Stanford Exploration Project, Report 67, January 13, 1998, pages 1–103

Stanford Exploration Project, Report 67, January 13, 1998, pages 1–103

Finite-difference calculation of Residual-migration operators

Lin Zhang[1]

keywords: *velocity, algorithm, modeling*

ABSTRACT

Using a kinematic approach, I derive two sets of partial differential equations whose solutions define the kinematics of residual-migration operators. These operators can transform an image migrated with one velocity model to an image migrated with another velocity model. Under the assumption of knowing the partial derivatives of traveltimes with respect to the coordinates of velocity models on a regular grid, these two sets of partial differential equations can then be solved with standard finite-difference techniques. This algorithm can efficiently calculate the residual-migration operators for common shot gathers or constant offset sections migrated with general velocity models. Examples with residual profile migration prove that the accuracy of the algorithm is sufficient for seismic applications.

INTRODUCTION

When perturbations of a velocity model are small, residual migration may be considerably more efficient than full migration. In a classic paper, Rothman et al., (1985) described residual migration of zero-offset sections. Their analysis shows that residual migration is efficient for two reasons: (1) the residual migration operator is sparse; and (2) it can be economically computed. Fowler and Al-Yahya (1986) made an effort to extend these results to prestack migration. They concluded that it is possible to construct a residual prestack migration operator. However, the operator is not equivalent to a full prestack migration operator with some velocity. This conclusion seems to imply that residual prestack migration operators are difficult to compute even when migration velocities are constant.

Using a kinematic approach, Etgen (1989) formulated residual prestack migration of constant offset sections that are initially migrated with a constant velocity. In his method, the residual migration operators are calculated by solving a pair of nonlinear

[1]**email:** not available

equations numerically. Examples with synthetic and field data show that the operators work despite troubles near their triplications. In my previous report (Zhang, 1990), I showed that the kinematic operators of residual profile migration can be efficiently computed with analytical formulas, provided the migration velocities are constant. I also proposed to calculate the operators with a finite-difference scheme when migration velocities are not constant.

In this paper, I present an algorithm for calculating residual migration operators that transform an image migrated with one velocity model to an image migrated with another velocity model. This algorithm is generally applicable to both post-stack and pre-stack migration, and to both common-shot and constant-offset geometries. In principle, it handles general velocity models. Because it uses finite-difference techniques, the algorithm is potentially efficient. To test the accuracy of the algorithm, I calculate residual profile migration operators, and compare them with the analytical solutions. The results show that the errors in position are small fractions of the spatial-sampling intervals.

I will start with an analysis of the computational costs of full prestack migration and residual prestack migration, and explain why calculating operators is an important task in residual migration. Then, I will derive two kinematic relations, and convert them into partial differential equations, which is the key step of the formulation. I will briefly describe the method I use to solve these differential equations. Finally, examples show the accuracy of the algorithm.

RESIDUAL MIGRATION OPERATOR

Why is the efficient calculation of the operators important in residual migration? Let us make a comparison.

Computational costs

I compare the computational costs of full migration and residual migration under the assumption that they are implemented with the Kirchhoff integral in the common shot geometry. I omit those costs that do not depends on the dimensions of data.

A Kirchhoff migration can be done in two steps: (1) calculating operators and (2) applying operators. Suppose a dataset consists of N_s shot gathers. In each shot gather, there are N_r receivers. The image we try to obtain from this dataset has a dimension of $N_x \times N_z$. Then, the computational cost of full migration is

$$\begin{aligned} C_f(\text{total}) &= C_f(\text{calculating operator}) + C_f(\text{applying operator}) \\ &= o(N_l \times N_x \times N_z) + o(N_s \times N_r \times N_x \times N_z) \\ &= o(N_s \times N_r \times N_x \times N_z) \end{aligned}$$

where $N_l < N_s + N_r$ is the number of surface locations sampled by the experiment, and the small letter o stands for *o*rder.

The computational cost of residual migration is

$$\begin{aligned} C_r(\text{total}) &= C_r(\text{calculating operator}) + C_r(\text{applying operator}) \\ &= o(N_s \times N_a \times N_x \times N_z) + o(N_s \times N_a \times N_x \times N_z) \\ &= o(N_s \times N_a \times N_x \times N_z) \end{aligned}$$

where N_a is the number of samples required to represent residual-migration operators. Because the aperture of a residual-migration operator is usually much smaller than that of a full-migration operator, N_a is much smaller than N_r, which makes the application of residual migration efficient. From this comparison, we see that, in full migration, applying operators is the major consumer of the computation-time. In residual migration, however, calculating residual-migration operators is computationally more expensive than applying operators. Therefore, efforts should be made on reducing the cost of calculating residual-migration operators.

Kinematic relations

Suppose a data gather $P(t, x_r)$ is recorded in an area that has a subsurface structure $Q(x, z)$ and a velocity model $v(x, z)$. Based on the stationary-phase approximations, this gather can be modeled by a Kirchhoff integral:

$$P(t, x_r) = \int_{x,z} Q(x, z) A(x, z, x_s) A(x, z, x_r) \delta(t - \tau(x, z; x_s) - \tau(x, z; x_r)) dx dz \quad (1)$$

where $A(x, z; x_c)$ and $\tau(x, z; x_c), x_c = x_s$ or x_r, are an amplitude function and a traveltime function respectively. The traveltime function can be found by solving the Eikonal equation:

$${\tau_x}^2 + {\tau_z}^2 = v^{-2}(x, z) \quad (2)$$

with initial condition $\tau(0, x_c, x_c) = 0$.

If this gather is migrated with a different velocity model $\hat{v}(x, z)$, the image obtained will be distorted from the true image of the subsurfaces.

$$\hat{Q}(\hat{x}, \hat{z}) = \int_{t,x_r} P(t, x_r) \hat{A}(\hat{x}, \hat{z}; x_s) \hat{A}(\hat{x}, \hat{z}; x_r) \delta(t - \hat{\tau}(\hat{x}, \hat{z}; x_s) - \hat{\tau}(\hat{x}, \hat{z}; x_r)) dt dx_r \quad (3)$$

where $\hat{\tau}(x, z; x_c), x_c = x_s$ or x_r is the traveltime function with the velocity model $\hat{v}(x, z)$.

Residual migration is a transformation from image $\hat{Q}(\hat{x}, \hat{x})$ to image $Q(x, z)$. Therefore, we want to find out the relationship between (x, z) and $(\hat{x}, \hat{z})$ that defines the kinematic operators of this transformation. Substituting $P(t, x_r)$ from equation (1) into equation (3) yields

$$\begin{aligned}\hat{Q}(\hat{x},\hat{z}) &= \int_{t,x_r}\int_{x,z} Q(x,z)A(x,z;x_s)A(x,z;x_r)\delta(t-\tau(x,z;x_s)-\tau(x,z;x_r))dxdz \\ &\quad \hat{A}(\hat{x},\hat{z};x_s)\hat{A}(\hat{x},\hat{z};x_r)\delta(t-\hat{\tau}(\hat{x},\hat{z};x_s)-\hat{\tau}(\hat{x},\hat{z};x_r))dtdx_r \\ &= \int_{x,z} Q(x,z)A(x,z;x_s)\int_{x_r} A(x,z;x_r)\hat{A}(\hat{x},\hat{z};x_s)\hat{A}(\hat{x},\hat{z};x_r) \\ &\quad \delta(\tau(x,z;x_s)+\tau(x,z;x_r)-\hat{\tau}(\hat{x},\hat{z};x_s)-\hat{\tau}(\hat{x},\hat{z};x_r))dx_r dxdz.\end{aligned}$$

The image of a single scatterer at (x, z) can then be found to be:

$$\begin{aligned}\hat{Q}_0(\hat{x},\hat{z}) &= A(x,z;x_s)\int_{x_r} A(x,z;x_r)\hat{A}(\hat{x},\hat{z};x_s)\hat{A}(\hat{x},\hat{z};x_r) \\ &\quad \delta(\tau(x,z;x_s)+\tau(x,z;x_r)-\hat{\tau}(\hat{x},\hat{z};x_s)-\hat{\tau}(\hat{x},\hat{z};x_r))dx_r.\end{aligned} \tag{4}$$

It is well known that the kinematics of the summation operator that does full migration is defined by the trajectory of the reflection event from a scatterer. Similarly, the kinematics of the summation operator that does residual migration can be determined from $\hat{Q}_0(\hat{x}, \hat{z})$, or more specificly from the argument of the δ-function in equation (4). Let

$$h(x_r) = \tau(x,z;x_s)+\tau(x,z;x_r)-\hat{\tau}(\hat{x},\hat{z};x_s)-\hat{\tau}(\hat{x},\hat{z};x_r),$$

$$G(x_r) = A(x,z;x_s)A(x,z;x_r)\hat{A}(\hat{x},\hat{z};x_s)\hat{A}(\hat{x},\hat{z};x_r).$$

Recall the properties of δ-function:

$$\begin{aligned}\hat{Q}_0(\hat{x},\hat{z}) &= \int_{x_r} G(x_r)\delta(h(x_r))dx_r \\ &= \int_{x_r} \frac{G(x_r)}{h'(x^*)}\delta(x_r - x^*)dx_r\end{aligned}$$

where $h(x^*) = 0$. Clearly, $\hat{Q}_0(\hat{x}, \hat{z})$ has extremal values when

$$\begin{cases} h(x_r) = 0 \\ h'(x_r) = 0. \end{cases}$$

Therefore the relationship between (x, z) and $(\hat{x}, \hat{z})$, is implicitly expressed by a pair of equations:

$$\begin{cases} \tau(x,z;x_s)+\tau(x,z;x_r) = \hat{\tau}(\hat{x},\hat{z};x_s)+\hat{\tau}(\hat{x},\hat{z};x_r) \\ \\ \tau_{x_s}(x,z;x_s)\dfrac{\partial x_s}{\partial x_r}+\tau_{x_r}(x,z;x_r) = \hat{\tau}_{x_s}(\hat{x},\hat{z};x_s)\dfrac{\partial x_s}{\partial x_r}+\hat{\tau}_{x_r}(\hat{x},\hat{z};x_r). \end{cases} \tag{5}$$

The partial derivative of x_s with respect to x_r is determined by the type of the data gather. For common shot gathers,

$$x_s = \text{constant} \quad \text{and} \quad \frac{\partial x_s}{\partial x_r} = 0.$$

For constant offset sections,

$$x_s = x_r - \text{offset} \quad \text{and} \quad \frac{\partial x_s}{\partial x_r} = 1.$$

For each point (x, z), equation (5) defines a curve in $(\hat{x}, \hat{z})$. This curve is exactly the kinematics of the residual-migration operator at point (x, z). For general velocity models, the traveltimes must be computed by some numerical methods. These methods generate traveltime tables rather than continuous functions. Therefore, it is natural to solve equation (5) numerically. A straight forward method is searching. For each x_r and (x, z), all points around (x, z) are checked to find the $(\hat{x}, \hat{z})$ that satisfies equation (5). But the this algorithm is time-consuming when the dimensions of images are large. Motivated by the results of the finite-difference calculation of traveltimes (Van Trier, 1990), I begin to explore the possibility to calculate residual-migration operators with finite-difference techniques.

Finite-difference method

To use a finite-difference approach, we need to derive formulas for partial derivatives:

$$\frac{\partial \hat{x}}{\partial x} \quad \text{and} \quad \frac{\partial \hat{z}}{\partial x},$$

or

$$\frac{\partial \hat{x}}{\partial z} \quad \text{and} \quad \frac{\partial \hat{z}}{\partial z}.$$

Let us first simplify notations by denoting

$$\tau(x_c) = \tau(x, z; x_c), \qquad \hat{\tau}(x_c) = \hat{\tau}(\hat{x}, \hat{z}; x_c)$$

and

$$\tau_{x_r}(x_c) = \tau_{x_c}(x, z; x_c)\frac{\partial x_c}{\partial x_r}, \qquad \hat{\tau}_{x_r}(x_c) = \hat{\tau}_{x_c}(\hat{x}, \hat{z}; x_c)\frac{\partial x_c}{\partial x_r}$$

where $x_c = x_s$ or x_r. Now we define

$$F(x, z, \hat{x}, \hat{z}) = \tau(x_s) + \tau(x_r) - \hat{\tau}(x_s) - \hat{\tau}(x_r) \tag{6}$$

$$G(x, z, \hat{x}, \hat{z}) = \tau_{x_r}(x_s) + \tau_{x_r}(x_r) - \hat{\tau}_{x_r}(x_s) - \hat{\tau}_{x_r}(x_r). \tag{7}$$

Then, equation (5) becomes

$$\begin{cases} F(x, z, \hat{x}, \hat{z}) = 0 \\ G(x, z, \hat{x}, \hat{z}) = 0. \end{cases} \tag{8}$$

Taking partial derivative with respect to x on both sides of equation (8) gives:

$$\begin{cases} F_x + F_{\hat{x}}\dfrac{\partial \hat{x}}{\partial x} + F_{\hat{z}}\dfrac{\partial \hat{z}}{\partial x} = 0 \\[2ex] G_x + G_{\hat{x}}\dfrac{\partial \hat{x}}{\partial x} + G_{\hat{z}}\dfrac{\partial \hat{z}}{\partial x} = 0. \end{cases} \tag{9}$$

Similarly, taking partial derivative with respect to z yields

$$\begin{cases} F_z + F_{\hat{x}} \dfrac{\partial \hat{x}}{\partial z} + F_{\hat{z}} \dfrac{\partial \hat{z}}{\partial z} & = 0 \\ G_z + G_{\hat{x}} \dfrac{\partial \hat{x}}{\partial z} + G_{\hat{z}} \dfrac{\partial \hat{z}}{\partial z} & = 0. \end{cases} \tag{10}$$

In matrix notations, these equations become:

$$\begin{pmatrix} F_{\hat{x}} & F_{\hat{z}} \\ G_{\hat{x}} & G_{\hat{z}} \end{pmatrix} \begin{pmatrix} \dfrac{\partial \hat{x}}{\partial x} \\ \dfrac{\partial \hat{z}}{\partial x} \end{pmatrix} = - \begin{pmatrix} F_x \\ G_x \end{pmatrix} \tag{11}$$

$$\begin{pmatrix} F_{\hat{x}} & F_{\hat{z}} \\ G_{\hat{x}} & G_{\hat{z}} \end{pmatrix} \begin{pmatrix} \dfrac{\partial \hat{x}}{\partial z} \\ \dfrac{\partial \hat{z}}{\partial z} \end{pmatrix} = - \begin{pmatrix} F_z \\ G_z \end{pmatrix}. \tag{12}$$

Solving these linear equation sets analytically, we can obtain two sets of partial differential equations for $(\hat{x}, \hat{z})$,

$$\begin{cases} \dfrac{\partial \hat{x}}{\partial x} & = \dfrac{F_{\hat{z}} G_x - F_x G_{\hat{z}}}{F_{\hat{x}} G_{\hat{z}} - F_{\hat{z}} G_{\hat{x}}} \\ \dfrac{\partial \hat{z}}{\partial x} & = \dfrac{F_x G_{\hat{x}} - F_{\hat{x}} G_x}{F_{\hat{x}} G_{\hat{z}} - F_{\hat{z}} G_{\hat{x}}} \end{cases} \tag{13}$$

and

$$\begin{cases} \dfrac{\partial \hat{x}}{\partial z} & = \dfrac{F_{\hat{z}} G_z - F_z G_{\hat{z}}}{F_{\hat{x}} G_{\hat{z}} - F_{\hat{z}} G_{\hat{x}}} \\ \dfrac{\partial \hat{z}}{\partial z} & = \dfrac{F_z G_{\hat{x}} - F_{\hat{x}} G_z}{F_{\hat{x}} G_{\hat{z}} - F_{\hat{z}} G_{\hat{x}}}. \end{cases} \tag{14}$$

On the right hand side of these equations, we have the partial derivatives of F and G. These partial derivatives are related to the partial derivatives of traveltimes. From equations (6) and (7), we can derive:

$$\begin{array}{lcll} F_{\hat{x}} & = & -\hat{\tau}_{\hat{x}}(x_s) - \hat{\tau}_{\hat{x}}(x_r) & \quad F_{\hat{z}} = -\hat{\tau}_{\hat{z}}(x_s) - \hat{\tau}_{\hat{z}}(x_r) \\ G_{\hat{x}} & = & -\hat{\tau}_{\hat{x}x_r}(x_s) - \hat{\tau}_{\hat{x}x_r}(x_r) & \quad G_{\hat{z}} = -\hat{\tau}_{\hat{z}x_r}(x_s) - \hat{\tau}_{\hat{z}x_r}(x_r) \\ F_x & = & \tau_x(x_s) + \tau_x(x_r) & \quad F_z = \tau_z(x_s) + \tau_z(x_r) \\ F_x & = & \tau_{xx_r}(x_s) + \tau_{xx_r}(x_r) & \quad F_z = \tau_{zx_r}(x_s) + \tau_{zx_r}(x_r) \end{array}$$

Because the upwind finite-difference algorithm calculates accurate traveltimes on a regular grid, I use this trave-time table to compute the derivatives of traveltimes, and in turn to compute the expressions on the right hand side of equations (13) and (14). Therefore, equations (13) and (14) become linear partial differential equations and are easy to be solved by many existing numerical algorithms. I use the leapfrog method, in which a partial derivative is taken as a difference between two neighboring points:

$$\frac{\partial y}{\partial x} = \frac{y(x+\Delta x) - y(x-\Delta x)}{2\Delta x}.$$

To obtain an initial solution, I use searching method to find the operator at some point. The finite-difference algorithm can then calculate the operators elsewhere, and step by step.

EXPERIMENTS

In the first experiment, I show the accuracy of the algorithm. In the second experiment, I demonstrate that residual migration can indeed be done by the residual migration operators formulated in this paper.

Accuracy of the algorithm

Using the finite-difference algorithm described in this paper, I calculate the kinematic residual-migration operators that convert a common shot profile migrated with a initial constant velocity to a common shot profile migrated with a new constant velocity. I assume that the operators are known at the bottom of the model, and use them as the initial solutions. The partial differential equations are solved from the bottom of the model to the top of the model. I also assume that all the partial derivatives of traveltimes are calculated precisely.

For fixed shot and receiver positions, the operators define a mapping between $(\hat{x}, \hat{z})$ and (x, z). Figures 1 and 2 show $\hat{z}$ and $\hat{x}$ as the functions of z and x. The initial velocity is 1.8 km/s for Figure 1 and 2.2 km/s for Figure 2. The new velocity is 2 km/s for both cases. The grid is evenly sampled in depth and laterally, with a sampling interval of 20 m. Figures 3 and 4 display the differences between these computed functions and the analytical solutions. We see that the errors are concentrated in the transition regions where $\hat{z}$ changes from positive to negative. In Figure 5, I plot the errors at the depth of 1.2 km, as functions of the horizontal axis. The maximum error at this depth is about two order of magnitude smaller than the spatial sampling-intervals.

Residual profile migration

In this experiment, I use the operators formulated in this paper to perform residual profile migration. Synthetic shot gathers are generated from a velocity model that contains a scatterer and a dipping reflector. Stacked sections in Figure 6 show that the operators can focus the image of the scatter and can migrate the distorted image of a dipping reflector to the correct positions. The amplitudes of the dipping

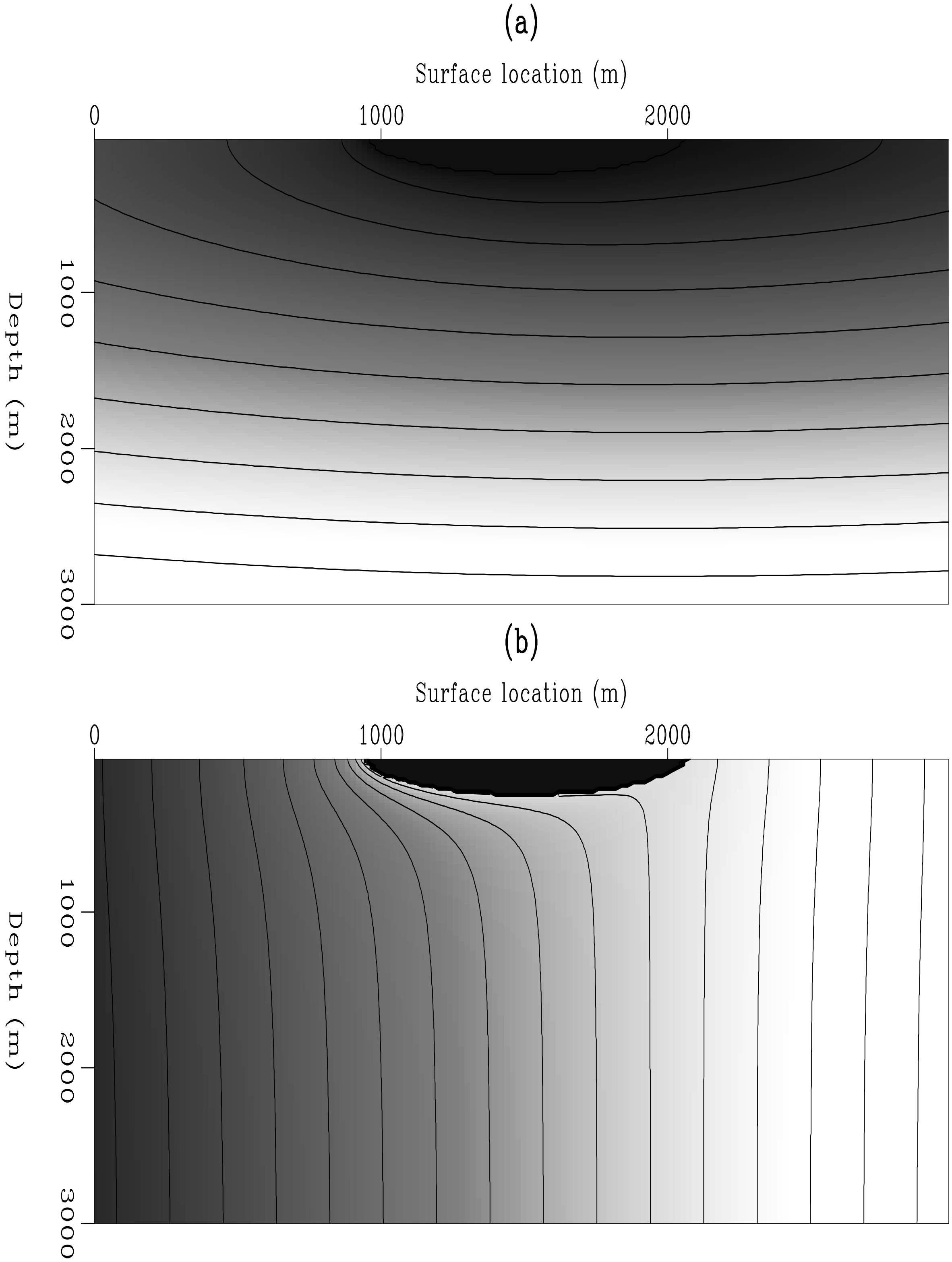

Figure 1: Residual profile migration operators computed by the finite-difference algorithm: (a) $\hat{z}$ as function of x and z; (b) $\hat{x}$ as function of x and z. The shot is at surface location 1 km, and the receiver is at surface location 2 km. The ratio between the initial velocity and new velocity is 0.9. The functions are displayed by both intensity and contour. Low intensity represents large values of the functions. Contour lines are drawn at 0.28 km intervals for $\hat{z}$, and 0.15 km intervals for $\hat{x}$. lin2-fdzxmap1 [NR]

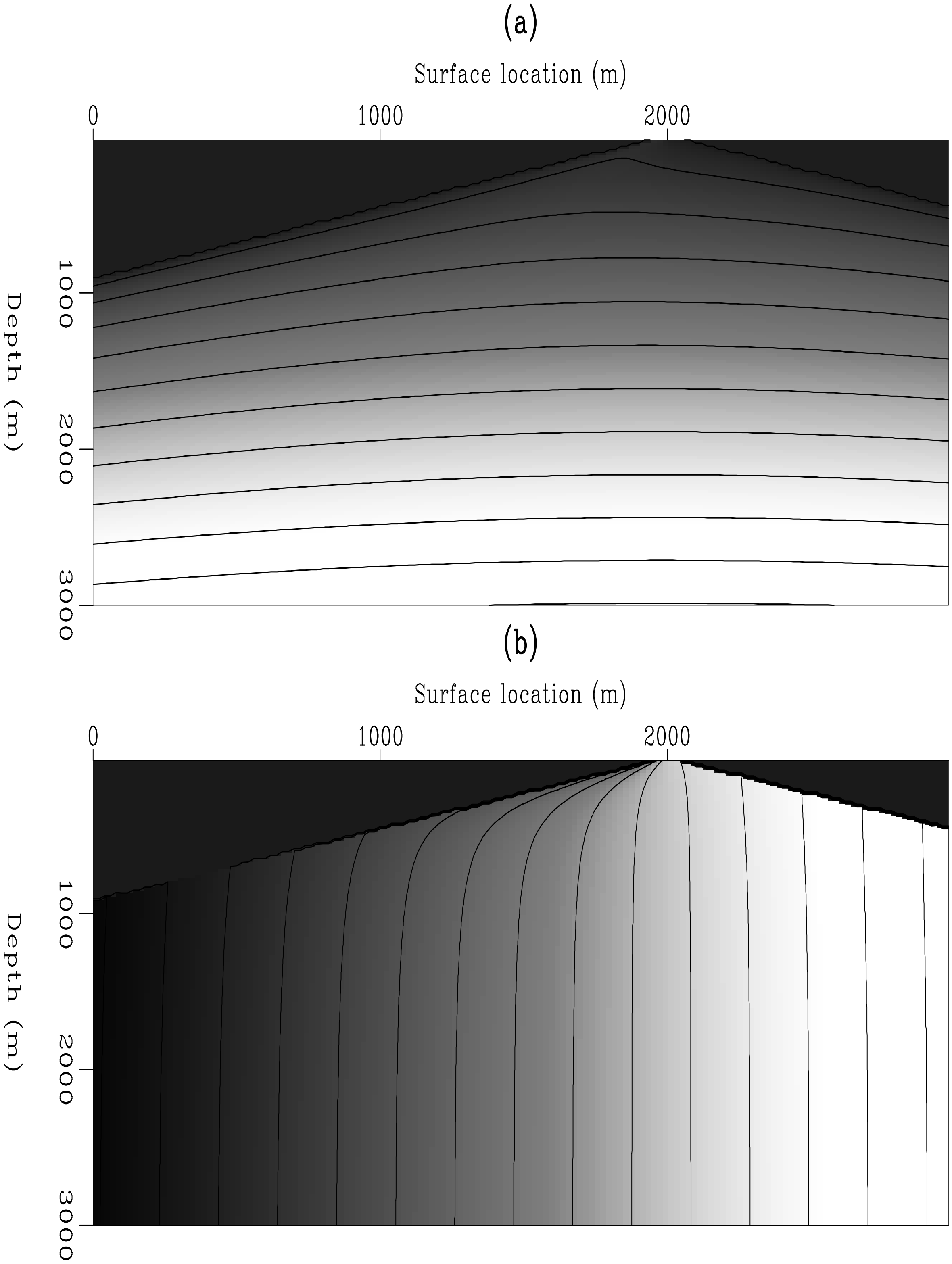

Figure 2: Residual profile migration operators computed by the finite-difference algorithm: (a) $\hat{z}$ as function of x and z; (b) $\hat{x}$ as function of x and z. The shot is at surface location 1 km, and the receiver is at surface location 2 km. The ratio between the initial velocity and new velocity is 1.1. The functions are displayed by both intensity and contour. Low intensity represents large values of the functions. Contour lines are drawn at 0.3 km intervals for $\hat{z}$, and 0.25 km intervals for $\hat{x}$. $\boxed{\text{lin2-fdzxmap2}}$ [NR]

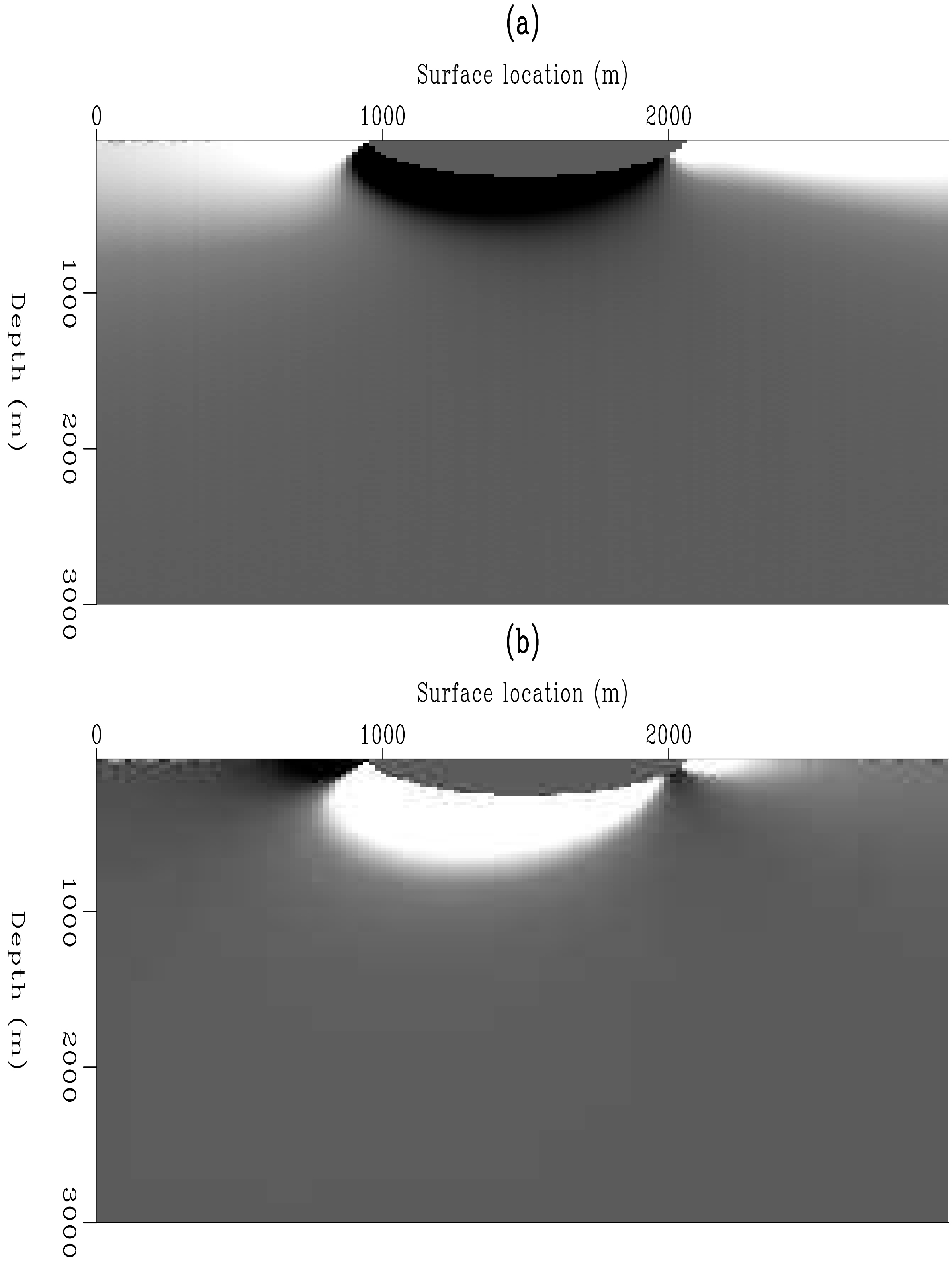

Figure 3: Differences between the functions shown in Figure 1 and the analytical solutions: (a) difference of $\hat{z}$; (b) difference of $\hat{x}$. lin2-anfdzxmap1 [NR]

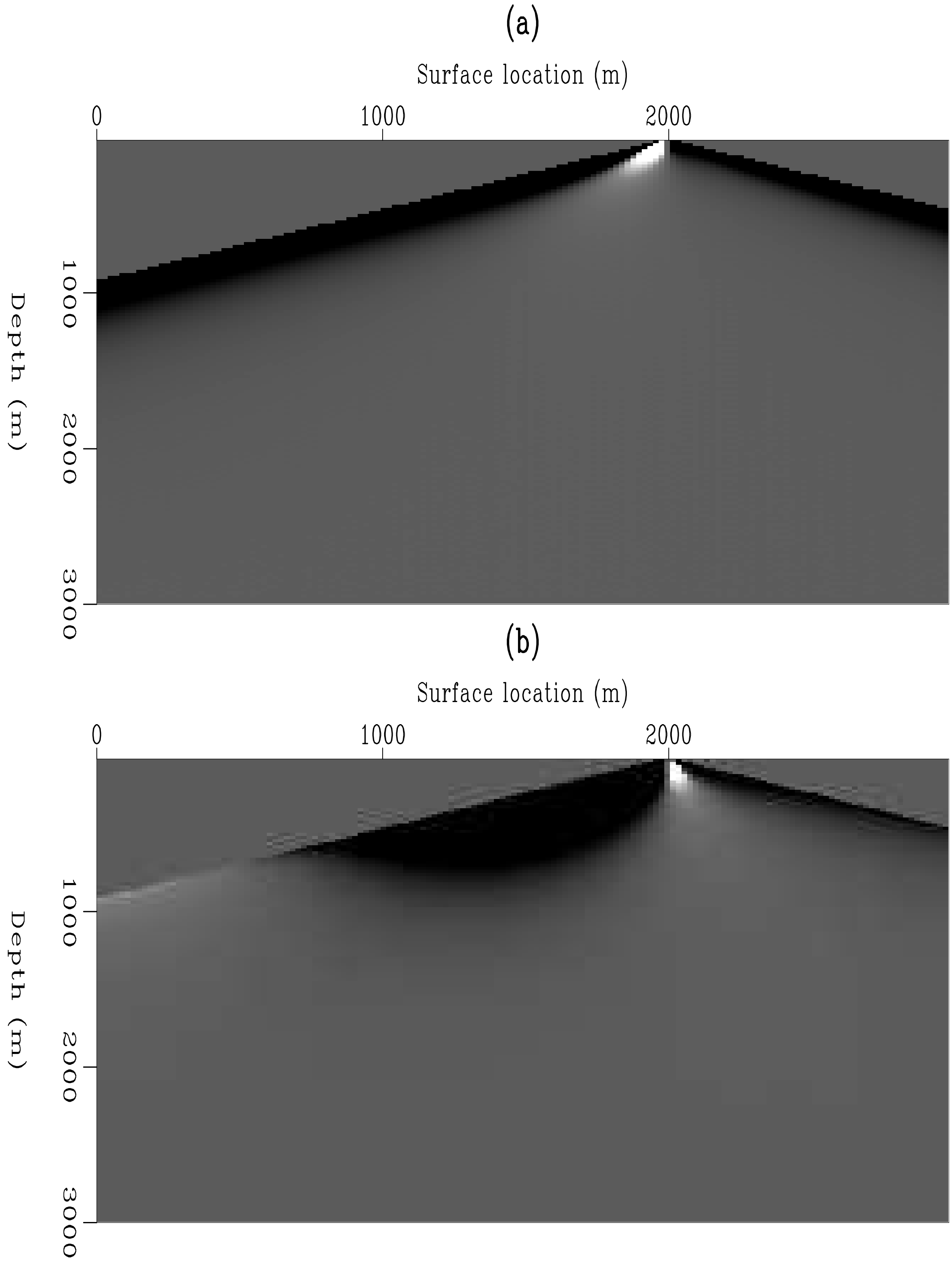

Figure 4: Differences between the functions shown in Figure 2 and the analytical solutions: (a) difference of $\hat{z}$; (b) difference of $\hat{x}$. lin2-anfdzxmap2 [NR]

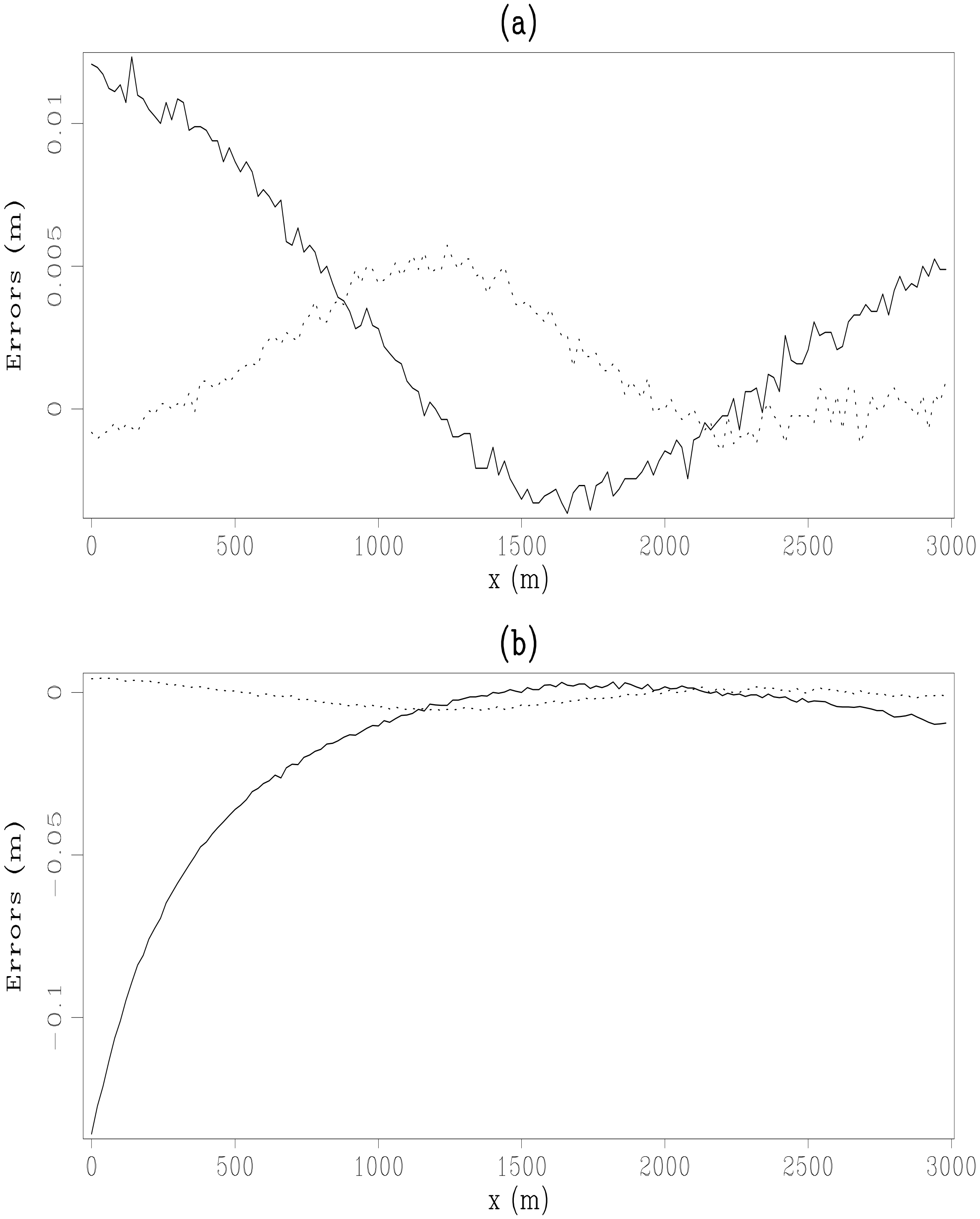

Figure 5: Errors of the calculated residual-migration operators at depth 1.2 km: (a) the ratio between the initial velocity and new velocity is 0.9; (b) the ratio between the initial velocity and new velocity is 1.1. The solid lines represent $\hat{z}$-functions, and dotted lines represent $\hat{x}$-functions. lin2-errslice [NR]

reflector are obviously incorrect. This is because the operators come from kinematic considerations.

CONCLUSIONS

I have presented an efficient algorithm for calculating residual migration operators. This algorithm is generally applicable to both post-stack and pre-stack migration, and to both common shot and constant offset geometry. I have used examples to demonstrate the accuracy of the algorithm. However, the conclusions drawn from these examples are based on the assumptions of having accurate partial derivatives of traveltimes. To understand how the errors in the traveltime calculation would affect the results of the algorithm, more experiments need to be done.

REFERENCES

Al-Yahya, K., and Fowler, P., 1986, Prestack residual migration: SEP–**50**, 219-229.

Etgen, J., 1989, Kinematic residual prestack migration: SEP–**61**, 79-101.

Rothman, D.H., Levin, S.A., and Rocca, F., 1985, Residual migration: applications and limitations: Geophysics, **50**, 110-126.

Zhang, L., 1990, Refining the image of profile migration: residual moveout and residual migration: SEP–**65**, 29-40.

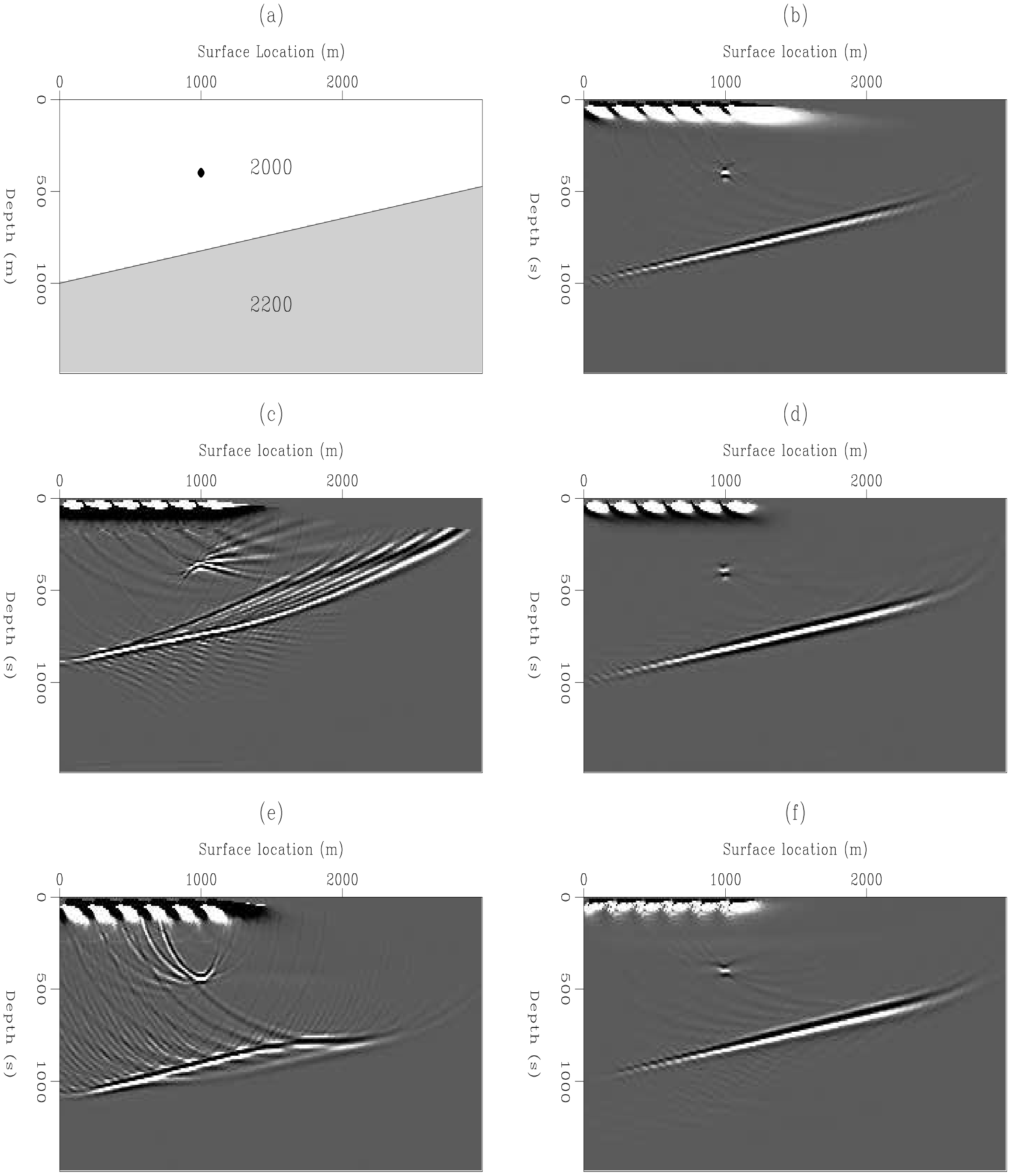

Figure 6: Synthetic examples. Six shot records are generated from the velocity model (a) by using a finite-difference algorithm. These profiles are stacked after: (b) migration with the correct velocity; (c) migration with a velocity of −10% error; (d) migration in (c) followed by the residual migration; (e) migration with the velocity of +10% error; (f) migration in (e) followed by the residual migration. lin2-synexa [NR]