Stanford Exploration Project, Report 70, December 18, 1997, pages 1–92

Stanford Exploration Project, Report 70, December 18, 1997, pages 1–92

Residual depth migration

Lin Zhang[1]

keywords: *algorithm, depth migration, modeling*

ABSTRACT

Residual depth migration is a process of converting an image migrated with one slowness model to an image migrated with another slowness model. The kinematic operators of this process are defined implicitly in a pair of nonlinear equations. This nonlinear system can be efficiently solved by first transforming it into the initial value problem of a pair of partial differential equations, and then solving the partial differential equations with a finite-difference scheme. Once the operators are calculated, the residual depth migration can be accomplished with Kirchhoff-type integrals. The proposed algorithm handles general variable-slowness models, and it is applicable to both pre- and post-stack images. A synthetic example of residual depth migration of a post-stack image shows that the algorithm gives kinematically accurate results.

INTRODUCTION

When perturbations of velocity models are small, residual migration may be considerably more efficient than full migration. In a classic paper, Rothman et al. (1985) described residual time 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 certain 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. He calculated the residual migration operators by solving a pair of nonlinear equations numerically. In SEP–65 (Zhang, 1990a), I derived analytical formulas for calculating the kinematic operators of residual profile migration. However, these results are limited to the cases in which migration velocities are

[1]**email:** not available

constant.

In my last report (Zhang, 1990b), I described a conceptual algorithm for calculating residual depth migration operators. The algorithm is generally applicable to both pre- and post-stack images. In principle, it can handle general variable-slowness models. This paper explains in detail the procedure for implementing this algorithm and demonstrates the results of the algorithm with a synthetic example. The next section reviews the key equations I derived in the last report. The sections that follow explain the algorithm step by step, including how to calculate the partial derivatives of traveltimes and surface horizontal slownesses with an upwind finite-difference method and how to extrapolate the operators starting from the known initial conditions at the surface. The procedure for applying the operators to migrated images is briefly described. Finally, the report includes a synthetic example of residual depth migration of a post-stack image.

REVIEW

In this section, I review the concepts and key equations I developed previously.

Kinematic relations

Let us suppose that a data gather $P(t, x_r)$ is recorded in an area that has a subsurface structure $Q(x, z)$ and slowness model $s(x, z)$. If this gather is migrated with a different slowness model $\hat{s}(x, z)$, the resulting image $\hat{Q}(\hat{x}, \hat{z})$ will be a distortion of the true image of the subsurfaces.

Residual depth migration is the transformation from image $\hat{Q}(\hat{x}, \hat{x})$ to image $Q(x, z)$. The kinematic operator that effects this transformation is defined by the mapping functions between the image locations (x, z) and $(\hat{x}, \hat{z})$. In my last report, I showed that these mapping functions are expressed implicitly in the following pair of equations:

$$\begin{cases} \tau(x, z; x_s) + \tau(x, z; x_r) = \hat{\tau}(\hat{x}, \hat{z}; x_s) + \hat{\tau}(\hat{x}, \hat{z}; x_r) \\ \\ p(x, z; x_s)\dfrac{\partial x_s}{\partial x_r} + p(x, z; x_r) = \hat{p}(\hat{x}, \hat{z}; x_s)\dfrac{\partial x_s}{\partial x_r} + \hat{p}(\hat{x}, \hat{z}; x_r). \end{cases} \tag{1}$$

where x_s is the surface location of shot and x_r is the surface location of receiver. The terms τ and $\hat{\tau}$ are traveltimes associated with slowness models $s(x, z)$ and $\hat{s}(\hat{x}, \hat{z})$, respectively. The terms p and $\hat{p}$ represent surface horizontal slownesses defined as follows

$$\begin{cases} p(x, z; x_c) & = & \dfrac{\partial \tau}{\partial x_c} & = & s(x_c, 0)\sin\alpha \\ \\ \hat{p}(\hat{x}, \hat{z}; x_c) & = & \dfrac{\partial \hat{\tau}}{\partial x_c} & = & \hat{s}(x_c, 0)\sin\hat{\alpha} \end{cases} \tag{2}$$

where $x_c = x_s$ or x_r. The angles α and $\hat{\alpha}$ are between ray directions and the vertical at the surface.

The partial derivative of x_s with respect to x_r depends on the type of a data gather. For post-stack data,

$$x_s = x_r \qquad \Longrightarrow \qquad \frac{\partial x_s}{\partial x_r} = 1.$$

For pre-stack data with common shot geometry,

$$x_s = \text{constant} \qquad \Longrightarrow \qquad \frac{\partial x_s}{\partial x_r} = 0,$$

with constant offset geometry,

$$x_s = x_r - \text{offset} \qquad \Longrightarrow \qquad \frac{\partial x_s}{\partial x_r} = 1.$$

Therefore, equation (1) is generally applicable.

For each image location (x, z) on $Q(x, z)$, equation (1) implicitly tells us the corresponding image location $(\hat{x}, \hat{z})$ on $\hat{Q}(\hat{x}, \hat{z})$. We want to solve $(\hat{x}, \hat{z})$ in this equation as functions of (x, z). For general slowness models, this nonlinear equation set has to be solved using a numerical method. Searching is a straightforward method, in which, for each x_r and (x, z), all points around (x, z) are checked to find the $(\hat{x}, \hat{z})$ that satisfies equation (1). However, this algorithm is time-consuming, especially when the image dimensions are large.

I have developed an algorithm that solves equation (1) using finite-difference techniques. The ideas are similar to those used in the finite-difference calculation of traveltimes (Van Trier, 1990). The solution of equation (1) is known on the surface. We can extrapolate this solution in depth once we know the derivative of the solution with respect to the depth.

Partial differential equations

To use a finite-difference approach, we need to find the expressions for partial derivatives:

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

or

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

In my preceding paper (Zhang,1990b), I showed that

$$\left\{ \begin{array}{lcl} \dfrac{\partial \hat{x}}{\partial x} & = & \dfrac{F_x \hat{G}_{\hat{z}} - \hat{F}_{\hat{z}} G_x}{\hat{F}_{\hat{x}} \hat{G}_{\hat{z}} - \hat{F}_{\hat{z}} \hat{G}_{\hat{x}}} \\ \\ \dfrac{\partial \hat{z}}{\partial x} & = & \dfrac{\hat{F}_{\hat{x}} G_x - F_x \hat{G}_{\hat{x}}}{\hat{F}_{\hat{x}} \hat{G}_{\hat{z}} - \hat{F}_{\hat{z}} \hat{G}_{\hat{x}}} \end{array} \right. \tag{3}$$

and

$$\left\{ \begin{array}{lcl} \dfrac{\partial \hat{x}}{\partial z} & = & \dfrac{F_z \hat{G}_{\hat{z}} - \hat{F}_{\hat{z}} G_z}{\hat{F}_{\hat{x}} \hat{G}_{\hat{z}} - \hat{F}_{\hat{z}} \hat{G}_{\hat{x}}} \\ \\ \dfrac{\partial \hat{z}}{\partial z} & = & \dfrac{\hat{F}_{\hat{x}} G_z - F_z \hat{G}_{\hat{x}}}{\hat{F}_{\hat{x}} \hat{G}_{\hat{z}} - \hat{F}_{\hat{z}} \hat{G}_{\hat{x}}} \end{array} \right. \tag{4}$$

where

$$\begin{array}{lcllcl} \hat{F}_{\hat{x}} & = & \hat{\tau}_{\hat{x}}(\hat{x}, \hat{z}; x_s) + \hat{\tau}_{\hat{x}}(\hat{x}, \hat{z}; x_r) & \hat{F}_{\hat{z}} & = & \hat{\tau}_{\hat{z}}(\hat{x}, \hat{z}; x_s) + \hat{\tau}_{\hat{z}}(\hat{x}, \hat{z}; x_r) \\ F_x & = & \tau_x(x, z; x_s) + \tau_x(x, z; x_r) & F_z & = & \tau_z(x, z; x_s) + \tau_z(x, z; x_r) \\ \hat{G}_{\hat{x}} & = & \hat{p}_{\hat{x}}(\hat{x}, \hat{z}; x_s) \dfrac{\partial x_s}{\partial x_r} + \hat{p}_{\hat{x}}(\hat{x}, \hat{z}; x_r) & \hat{G}_{\hat{z}} & = & \hat{p}_{\hat{z}}(\hat{x}, \hat{z}; x_s) \dfrac{\partial x_s}{\partial x_r} + \hat{p}_{\hat{z}}(\hat{x}, \hat{z}; x_r) \\ G_x & = & p_x(x, z; x_s) \dfrac{\partial x_s}{\partial x_r} + p_x(x, z; x_r) & G_z & = & p_z(x, z; x_s) \dfrac{\partial x_s}{\partial x_r} + p_z(x, z; x_r). \end{array}$$

If the traveltime table is calculated with polar coordinates (r, θ), instead of Cartesian coordinates (x, z), then we can simply change x and z in equations (3) and (4) to θ and r, respectively.

With equations (3) and (4), we can extrapolate the operators in both the horizontal and vertical directions. Because the operators are usually known at the surface, one can use equation (4) to extrapolate the operators vertically to each depth. The next section shows how this extrapolation is done.

CALCULATING THE OPERATORS

In this section, I explain in detail the procedure for calculating the operator.

Derivatives of traveltimes

For a general variable-slowness model, traveltimes can be calculated on a regular grid by using a finite difference scheme. Two such algorithms are commonly used, Vidale's method and Van Trier's method. Vidale (1988) approximates the eikonal equation by finite differences and solves directly for traveltime using a planar or circular wavefront extrapolation. In Van Trier's method (Van Trier and Symes, 1990), the eikonal equation is transformed into a hyperbolic conservation law that relates the gradient components of traveltimes. A first-order upwind finite-difference scheme is used to solve this transformed equation and obtain the partial derivatives of the traveltimes. The traveltimes are then calculated by integrating these partial derivatives along any chosen axis.

Comparing his method with Vidale's method, Van Trier pointed out that his algorithm is vectorizable and thus more efficient. For the present purposes, there is also another reason to choose Van Trier's method. The partial differential equation

(4) contains the partial derivatives of traveltimes, but not the traveltimes themselves. If we use Vidale's method, the partial derivatives of traveltimes have to be calculated by the finite difference approximations

$$\frac{\partial \tau}{\partial x} = \frac{\Delta \tau}{\Delta x} \qquad \frac{\partial \tau}{\partial z} = \frac{\Delta \tau}{\Delta z}.$$

Since the use of finite difference operators increases numerical noise, it it should be avoided whenever possible. With Van Trier's method, we do not have this problem because the partial derivatives of traveltimes are intermediate results of the algorithm. For reasons of stability, we need the partial derivatives to be smooth. Van Trier's algorithm sometimes gives rough derivative functions because the finite difference scheme depends on the signs of the partial derivatives that are discontinuous functions. Therefore I take some special measures to maintain smooth solutions.

Derivatives of surface horizontal slownesses

Another set of variables involved in partial differential equation (4) are partial derivatives of surface horizontal slownesses. According to the definition in equation (2), we can calculate these dependent variables by using a finite difference approximation. In Cartesian coordinates,

$$\begin{cases} p_x(x,z;x_c) & \approx & \dfrac{\Delta \tau_x(x,z;x_c)}{\Delta x_c} \\ \\ p_z(x,z;x_c) & \approx & \dfrac{\Delta \tau_z(x,z;x_c)}{\Delta x_c} \end{cases}$$

and in polar coordinates,

$$\begin{cases} p_r(r,\theta;x_c) & \approx & -\dfrac{\Delta \tau_r}{\Delta r}\sin\theta + \left(\dfrac{1}{r}\tau_\theta - \dfrac{\Delta \tau_r}{\Delta \theta}\right)\dfrac{\cos\theta}{r} + \dfrac{\Delta \tau_r}{\Delta x_c} \\ \\ p_\theta(r,\theta;x_c) & \approx & \left(\dfrac{1}{r}\tau_\theta - \dfrac{\Delta \tau_r}{\Delta \theta}\right)\sin\theta - \left(\tau_r + \dfrac{1}{r}\dfrac{\Delta \tau_\theta}{\Delta \theta}\right)\cos\theta + \dfrac{\Delta \tau_\theta}{\Delta x_c} \end{cases}$$

where $x_c = x_s$ or x_r. There are two drawbacks to using these equations. The first is the property of the difference operators that I have mentioned before. The second disadvantage is that calculating the traveltime-changes with respect to the change of shot or receiver position requires us to compute the partial derivatives of traveltimes at least twice at the neighbor points, which significantly complicates the algorithm.

The solution to this problem is to find the partial differential equations for p. Appendix A shows that p satisfies the following partial differential equation:

$$\tau_x p_x + \tau_z p_z = 0, \tag{5}$$

which implies the orthogonal relation between rays and wavefront. The equivalent of this equation in polar coordinates is

$$\tau_r p_r + \frac{1}{r^2}\tau_\theta p_\theta = 0. \tag{6}$$

To solve this equation, we can use the same algorithm as the one used for traveltime calculation. Let us use the equation in polar coordinates as an example. We define

$$u = p_\theta \quad \text{and} \quad w = p_r.$$

Equation (6) can then be rewriten as

$$w = -\frac{\tau_\theta}{r^2 \tau_r} u.$$

Because

$$\frac{\partial^2 p}{\partial\theta\partial r} = \frac{\partial^2 p}{\partial r\partial\theta} \quad \text{or} \quad \frac{\partial w}{\partial\theta} = \frac{\partial u}{\partial r},$$

we have the transformed partial differential equation

$$\frac{\partial u}{\partial r} = -\frac{\partial}{\partial\theta}\left[\frac{\tau_\theta}{r^2\tau_r}u\right]. \tag{7}$$

If s_0 is the slowness at the origin of polar coordinates, then the initial conditions are

$$p(0,\theta;x_c) = -s_0\sin\theta, \qquad u(0,\theta;x_c) = -s_0\cos\theta \quad \text{and} \quad w(0,\theta;x_c) = 0.$$

We can solve equation (7) by using the standard up-wind finite-difference method. The direct results are the partial derivatives of p which appear in equation (4).

Finite difference calculation of the operators

Now that we can calculate the partial derivatives of traveltimes and surface horizontal slownesses, the next step is to solve equation (4) with proper initial conditions. Let us take the residual depth migration of post-stack images as an example. Assume that calculations are done in polar coordinates. The initial conditions at the surface are

$$\hat{r} = 0 \qquad \text{and} \qquad \hat{\theta} = \sin^{-1}\left(\frac{s(0,\theta)}{\hat{s}(0,\hat{\theta})}\sin\theta\right)$$

where $s(0,\theta)$ and $\hat{s}(0,\hat{\theta})$ are slowness at the shot-receiver position. From these initial conditions we can extrapolate the kinematic operators of residual depth migration in the radius direction as follows:

$$\begin{cases} \hat{r}_{j+1} &= \hat{r}_j + \left[\dfrac{\partial\hat{r}}{\partial r}\right]_j \Delta r \\ \hat{\theta}_{j+1} &= \hat{\theta}_j + \left[\dfrac{\partial\hat{\theta}}{\partial r}\right]_j \Delta r \end{cases} \tag{8}$$

where j is the index of the grid in the radius direction. Finally, the calculated operators are mapped from polar coordinates to Cartesian coordinates.

APPLYING THE OPERATORS TO MIGRATED IMAGES

Compared with the calculation of the operators, the application of the operators to migrated images is relatively simple. Let us take residual depth migration of post-stacked depth image as an example. We have the following algorithm:

`Loop over surface location` x_c

`Loop over output image location` (x, z)

$$Q(x, z) = Q(x, z) + \hat{Q}(\hat{x}, \hat{z}).$$

Here the amplitudes of the operators are set to be constant. It is possible to compute more correct amplitudes of the operators with WKBJ approximation.

EXAMPLES

This section gives an example of residual depth migration of a post-stacked depth image. Figure **??** shows a synthetic slowness model that consists of several dipping layers and a low velocity cap over a high velocity dome. A synthetic zero-offset section is generated from this model, as shown in Figure **??**. If we migrate this section with a constant slowness, we will obtain the image displayed in Figure **??**. The images of the shallow reflectors are positioned correctly, because the interval slownesses in these layers are close to the constant migration slowness. On the other hand, the images of the deep reflectors are seriously mispositioned, because the interval slownesses in these layers are considerably smaller than the slowness used in the migration. The low velocity cap causes the images of the bottom of the dome to be flattened.

In order to correct the distortions of the images, I use the algorithm described in the previous sections to calculate the residual migration operator associated with the true slowness model and the initial constant migration slowness. Then, I apply this operator to the distorted image shown in Figure **??**. Figure **??** shows the output of this residual migration process. The distortion of the images of the reflectors in position are completely corrected.

CONCLUSIONS

I have presented an efficient algorithm for residual depth migration in general variable-slowness models. The algorithm consists of two steps: (1) calculating the operators with a finite-difference method, (2) applying the calculated operators to the migrated image. 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 a synthetic example of residual depth migration of a post-stack image to demonstrate the accuracy of the algorithm. Further experiments are necessary to determine how efficient the residual migration is in comparison to full migration.

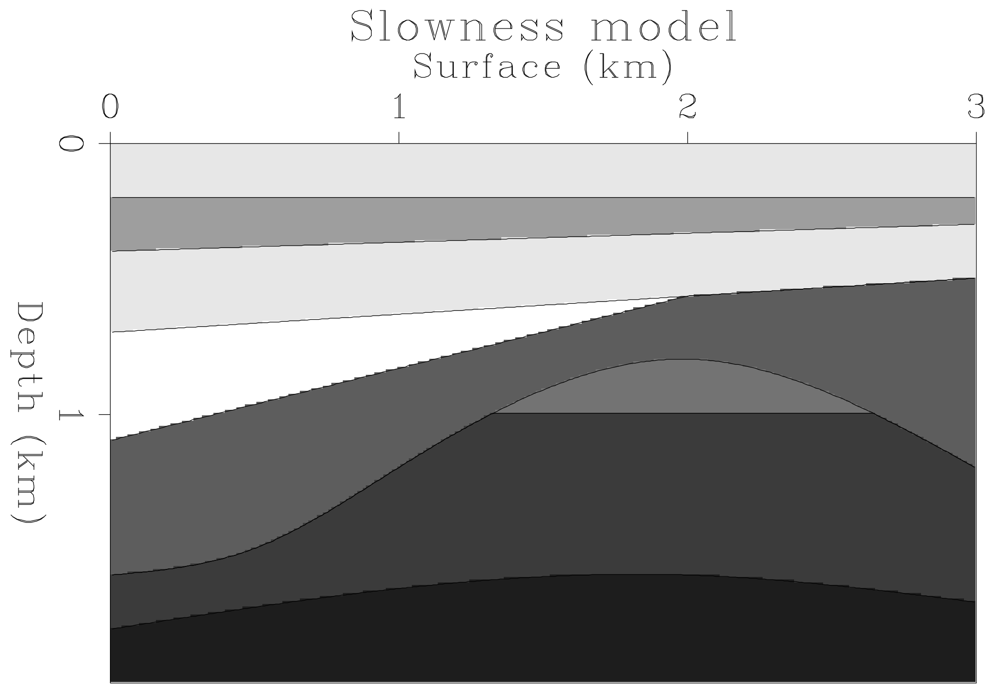

Figure 1: Synthetic slowness model. High intensities denote high velocities or low slowness. The solid curves indicate the boundaries of layers. lin1-sintl [NR]

ACKNOWLEDGMENTS

I thank John Etgen and Jos Van Trier for many interesting discussions on this subject and for sharing their wisdom in programming with me.

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.

Van Trier, J., and Symes, W.W., 1990, Upwind finite-difference calculation of traveltimes: SEP–**65**, 41-58.

Vidale, J., 1988, Finite-difference calculation of traveltimes: Bull., Seis. Soc. Am., **78**, no.6 2062-2076.

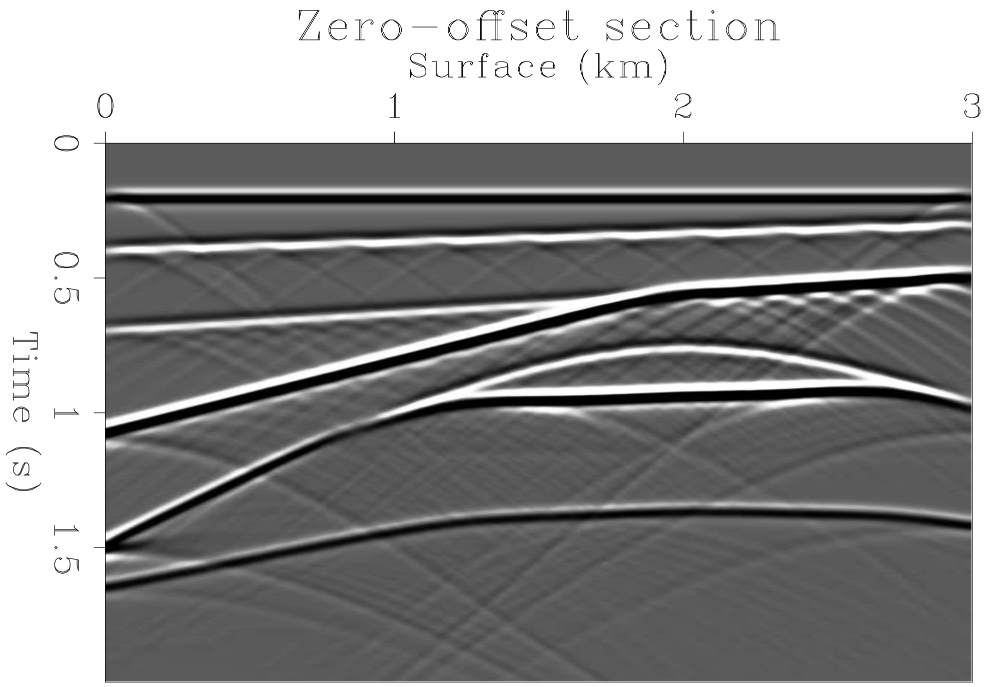

Figure 2: Synthetic zero-offset section generated from the slowness model in Figure 1. lin1-sycosdat [NR]

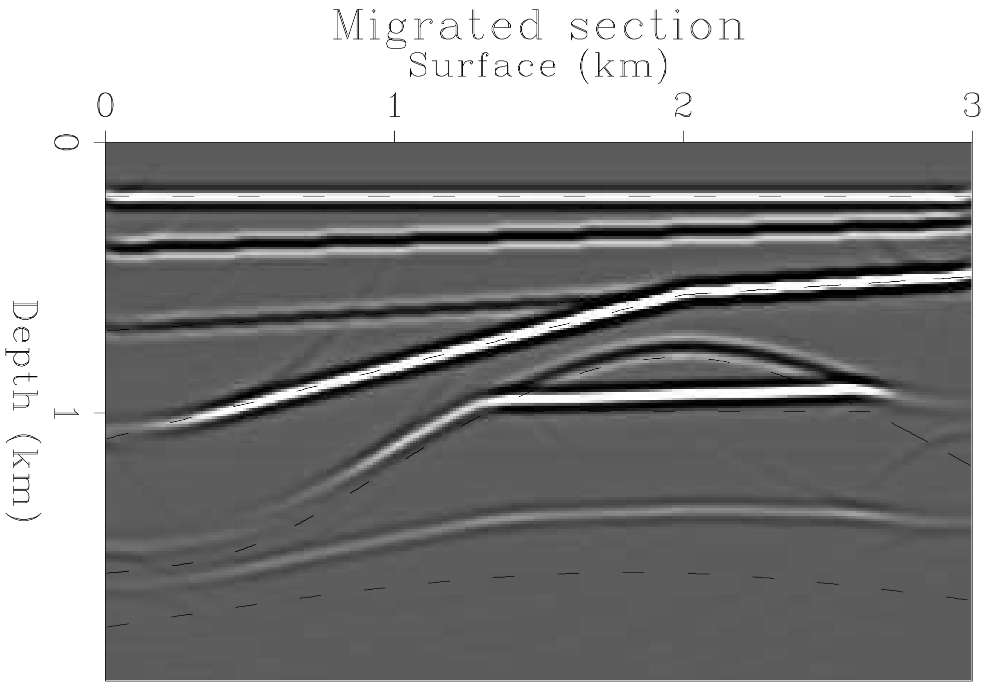

Figure 3: Image migrated with a constant slowness. The dashed curves shows the true positions of the reflectors lin1-sycosimg0 [NR]

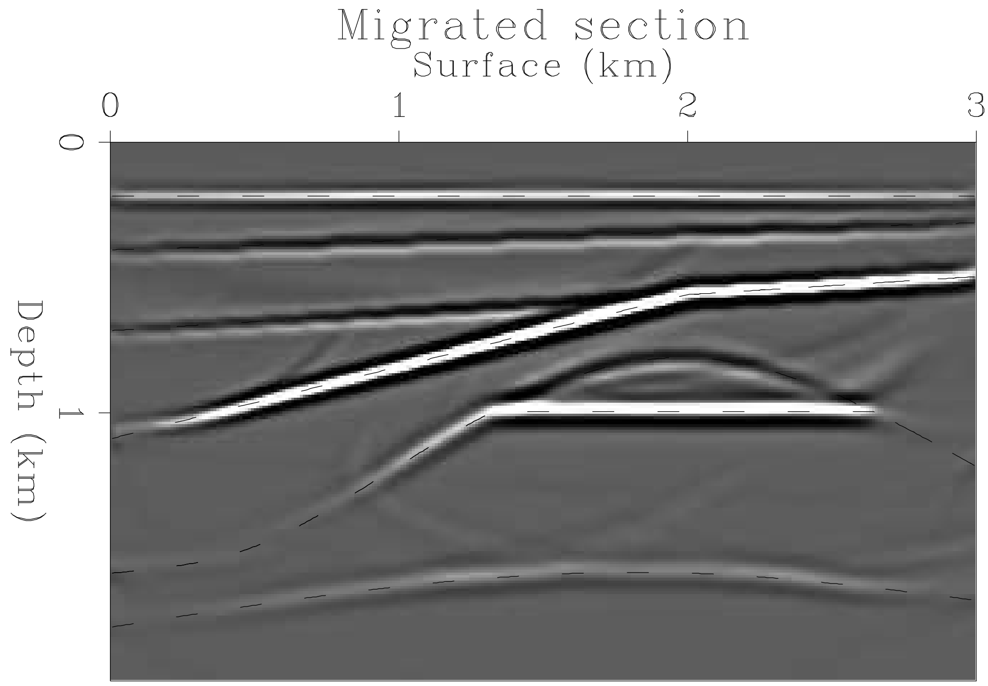

Figure 4: Image after residual depth migration. The dashed curves shows the true positions of the reflectors. lin1-sycosrmg [NR]

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

Zhang, L., 1990b, Finite-difference calculation of residual-migration operators: SEP–**67**, 93-105.

APPENDIX A

In Cartesian coordinates, the eikonal equation is

$$\tau_x^2(x,z;x_c) + \tau_z^2(x,z;x_c) = s^2(x,z) \tag{A.1}$$

where $x_c = x_s$ or x_r are shot or receiver position. Taking partial derivative of both sides of equation (A.1) with respect to x_c gives

$$2\tau_x(x,z;x_c)\tau_{xx_c}(x,z;x_c) + 2\tau_z(x,z;x_c)\tau_{zx_c}(x,z;x_c) = 0. \tag{A.2}$$

The right-hand side of the equation is zero because the slowness model is independent of shot-receiver position. Assume the derivatives of the traveltimes are continuous functions of (x,z) and x_c, we have

$$\begin{array}{rcccl} p_x(x,z;x_c) & = & \tau_{x_cx}(x,z;x_c) & = & \tau_{xx_c}(x,z;x_c) \\ p_z(x,z;x_c) & = & \tau_{x_cz}(x,z;x_c) & = & \tau_{zx_c}(x,z;x_c). \end{array} \tag{A.3}$$

Combining equations (A.2) and (A.3) gives

$$\tau_x(x,z;x_c)p_x(x,z;x_c) + \tau_z(x,z;x_c)p_z(x,z;x_c) = 0. \qquad (A.4)$$

Using the relations between Cartesian coordinates and polar coordinates:

$$\left\{ \begin{array}{lcl} x & = & x_c + r\sin\theta \\ z & = & r\cos\theta, \end{array} \right. \qquad (A.5)$$

we can derive

$$\tau_r(r,\theta;x_c)p_r(r,\theta;x_c) + \frac{1}{r^2}\tau_\theta(r,\theta;x_c)p_\theta(r,\theta;x_c) = 0. \qquad (A.6)$$