1}\mathcal{M}_{m}$ of section 4.5 takes migrated reflectors and calculates their new positions for an updated velocity model. The operator is nonlinear with respect to the model parameters, and capable of handling large changes in velocity. In this section I describe a linear residual-event-migration operator, which I use in the velocity optimization described in the next chapter. Because the operator is linear, its application is limited to moderate perturbations in velocity. Nevertheless, it can be calculated for any background-velocity model, and is not restricted by any assumptions about the nature of reflector movement. In particular, the operator is not based on an analytical description of reflector movement in a constant-velocity medium (Rothman et al., 1985; Fowler, 1988; Etgen, 1990).

Both the nonlinear and linear operator are derived from the same principle: ***the time and stepout of an event point have to be conserved in residual migration.*** Suppose a depth

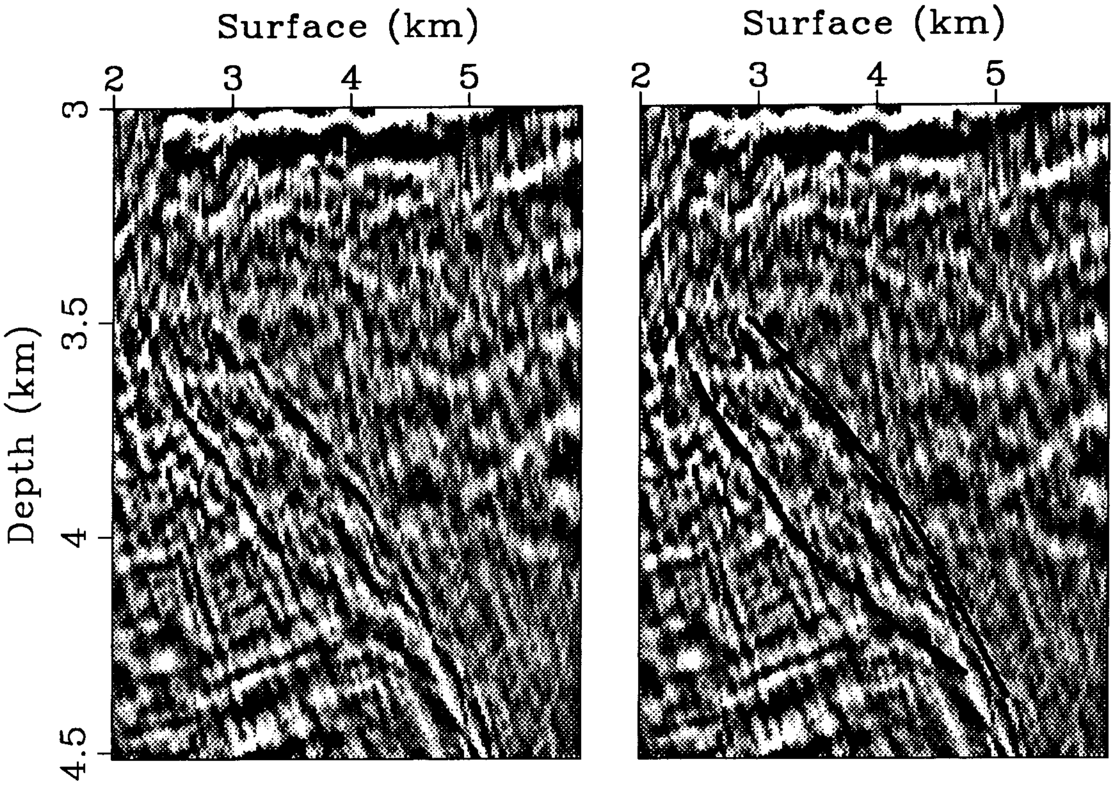

FIG. 4.7. Same section as in Figure 4.5, but remigrated with the velocity model displayed in Figure 6.1. The right plot shows the residually migrated reflectors.

point $\mathbf{r}$ on the migrated reflector is found by imaging the data point $\mathbf{d}$ with velocity model $\mathbf{m}$. The same data point will be imaged at a different position $\mathbf{r}+\delta\mathbf{r}$ if the velocity model is perturbed by $\delta\mathbf{m}$. The equality can be written with the help of the modeling operator $\mathcal{M}$, as

$$\mathbf{d} \;=\; \mathcal{M}_{\mathbf{m}}(\mathbf{r}) \;=\; \mathcal{M}_{\mathbf{m}+\delta\mathbf{m}}(\mathbf{r}+\delta\mathbf{r}). \tag{4.16}$$

In the previous section, the new reflector position is simply found by applying the migration operator $\mathcal{M}^{-1}_{\mathbf{m}+\delta\mathbf{m}}$, the inverse of the modeling operator, to both sides of the equation (resulting in equation (4.15)).

Here, I first linearize the right-hand side of the equality:

$$\mathcal{M}_{\mathbf{m}+\delta\mathbf{m}}(\mathbf{r}+\delta\mathbf{r}) \;=\; \mathcal{M}_{\mathbf{m}}(\mathbf{r}) \;+\; \left.\frac{\partial\mathcal{M}}{\partial\mathbf{m}}\right|_{(\mathbf{r},\mathbf{m})} \delta\mathbf{m} \;+\; \left.\frac{\partial\mathcal{M}}{\partial\mathbf{r}}\right|_{(\mathbf{r},\mathbf{m})} \delta\mathbf{r}. \tag{4.17}$$

Then, equation (4.16) becomes

$$0 = \left.\frac{\partial \mathcal{M}}{\partial \mathbf{m}}\right|_{(\mathbf{r},\mathbf{m})} \delta\mathbf{m} + \left.\frac{\partial \mathcal{M}}{\partial \mathbf{r}}\right|_{(\mathbf{r},\mathbf{m})} \delta\mathbf{r}, \tag{4.18}$$

or, inserting the expressions for depth and data point, and substituting for the modeling operation $(t, p_y) = (t(x_r, z_r; \mathbf{m}), p_y(x_r, z_r; \mathbf{m}))$ (equation (4.9)),

$$0 = \begin{pmatrix} \dfrac{\partial t}{\partial \mathbf{m}} \\[2ex] \dfrac{\partial p_y}{\partial \mathbf{m}} \end{pmatrix} \delta\mathbf{m} + \begin{pmatrix} \dfrac{\partial t}{\partial x} & \dfrac{\partial t}{\partial z} \\[2ex] \dfrac{\partial p_y}{\partial x} & \dfrac{\partial p_y}{\partial z} \end{pmatrix} \begin{pmatrix} \delta x_r \\ \delta z_r \end{pmatrix} \tag{4.19}$$

The perturbations in the depth point can now be linearly related to the model perturbations: the equation is multiplied with the inverse of the 2 × 2 matrix in the second term, so that

$$\begin{pmatrix} \delta x_r \\ \delta z_r \end{pmatrix} = - \begin{pmatrix} \dfrac{\partial t}{\partial x} & \dfrac{\partial t}{\partial z} \\[2ex] \dfrac{\partial p_y}{\partial x} & \dfrac{\partial p_y}{\partial z} \end{pmatrix}^{-1} \begin{pmatrix} \dfrac{\partial t}{\partial \mathbf{m}} \\[2ex] \dfrac{\partial p_y}{\partial \mathbf{m}} \end{pmatrix} \delta\mathbf{m} = \mathbf{M}_r \delta\mathbf{m}. \tag{4.20}$$

The linear residual-migration operator $\mathbf{M}_r$ is thus defined as:

$$\mathbf{M}_r = \frac{\partial \mathbf{r}}{\partial \mathbf{m}} = \begin{pmatrix} \dfrac{\partial x_r}{\partial \mathbf{m}} \\[2ex] \dfrac{\partial z_r}{\partial \mathbf{m}} \end{pmatrix} = - \begin{pmatrix} \dfrac{\partial t}{\partial x} & \dfrac{\partial t}{\partial z} \\[2ex] \dfrac{\partial p_y}{\partial x} & \dfrac{\partial p_y}{\partial z} \end{pmatrix}^{-1} \begin{pmatrix} \dfrac{\partial t}{\partial \mathbf{m}} \\[2ex] \dfrac{\partial p_y}{\partial \mathbf{m}} \end{pmatrix}. \tag{4.21}$$

$\mathbf{M}_r$ plays an important role in the gradient calculation discussed in the next chapter.

### 4.7.1 Calculating the linear residual-migration operator

I describe the calculation of the two types of derivatives in the right-hand side of equation (4.21)—that is, event derivatives with respect to changes in the reflector position and derivatives with respect to model parameters—in Appendix A. The computations are straightforward with the use of the traveltime maps of the previous chapter. I calculate the inverse of the $2 \times 2$ matrix analytically. (In implementing the residual-migration operator, I experienced no problems in determining the inverse. I found one degenerate case: the matrix is singular for a flat reflector in a constant-velocity medium (for which $p_y = 0$). However, $\mathbf{M}_r$ can be determined analytically for constant-velocity media.)

### 4.7.2 Example

I illustrate the linear residual-migration operator $\mathbf{M}_r$ with a simple example of a dipping reflector in a constant-velocity medium, which turns out to be already quite complicated. In this example, I migrate reflection data first with a migration-velocity model $\mathbf{m}$, the result of which I residually migrate with a perturbed model $\mathbf{m} + \delta\mathbf{m}$. Next, I model reflection data for the residually migrated reflector, and compare these data with the true reflection event. If both are the same, the residual operator is correct: the operator should conserve time and stepout of the data. The reason for choosing a constant-velocity example is that reflection data can be calculated analytically in a constant-velocity medium, so that a fair comparison can be made.

The dashed line in Figure 4.8a shows this analytically calculated constant-offset reflection event, and the dashed line in Figure 4.8b shows the dipping reflector after migration with the constant-velocity model $\mathbf{m}$, which has the correct velocity. The points on the reflector are now residually migrated with a perturbed model $\mathbf{m} + \delta\mathbf{m}$, where the model perturbation $\delta\mathbf{m}$ is given by a two-dimensional spline function (see section 5.2). The spline coefficients that describe this function are all zero, except for one; the result is a "blob" perturbation (shown by the contour lines in Figure 4.8b). The horizontal width of the spline cells is larger than the vertical one; this condition explains the elliptical shape of the blob. The solid line in Figure 4.8b displays the residually migrated reflector; the solid line in Figure 4.8a shows the corresponding constant-offset reflection event, modeled with the modeling operator $\mathcal{M}_{\mathbf{m}+\delta\mathbf{m}}$. As can be seen in these figures, the events are virtually identical, which proves that the residual-migration operator is accurate for this example.

Because the velocity in the perturbed model is lower (up to 25%) than the original velocity of 2 km/s, the reflector is overcorrected and moves upward. However, examining individual reflector points reveals a more complicated behavior than just this upward movement: depending on the position of the reflector point with respect to the blob, the point moves to the left or right after residual migration. This complex movement

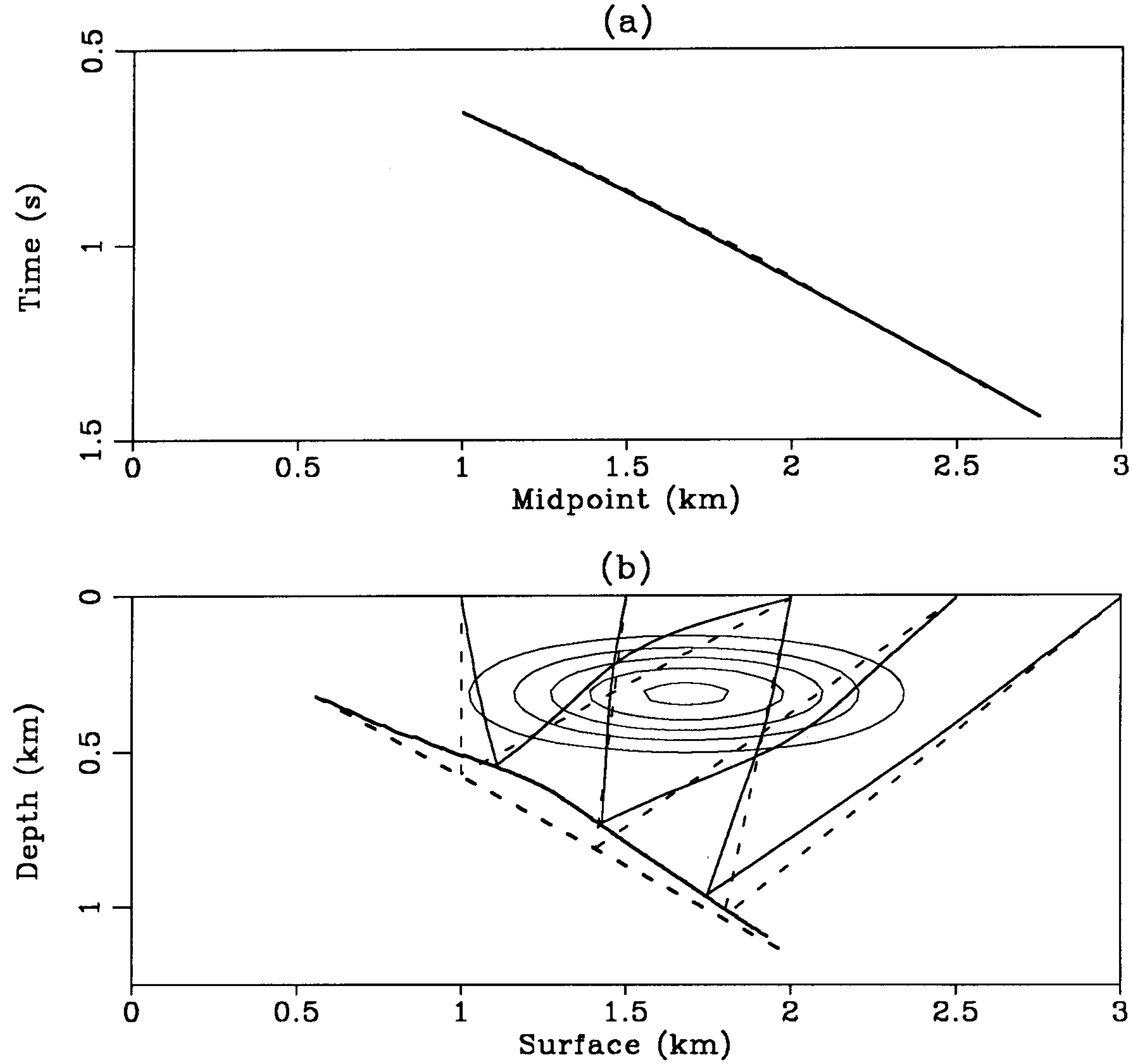

FIG. 4.8. Linear residual migration of a constant-offset reflection event from a dipping reflector (offset 1 km). The original migration-velocity model has a constant velocity of 2 km/s. The linear residual-migration operator migrates the reflector for a perturbed model, which is found by perturbing the constant-velocity model by a 2-D spline function that has one non-zero spline coefficient. The perturbed model is shown by a contour plot (contour interval .1 km/s); the center contour denotes a velocity of 1.5 km/s. The fat line in figure (b) displays the residually migrated reflector. Ray pairs are shown for 3 data points, recorded at midpoints 1.5, 2.0, and 2.5 km. Figure (a) shows reflection events for background and perturbed model (dashed and solid line, respectively).

is needed to preserve traveltime and stepout, and accommodates for ray-bending effects caused by the velocity anomaly. These ray-bending effects are shown in the figure for ray pairs constructed for several data points before (dashed rays) and after residual migration (solid rays). Note that no elaborate ray-tracing scheme is needed in the calculation of the residual-migration operator; the rays in Figure 4.8b are shown merely for illustration, and are found simply by following time gradients in various shot and geophone traveltime maps.

Although this example shows the operator for a local velocity anomaly, the method handles large-scale velocity perturbations just as well. (Section 5.4.2 shows an example in which the operator is calculated for a constant shift in velocity.) Similarly, the operator is not restricted to a constant-velocity background model, but can be calculated for any velocity model.

A question raised by this example is how one can verify that the migration output is correct. Both migration results (with velocity $\mathbf{m}$ and $\mathbf{m} + \delta\mathbf{m}$) are valid, but only one accurately shows the dipping reflector. The only way to know which result is correct is by comparing the position of the migrated reflector at several offsets. This is exactly what I do in the next chapter, where the linear residual-migration operator is used to predict velocity anomalies from discrepancies in migration results at different offsets.

## 4.8 SUMMARY

The operators $\mathcal{M}$ and $\mathcal{M}^{-1}$ are not analytical operators that can be calculated directly from a velocity model; instead, the velocity model is used for computing traveltime maps, which are then manipulated in the event modeling and migration. As a result, neither the modeling nor migration operator makes assumptions on velocity model and reflector shape. The same applies to the nonlinear residual-migration operator, which is just a cascade of $\mathcal{M}$ and $\mathcal{M}^{-1}$. Other residual-migration operators are commonly derived for constant-velocity media. The applicability of the method to all types of velocity models and reflectors is an important advantage in dealing with data recorded in structurally complex areas, where velocities can change rapidly both vertically and horizontally, and structural interfaces can be elaborate. The same consideration is taken into account in the derivation of the linear operator: although the linear operator can handle only moderate changes in velocity, its application is not limited by any assumption about the background velocity model or the shape of the reflectors.