Seismic Velocity Estimation from Time Migration Velocities

M. K. Cameron, S. B. Fomel, J. A. Sethian *

October 10, 2006

Submitted for Publication: Inverse Problems, Oct. 2006

Abstract

We address the problem of estimating seismic velocities inside the earth which is necessary for obtaining seismic images in regular Cartesian coordinates. We derive a relation between the true seismic velocities and the routinely obtained so called "time migration velocities", which are some kind of mean velocities. We formulate an inverse problem and show that it is ill-posed. We suggest three different approaches for solving it. We test these algorithms on synthetic data and apply them to a field data example.

1 Introduction

Seismic data are the records of the pressure wave amplitudes P described by the wave equation

$$\Delta P(x,y,z;t) = \frac{1}{v^2(x,y,z)}\frac{\partial^2}{\partial t^2}P(x,y,z;t), \tag{1}$$

where $v(x,y,z)$, common referred to as the "seismic velocity", is the sound speed of the earth. This velocity is typically unknown, and its determination is the subject of the present work.

One common fast and robust process of obtaining seismic images is called *time migration* (see e.g. [Yilmaz, 2001]). This process is considered adequate for the areas with mild lateral velocity variation, i.e. where v depends mostly on z and only slightly on x and y. However, even mild lateral velocity variations can significantly distort subsurface structures on the time migrated images. Moreover, time migration produces images in very specific time migration coordinates $(\mathbf{x}_0, t_0)$ (explained below), and the relation between them and the Cartesian coordinates can be nontrivial if the velocity varies laterally.

One "side product" of time migration is mean velocities $\mathbf{v}_m(\mathbf{x}_0, t_0)$, known as *time migration velocities*. We will refer to them as *migration velocities* for brevity. In the case where the seismic velocity depends only on the depth, these velocities are close to the root-mean-square (RMS) velocities [Dix, 1955]. In the general case, these velocities relate to the radius of curvature of the emerging wave front [Hubral and Krey, 1980].

An alternative approach to obtaining seismic images is called *depth migration* [Yilmaz, 2001]. This approach is adequate for areas with lateral velocity variation, and produces seismic images

*This work was supported in part by the Applied Mathematical Science subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract Number DE-AC03-76SF00098, and by the Computational Mathematics Program of the National Science Foundation

in regular Cartesian coordinates. The major problem with this approach is that its implementation requires the construction of a *velocity model* for the seismic velocity $v(x, y, z)$. It can be both difficult and time consuming to construct an adequate velocity model: an iterative approach of guesswork followed by correction is often employed.

The main idea of this work is to construct a velocity model $v(\mathbf{x})$ from the migration velocities given in the time migration coordinates $(\mathbf{x}_0, t_0)$ (see a block-scheme in Fig. 1) . Using these velocities one can then perform depth migration to obtain an improved seismic image in the Cartesian coordinates $\mathbf{x}$. As an alternative to depth migration, one can instead directly convert a time migrated image to "depth" (regular Cartesian coordinates) using the additional outputs of our construction $\mathbf{x}_0(\mathbf{x})$ and $t_0(\mathbf{x})$.

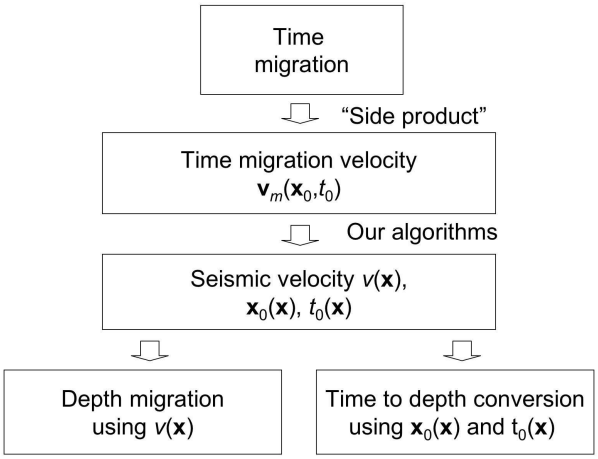


Figure 1: The main idea of this work.

Thus, our goals are to create fast and robust algorithms to:

1. Convert the migration velocities $\mathbf{v}_m(\mathbf{x}_0, t_0)$ to the true seismic velocities $v(\mathbf{x})$;
2. Convert time migrated images (in $(\mathbf{x}_0, t_0)$ coordinates) to "depth" (to images in regular Cartesian coordinates $\mathbf{x}$).

The end result is to construct more accurate seismic images cheaply and routinely.

Our results are the following. To this end, we have obtained theoretical relations between the migration velocity and the true seismic velocity in 2D and 3D. It came out in the theoretical relation in 2D that the Dix velocities $v_{Dix}(x_0, t_0)$ which are a conventional estimate of true seismic velocities from the migration velocities, can be used instead of the migration velocities as a more convenient input. We have stated an inverse problem of finding the true seismic velocity from the Dix velocity and developed two approaches for solving it: the ray tracing approach and the level set approach.

These two approaches involve three numerical algorithms:

1. An efficient time-to-depth conversion algorithm;
2. A ray tracing algorithm;
3. A level set algorithm.

The relations between the approaches and the algorithms are outlined in the block-scheme in Fig. 2.

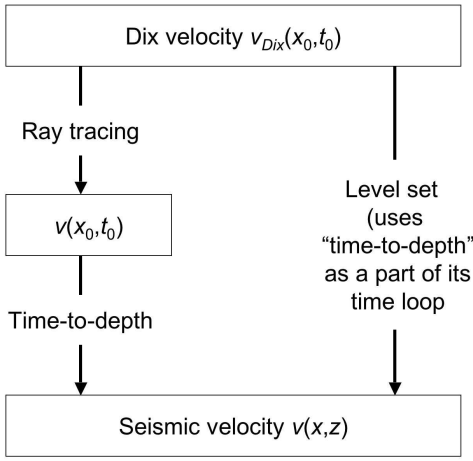


Figure 2: The approaches and the algorithms.

1.1 Time migration coordinates and image rays

For many decades, seismic imaging was based on the assumption that the velocity inside the earth depends only on the depth and that the subsurface structures are horizontal or, at worst, planar with the same dipping angle. To obtain more complex structure distortions, Hubral [Hubral, 1977] introduced the concept of the *image ray*, which gives the connection between the time migration coordinates and the regular Cartesian coordinates.

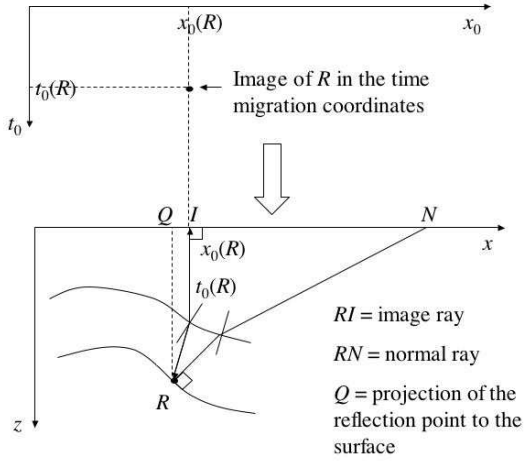


Figure 3: Image rays and time migration coordinates.

To explain this idea, we begin with the *high frequency approximation* applied to the wave equation (1), in which the wave front $T(x, y, z)$ propagates according to the Eikonal equation (see e.g. [Popov, 2002]):

$$|\nabla T(x, y, z)|^2 = \frac{1}{v^2(x, y, z)}. \tag{2}$$

The characteristics of the Eikonal equation can be viewed as rays. Among all rays starting at the reflection point R and reaching earth's surface (Fig. 3), some have minimal travel time. These rays are called image rays, and it is easy to see that they must arrive perpendicular to the surface. The ray RI in Fig. 3 is one such image ray. Thus, we may characterize the reflection point R in one of two coordinate systems: either (1) its natural Cartesian coordinates $\mathbf{x}$ or (2) the point on the surface such that an image ray leaving $\mathbf{x}_0$ and traveling for a given time reaches the point R. The former given are called depth coordinates, while the latter are called time migration coordinates.

The conventional time migration coordinates are $(\mathbf{x}_0, t_0)$ where $\mathbf{x}_0$ is the escape location of the image ray, and t_0 is the doubled (two-way) travel time along it. Note that the lateral position of the point R in the time migrated image is determined namely by the escape location of the image ray I rather than by its projection Q to the surface (Fig. 3).

1.2 Travel time approximation

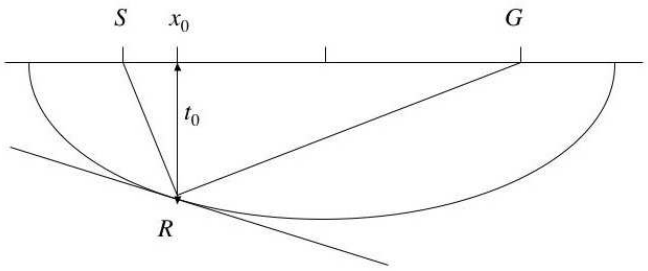


Figure 4: Travel time approximation.

Let S be a source and G be a receiver (Fig. 4), and let R be the reflection point. Suppose first that the velocity v inside the earth is constant. Then the total travel time from S to R and from R to G is:

$$t_{SR} + t_{RG} = \sqrt{\frac{t_0^2}{4} + \frac{|\mathbf{x}_0 - S|^2}{v^2}} + \sqrt{\frac{t_0^2}{4} + \frac{|\mathbf{x}_0 - G|^2}{v^2}}. \tag{3}$$

The ellipse in Fig. 4 is the locus of the reflection points A such that the total travel time $t_{SA} + t_{AG}$ is the same as $t_{SR} + t_{RG}$.

In the general case where the velocity inside the earth is arbitrary, formula (3) serves as a travel time approximation for time migration (namely its modern variant called "prestack time migration"), see [Yilmaz, 2001]. In this approximation, $\mathbf{x}_0$ is the escape location of an image ray from the reflection point R, and t_0 is the two-way travel time along it. The velocity v present in formula (3) is replaced with a parameter with dimension of velocity which depends on $\mathbf{x}_0$ and t_0. These parameters are called the *migration velocities* and denoted by $v_m(\mathbf{x}_0, t_0)$. They are chosen to provide the best fit to formula (3) in the process of time migration. Thus, formula (3) can be rewritten as

$$t(S, G, \mathbf{x}_0, t_0) = \sqrt{\frac{t_0^2}{4} + \frac{|\mathbf{x}_0 - S|^2}{v_m^2(\mathbf{x}_0, t_0)}} + \sqrt{\frac{t_0^2}{4} + \frac{|\mathbf{x}_0 - G|^2}{v_m^2(\mathbf{x}_0, t_0)}}. \tag{4}$$

In the case where the velocity inside the earth depends only on the depth and the distance between the source and the receiver is small, the migration velocity $v_m(\mathbf{x}_0, t_0)$ is the RMS velocity, given by

$$v_m(t_0) = \sqrt{\frac{1}{t_0}\int_0^{t_0} v^2(z(\tau))d\tau}. \tag{5}$$

.

1.3 Emerging wave front

In this section, our aim is to justify the travel time approximation given by formula (4).

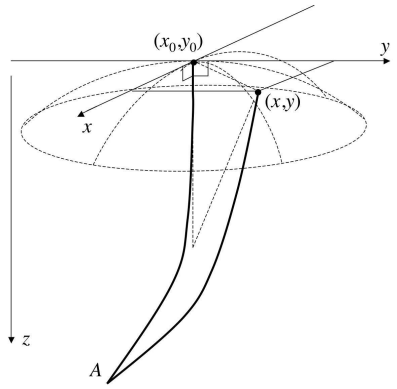


Figure 5: Emerging wave front.

Consider an emerging wave front from a point source A (Fig. 5) [Hubral and Krey, 1980]. Let the image ray arrive at the surface point (x_0, y_0) at time t_0 (here t_0 is the one-way travel time along the image ray). The travel time from A to the surface along some other ray close to the image ray, arriving at the surface point (x, y), is given by the Taylor expansion

$$t(x,y) = t_0 + \frac{1}{2}\Delta\mathbf{x}^T\mathbf{\Gamma}\Delta\mathbf{x} + O(\delta^3), \tag{6}$$

where $\Delta\mathbf{x} = \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$, $\mathbf{\Gamma}$ is the matrix of the second derivatives of $t(x, y)$ evaluated at the point (x_0, y_0) and $\delta = \sqrt{(x - x_0)^2 + (y - y_0)^2}$. From geometrical considerations, one can obtain [Hubral and Krey, 1980] a relation between the matrix $\mathbf{\Gamma}$ and the matrix $\mathbf{R}$ of the radii of curvature of the emerging wave front, namely

$$\mathbf{\Gamma}^{-1} = \mathbf{R}v(x_0, y_0), \tag{7}$$

where $v(x_0, y_0) = v(x = x_0, y = y_0, z = 0)$ is the velocity at the surface point (x_0, y_0).

In the case where sources and receivers are arranged along some straight line, seismic imaging becomes a 2D problem, and equation (6) can be simplified to

$$t(x) = t_0 + \frac{1}{2}(x - x_0)^2 t_{xx}(x = x_0) + O(\delta^3) = t_0 + \frac{(x - x_0)^2}{2Rv(x_0)} + O(\delta^3), \tag{8}$$

where $v(x_0) \equiv v(x = x_0, z = 0)$. By squaring both sides of equation (8) we get:

$$t^2(x) = t_0^2 + (x - x_0)^2 t_0 t_{xx}(x = x_0) + O(\delta^3) = t_0^2 + (x - x_0)^2 \frac{t_0}{Rv(x_0)} + O(\delta^3). \tag{9}$$

Suppose we want to compute the total travel time from a source S to the reflection point A and from A to a receiver G. Using equation (9) we obtain:

$$t(x_0, t_0, S, G) = t_{SA} + t_{AG} = \sqrt{t_0^2 + (S - x_0)^2 \frac{t_0}{Rv(x_0)}} + \sqrt{t_0^2 + (G - x_0)^2 \frac{t_0}{Rv(x_0)}} + O(\delta^3). \tag{10}$$

Comparing equations (10) and (4) we see that the travel time approximation given by formula (4) follows from the Taylor expansion in 2D. Moreover, the migration velocity and the radius of curvature of the emerging wave front are converted through the relation

$$t_0 v_m^2(x_0, t_0) = v(x_0) R(x_0, t_0). \tag{11}$$

On the other hand, in 3D the travel time approximation given by formula (4) is not a consequence of the Taylor expansion as it is in 2D. Instead, one can easily derive the following travel time formula from equations (6) and (7):

$$\begin{aligned} t(\mathbf{x}_0, t_0, S, G) = &\sqrt{t_0^2 + t_0 (S - \mathbf{x_0})^T [v(\mathbf{x_0}) \mathbf{R}(\mathbf{x_0}, t_0)]^{-1} (S - \mathbf{x_0})} \\ &+ \sqrt{t_0^2 + t_0 (G - \mathbf{x_0})^T [v(\mathbf{x_0}) \mathbf{R}(\mathbf{x_0}, t_0)]^{-1} (G - \mathbf{x_0})}. \end{aligned} \tag{12}$$

However note that if the velocity depends only on the depth, the matrix $\mathbf{R}$ is a multiple of the identity matrix, and hence formula (4) is the consequence of the Taylor expansion.

1.4 Dix inversion

Dix [Dix, 1955] established the first connection between the migration velocities and the seismic velocities for the case where the velocity depends only on the depth. He showed that the migration velocities are the RMS velocities if the distances between the sources and the receivers are small and came up with the following inversion method. Consider an earth model as in Fig.

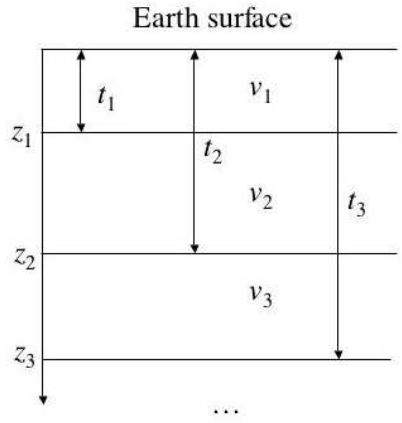


Figure 6: Dix inversion.

6. Let the layers be flat and horizontal, and the velocity be constant within each layer. We are given the RMS velocities V_i and the travel times t_i, $i = 1, 2, ..., n$, where V_i is the RMS velocity of the first i layers with respect to the time, and t_i is the two-way vertical travel time from the earth surface to the bottom of the i-th layer. Then the layer velocities (or as they are called in geophysics "interval velocities") v_i can be found successively from $i = 2$ to n:

$$v_i = \sqrt{\frac{V_i^2 t_i - V_{i-1}^2 t_{i-1}}{t_i - t_{i-1}}}. \tag{13}$$

The depths of the lower boundaries of the layers are:

$$z_i = z_{i-1} + v_i \frac{t_i - t_{i-1}}{2}. \tag{14}$$

Sometimes Dix inversion is applied to find the interval velocities from the migration velocities in the case where the velocity varies laterally. Then for the continuously changing velocity in 2D the Dix velocities are given by:

$$v(x_0, t_0) = \sqrt{\frac{\partial}{\partial t_0}\left(t_0 v_m^2(x_0, t_0)\right)}. \tag{15}$$

2 Forward modeling of the time migration velocities

In this section we derive our main theoretical result: the relation between the migration velocities and the true seismic velocities in 2D and between the matrix $\mathbf{\Gamma}$ in formula (6) in 3D.

2.1 Paraxial ray tracing

Consider a ray propagating in a 3D medium with a smooth velocity (Fig. 7). Call this ray *central* and attach a coordinate system (t, q_1, q_2) to it (see [Popov, 2002], [Červený, 2001]). Let t be the travel time along the central ray. Draw a plane perpendicular to the central ray at the point which it reaches at time t. Pick two mutually orthogonal directions in this plane and call them $\vec{e}_1$ and $\vec{e}_2$. Then the location of any point M in the space can be expressed as

$$\vec{r}_M = \vec{r}_0(t) + q_1\vec{e}_1 + q_2\vec{e}_2$$

for some t, q_1 and q_2, where $\vec{r}_0(t)$ gives the point reached by the central ray at time t. If M is close enough to the ray, its location can be described by (t, q_1, q_2) uniquely.

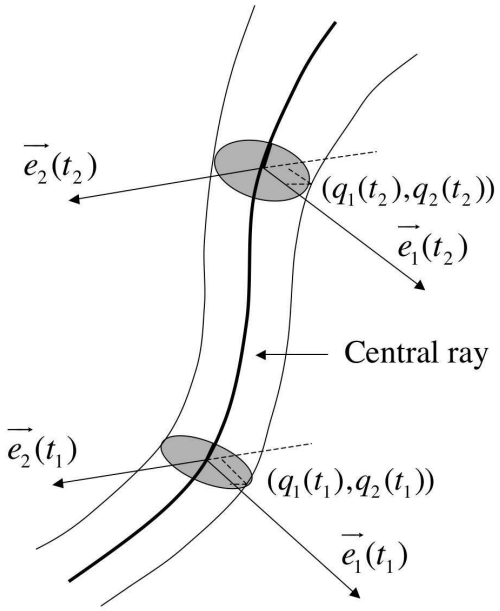


Figure 7: Paraxial ray tracing.

Suppose that the central ray is surrounded by a family of close rays and we want to write equations of those rays in terms of $q_1(t)$ and $q_2(t)$ (Fig. 7). In order to apply the Hamiltonian formalism, we need to introduce the generalized momentums p_1 and p_2 corresponding to the

generalized coordinates q_1 and q_2. We first note the fact that the central ray is a ray itself, and this imposes the following requirements on the evolution of $\vec{e}_1$ and $\vec{e}_2$:

$$\frac{d\vec{e}_1}{dt} = \left.\frac{\partial v(t,q_1,q_2)}{\partial q_1}\right|_{q_1=q_2=0}\vec{\tau}, \qquad \frac{d\vec{e}_2}{dt} = \left.\frac{\partial v(t,q_1,q_2)}{\partial q_2}\right|_{q_1=q_2=0}\vec{\tau},$$

where $\vec{\tau}$ is the unit tangent vector to the central ray ([Popov, 2002]). The ray equations in the Hamiltonian form are ([Popov, 2002], [Popov and Pšenčik, 1978],[Červený, 2001]):

$$\frac{d}{dt}\begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{0} & v_0^2 I_2 \\ -\frac{1}{v_0}\mathbf{V} & \mathbf{0} \end{pmatrix}\begin{pmatrix} q \\ p \end{pmatrix}. \tag{16}$$

Here v_0 is the velocity along the central ray, I_2 is the 2×2 identity matrix, and $\mathbf{V}$ is a 2×2 matrix of the second derivatives of the velocity:

$$\mathbf{V}_{ij} = \frac{\partial^2 v(t,q_1,q_2)}{\partial q_i \partial q_j}, \quad i,j = 1,2.$$

Suppose that the family of rays depends upon two parameters (α_1,α_2). There are two important cases:

- All rays start perpendicular to the same plane. Then (α_1,α_2) can be chosen to be the initial coordinates (x_0,y_0) of the rays at this plane. We will call such a family of rays *telescopic*.
- All rays start at the same point, but in different directions. Then (α_1,α_2) can be chosen to be the initial momentums $(p_1(0),p_2(0))$ of the rays. We will call such a family the *point source family*.

Consider the following 2×2 matrices ([Popov, 2002], [Červený, 2001]):

$$\mathbf{Q}_{ij} \equiv \frac{\partial q_i}{\partial \alpha_j}, \qquad \mathbf{P}_{ij} \equiv \frac{\partial p_i}{\partial \alpha_j}, \qquad i,j = 1,2. \tag{17}$$

The equations of time evolution for $\mathbf{Q}$ and $\mathbf{P}$ are the equations in variations for equation (16):

$$\frac{d}{dt}\begin{pmatrix} \mathbf{Q} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & v_0^2 I_2 \\ -\frac{1}{v_0}\mathbf{V} & \mathbf{0} \end{pmatrix}\begin{pmatrix} \mathbf{Q} \\ \mathbf{P} \end{pmatrix}. \tag{18}$$

The initial conditions for the telescopic family of rays are

$$\mathbf{Q}(0) = I_2, \qquad \mathbf{P}(0) = \mathbf{0}, \tag{19}$$

and for the point source family they are

$$\mathbf{Q}(0) = \mathbf{0}, \qquad \mathbf{P}(0) = \frac{1}{v_0(0)} I_2, \tag{20}$$

where $v_0(0)$ is the velocity at the source point. The absolute value of the determinant of the matrix $\mathbf{Q}$ has a nice geometrical sense ([Popov, 2002]):

$|\det \mathbf{Q}|$ is the geometrical spreading of the family of rays.

Let the central ray arrive orthogonal to some plane at a point (x_0,y_0). Consider the matrix $\mathbf{\Gamma}$ of the second derivatives of the travel times of the family of rays around the central ray, evaluated at the point (x_0,y_0). E.g., the central ray can be the image ray arriving to the earth surface. Then the matrix $\mathbf{\Gamma}$ is defined by formula 6) for the source point family of rays from the source point A as in Fig. 5. In [Popov, 2002], [Červený, 2001] it was shown that

$$\mathbf{\Gamma} = \mathbf{P}\mathbf{Q}^{-1} \tag{21}$$

and

$$\frac{d}{dt}\mathbf{\Gamma} = -v_0^2\mathbf{\Gamma}^2 - \frac{1}{v_0}\mathbf{V}. \tag{22}$$

For convenience, in the present work we will deal with the matrix $\mathbf{K} = \mathbf{\Gamma}^{-1}$, which is the matrix of radii of curvature of the wave front scaled by the velocity at the image ray, namely

$$\mathbf{K} = v_0\mathbf{R} = \mathbf{Q}\mathbf{P}^{-1}. \tag{23}$$

One can easily derive from equation (22) that the time evolution of $\mathbf{K}$ is given by:

$$\frac{d}{dt}\mathbf{K} = v_0^2 I_2 + \frac{1}{v_0}\mathbf{K}\mathbf{V}\mathbf{K}. \tag{24}$$

For the point source family of rays the initial conditions for the matrix $\mathbf{K}$ are:

$$\mathbf{K}(0) = \mathbf{0}. \tag{25}$$

2.2 Relation between the matrix K and the true seismic velocities in 3D

Theorem 1 *Let an image ray starting from a subsurface point* $\mathbf{x}$ *(Fig. 8) arrive at the earth surface point* $\mathbf{x}_0$ *at time* t_0*. Designate this ray to be central. Let the matrix* $\mathbf{K}(\mathbf{x}_0, t_0)$ *be evaluated at the surface for a point source family of rays around the image ray, starting at the same point* $\mathbf{x}$*. Suppose there is also a telescopic family of rays around the image ray starting perpendicular to the earth surface which we trace backwards w.r.t. the image ray for time* t_0 *and compute the matrices* $\mathbf{Q}$ *and* $\mathbf{P}$*. Let* $\mathbf{Q}(\mathbf{x}_0, t_0)$ *be the matrix* $\mathbf{Q}$ *for the telescopic family of rays evaluated at the time* t_0 *(i.e., at the subsurface point* $\mathbf{x}$*) in this backward tracing. Then*

$$\frac{\partial}{\partial t_0}\mathbf{K}(\mathbf{x}_0, t_0) = v^2(\mathbf{x}(\mathbf{x}_0, t_0))\left(\mathbf{Q}(\mathbf{x}_0, t_0)^T\mathbf{Q}(\mathbf{x}_0, t_0)\right)^{-1}. \tag{26}$$

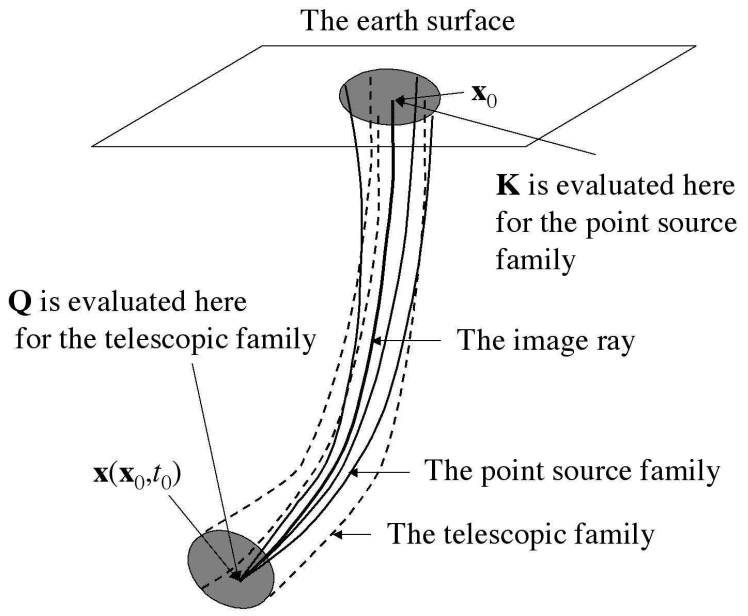


Figure 8: Illustration for Theorem 1.

Rewrite the travel time approximation (12) using the notation $\mathbf{K}$:

$$t(\mathbf{x}_0, t_0, S, G) = \sqrt{t_0^2 + t_0(S-\mathbf{x_0})^T[\mathbf{K}(\mathbf{x_0}, t_0)]^{-1}(S-\mathbf{x_0})} + \sqrt{t_0^2 + t_0(G-\mathbf{x_0})^T[\mathbf{K}(\mathbf{x_0}, t_0)]^{-1}(G-\mathbf{x_0})}. \tag{27}$$

The matrix $\mathbf{K}$ in formula (27) is a matrix of parameters depending on $\mathbf{x}_0$ and t_0, which can be estimated from the measurements. Theorem 1 provides a connection between the matrix $\mathbf{K}$ and the true seismic velocity at the subsurface point $\mathbf{x}$ reached by the image ray arriving at $\mathbf{x}_0$ and traced backwards for time t_0 (Fig. 8).

Proof Let an image ray arrive at the surface point $\mathbf{x}_0$ at time t_1. Fix a moment of time $t_0 < t_1$ and consider a point source family of rays starting at the subsurface point $\mathbf{x}(\mathbf{x}_0, t_0)$ which the image ray passes at time t_0. Introduce the following notations:

$$\mathbf{X} = \begin{pmatrix} \mathbf{Q} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \\ P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}, \qquad \mathbf{A}(t) = \begin{pmatrix} \mathbf{0} & v_0^2 I_2 \\ -\frac{1}{v_0}\mathbf{V} & \mathbf{0} \end{pmatrix}.$$

Let $\mathbf{X}_*$ be the 4×4 matrix of derivatives of $\mathbf{X}$ with respect to the initial conditions:

$$\mathbf{X}(t_0) = \begin{pmatrix} Q_{110} & Q_{120} \\ Q_{210} & Q_{220} \\ P_{110} & P_{120} \\ P_{210} & P_{220} \end{pmatrix}:$$

$$\mathbf{X}_* = \begin{pmatrix} \frac{\partial Q_{11}}{\partial Q_{110}} & \frac{\partial Q_{11}}{\partial Q_{210}} & \frac{\partial Q_{12}}{\partial Q_{120}} & \frac{\partial Q_{12}}{\partial Q_{220}} \\ \frac{\partial Q_{21}}{\partial Q_{110}} & \frac{\partial Q_{21}}{\partial Q_{210}} & \frac{\partial Q_{22}}{\partial Q_{120}} & \frac{\partial Q_{22}}{\partial Q_{220}} \\ \frac{\partial P_{11}}{\partial P_{110}} & \frac{\partial P_{11}}{\partial P_{210}} & \frac{\partial P_{12}}{\partial P_{120}} & \frac{\partial P_{12}}{\partial P_{220}} \\ \frac{\partial P_{21}}{\partial P_{110}} & \frac{\partial P_{21}}{\partial P_{210}} & \frac{\partial P_{22}}{\partial P_{120}} & \frac{\partial P_{22}}{\partial P_{220}} \end{pmatrix}.$$

Note that since each of the columns of $\mathbf{X}$ is a linear independent solution of equation (16) the derivatives not included into $\mathbf{X}_*$ are zeros. $\mathbf{X}(t)$ and $\mathbf{X}_*(t)$ are solutions of the following initial value problems:

$$\frac{d\mathbf{X}}{dt} = \mathbf{A}(t)\mathbf{X}, \qquad \mathbf{X}(t_0) = \frac{1}{v(t_0)} \begin{pmatrix} \mathbf{0} \\ I_2 \end{pmatrix}, \tag{28}$$

where $v(t_0) = v(\mathbf{x}(\mathbf{x}_0, t_0))$, and

$$\frac{d\mathbf{X}_*}{dt} = \mathbf{A}(t)\mathbf{X}_*, \qquad \mathbf{X}_*(t_0) = I_4. \tag{29}$$

Denote the solution of equation (29) by $\mathbf{B}(t_0; t_1)$ as it is done in [Červený, 2001]:

$$\mathbf{B}(t_0; t_1) = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{P}_1 & \mathbf{P}_2 \end{pmatrix},$$

where $\mathbf{Q}_i$, $\mathbf{P}_i$, $i = 1, 2$ are 2×2 matrices. $\begin{pmatrix} \mathbf{Q}_1 \\ \mathbf{P}_1 \end{pmatrix}$ satisfies the initial conditions corresponding to a telescopic point, and $\begin{pmatrix} \mathbf{Q}_2 \\ \mathbf{P}_2 \end{pmatrix}$ satisfies the initial conditions corresponding to a normalized point source. $\mathbf{B}(t_0, t_1)$ is called *the propagator matrix*. Then the solution of (28) is:

$$\mathbf{X}(t) = \frac{1}{v(t_0)} \begin{pmatrix} \mathbf{Q}_2 \\ \mathbf{P}_2 \end{pmatrix}. \tag{30}$$

Now turn to the matrix $\mathbf{K}$: $\mathbf{K}(t_0;t_1) = \mathbf{Q}(t_0;t_1)\mathbf{P}(t_0;t_1)^{-1} = \mathbf{Q}_2\mathbf{P}_2^{-1}$.

Shift the initial time t_0 by $-\Delta t$. Then, according to equation (28) at time t_0

$$\mathbf{Q}(t_0-\Delta t;t_0) = 0 + \Delta t v^2(t_0)\frac{1}{v(t_0)}I_2 + O((\Delta t)^2),$$

$$\mathbf{P}(t_0-\Delta t;t_0) = \frac{1}{v(t_0)}I_2 + O((\Delta t)^2).$$

Hence the change in the initial conditions for equation (28) is:

$$\Delta\mathbf{Q}_0 = v_0\Delta t I_2 + O((\Delta t)^2), \qquad \Delta\mathbf{P}_0 = \mathbf{0} + O((\Delta t)^2). \tag{31}$$

Then

$$\begin{aligned}\mathbf{K}(t_0-\Delta t;t_1) &= \mathbf{K}(t_0;t_1) + \sum_{i,j=1}^{2}\frac{\partial\mathbf{K}}{\partial Q_{ij0}}\Delta Q_{ij0} + \sum_{i,j=1}^{2}\frac{\partial\mathbf{K}}{\partial P_{ij0}}\Delta P_{ij0} + O((\Delta t)^2) \qquad (32)\\ &= \mathbf{K}(t_0;t_1) + \left(\frac{\partial\mathbf{K}}{\partial Q_{110}} + \frac{\partial\mathbf{K}}{\partial Q_{220}}\right)v(t_0)\Delta t + O((\Delta t)^2).\end{aligned}$$

Let us find the partial derivatives in the expression above:

$$\frac{\partial\mathbf{K}}{\partial Q_{ii0}} = \frac{\partial\mathbf{Q}}{\partial Q_{ii0}}\mathbf{P}^{-1} - \mathbf{Q}\mathbf{P}^{-1}\frac{\partial\mathbf{P}}{\partial Q_{ii0}}\mathbf{P}^{-1}, \quad i=1,2. \tag{33}$$

In terms of the entries of the matrix $\mathbf{B}(t_0;t_1)$

$$\frac{\partial\mathbf{K}}{\partial Q_{110}} + \frac{\partial\mathbf{K}}{\partial Q_{220}} = v_0(\mathbf{Q}_1\mathbf{P}_2^{-1} - \mathbf{Q}_2\mathbf{P}_2^{-1}\mathbf{P}_1\mathbf{P}_2^{-1}). \tag{34}$$

In [Červený, 2001] the symplectic property of the matrix $\mathbf{B}(t_0;t_1)$ was proved:

$$\mathbf{B}^T\mathbf{J}\mathbf{B} = \mathbf{J}, \tag{35}$$

where $\mathbf{J}$ is the 4×4 matrix

$$\mathbf{J} = \begin{pmatrix} \mathbf{0} & I_2 \\ -I_2 & \mathbf{0}\end{pmatrix}.$$

To simplify formula (34) we will use the following consequences of the symplectic property (35):

$$\mathbf{P}_2^T\mathbf{Q}_1 - \mathbf{Q}_2^T\mathbf{P}_1 = I_2, \qquad \mathbf{P}_2^T\mathbf{Q}_2 = \mathbf{Q}_2^T\mathbf{P}_2. \tag{36}$$

Then the matrix expression in equation (34) simplifies to:

$$\begin{aligned}&\mathbf{Q}_1\mathbf{P}_2^{-1} - \mathbf{Q}_2\mathbf{P}_2^{-1}\mathbf{P}_1\mathbf{P}_2^{-1} =\\ &(\mathbf{P}_2^T)^{-1}\mathbf{P}_2^T\mathbf{Q}_1\mathbf{P}_2^{-1} - (\mathbf{P}_2^T)^{-1}\mathbf{P}_2^T\mathbf{Q}_2\mathbf{P}_2^{-1}\mathbf{P}_1\mathbf{P}_2^{-1} =\\ &(\mathbf{P}_2^T)^{-1}(\mathbf{P}_2^T\mathbf{Q}_1 - \mathbf{P}_2^T\mathbf{Q}_2\mathbf{P}_2^{-1}\mathbf{P}_1)\mathbf{P}_2^{-1} =\\ &(\mathbf{P}_2^T)^{-1}(\mathbf{P}_2^T\mathbf{Q}_1 - \mathbf{Q}_2^T\mathbf{P}_2\mathbf{P}_2^{-1}\mathbf{P}_1)\mathbf{P}_2^{-1} =\\ &(\mathbf{P}_2^T)^{-1}(\mathbf{P}_2^T\mathbf{Q}_1 - \mathbf{Q}_2^T\mathbf{P}_1)\mathbf{P}_2^{-1} =\\ &(\mathbf{P}_2^T)^{-1}\mathbf{P}_2^{-1}. \qquad (37)\end{aligned}$$

Substituting Eqn. (37) to Eqn. (34) and then to Eqn. (32) we get:

$$\mathbf{K}(t_0-\Delta t;t_1) = \mathbf{K}(t_0;t_1) + \Delta t v^2(t_0)(\mathbf{P}_2^T)^{-1}\mathbf{P}_2^{-1} + O((\Delta t)^2). \tag{38}$$

Then the derivative of $\mathbf{K}$ with respect to the initial time is:

$$-\frac{\partial\mathbf{K}(t_0;t_1)}{\partial t_0} = v^2(t_0)(\mathbf{P}_2^T)^{-1}\mathbf{P}_2^{-1}. \tag{39}$$

In [Červený, 2001] the following reciprocity property was proved:

$$\mathbf{P}_2^T(\mathbf{x}_1,\mathbf{x}_2) = \mathbf{Q}_1(\mathbf{x}_2,\mathbf{x}_1), \tag{40}$$

where $\mathbf{x}_1$, $\mathbf{x}_2$ are the end points of the central ray. Applying it to equation (39) and taking the time reverse into account we obtain formula (26).

2.3 Relation between the Dix velocities and the true seismic velocities in 2D

In 2D the matrices **Q**, **P** and **K** become scalars which we denote by Q, P and K respectively. K is the radius of curvature of the wave front scaled by the velocity at the central ray: $K = vR$. The time evolution of Q, P and K are given by:

$$\frac{d}{dt}\begin{pmatrix} Q \\ P \end{pmatrix} = \begin{pmatrix} 0 & v_0^2 \\ -\frac{v_{qq}}{v_0} & 0 \end{pmatrix}\begin{pmatrix} Q \\ P \end{pmatrix}, \qquad \frac{dK}{dt} = v^2 + \frac{v_{qq}}{v}K^2. \tag{41}$$

In a similar way as it was done in 3D, it can be proven that

$$\frac{\partial}{\partial t_0}K(x_0,t_0) = \frac{v^2(x(x_0,t_0),z(x_0,t_0))}{Q^2(x_0,t_0)}. \tag{42}$$

Then taking into account the definition of the Dix velocity (15) and the relation (11) between the migration velocities and the radius of curvature of the emerging wave front we have the following:

Theorem 2 *Let an image ray arrive to the earth surface point x_0 at time t_0 from a subsurface point (x,z). Suppose there is a telescopic family of rays around the image ray starting perpendicular to the earth surface which we trace backwards w.r.t. the image ray for time t_0 and compute the quantities Q and P. Let $Q(x_0,t_0)$ be the quantity Q for the telescopic family of rays evaluated at the time t_0 (i.e., at the subsurface point (x,z)) in this backward tracing. Then the Dix velocity $v_{Dix}(x_0,t_0)$ is the ratio of the true seismic velocity $v(x,z)$ and the absolute value of $Q(x_0,t_0)$:*

$$v_{Dix}(x_0,t_0) = \frac{v(x(x_0,t_0),z(x_0,t_0))}{|Q(x_0,t_0)|}. \tag{43}$$

Note that here, t_0 is the one-way travel time along the image ray and that we denote the depth direction by z.

2.4 Statement of the inverse problem

We state an inverse problem in 2D: the below results may be extended to 3D after some work. Here, t_0 will denote the one-way travel time along the image ray.

Suppose there is an image ray arriving at each surface point x_0, $x_{min} \le x_0 \le x_{max}$. For any $0 \le t_0 \le t_{max}$, trace the image ray backward for time t_0 together with a small telescopic family of rays. Let the image ray being traced backward reach a subsurface point (x,z) at time t_0. Denote by $v(x_0,t_0)$ the velocity at the point (x,z), and by $Q(x_0,t_0)$ the quantity Q for the corresponding telescopic family at the point (x,z). We are given $v_{Dix}(x_0,t_0) = \frac{v(x(x_0,t_0),z(x_0,t_0))}{|Q(x_0,t_0)|} \equiv f(x_0,t_0)$, $x_{min} \le x_0 \le x_{max}$, $0 \le t_0 \le t_{max}$. We need to find $v(x,z)$, the velocity inside the domain covered with the image rays arriving to the surface in the interval $[x_{min}, x_{max}]$.

The first question is whether this problem is well-posed. In the next sections, we will show that both the direct problem (given $v(x,z)$ find $f(x_0,t_0)$) and the inverse problem (given $f(x_0,t_0)$ find $v(x,z)$) are ill-posed. We will use the notation $f(x_0,t_0) \equiv \frac{v(x(x_0,t_0),z(x_0,t_0))}{|Q(x_0,t_0)|}$ rather than $v_{Dix}(x_0,t_0)$ to emphasize that f is computed as the ratio $v/|Q|$ rather than from the optimal migration velocities.

2.5 Ill-posedness of the direct problem

Direct Problem: Given $v(x,z)$, $x_{min} \le x \le x_{max}$, $x_{min} < 0$, $x_{max} > 0$, $z \ge 0$ and t_{max} find $f(x_0,t_0) = \frac{v(x_0,t_0)}{|Q(x_0,t_0)|}$, $x_{min} \le x_0 \le x_{max}$, $0 \le t_0 \le t_{max}$.

We shall show that small changes in $v(x,z)$ can lead to large changes in $f(x_0,t_0)$. Take $v(x,z)=1$ and $\tilde{v}(x,z)=1+a\cos(kx)$, $-1\le x\le 1$. Then

$$||\tilde{v}-v||_\infty = a.$$

Obviously, $f(x_0,t_0)=1$ for $v(x,z)=1$. Compute $\tilde{f}(x_0,t_0)$ for $\tilde{v}(x,z)$ at $x_0=0$. As the image ray arriving at $x_0=0$ is straight, we have that

$$v_{qq}=v_{xx}(x=0)=-ak^2.$$

Then we have:

$$\frac{dQ}{dt_0}=(1+a^2)P, \qquad \frac{dP}{dt_0}=\frac{ak^2}{1+a}Q, \qquad Q(0)=1, \qquad P(0)=0.$$

Therefore,

$$\frac{d^2Q}{dt_0^2}=\frac{ak^2(1+a^2)}{1+a}Q, \quad Q(0)=1, \quad \frac{dQ}{dt_0}=0.$$

Hence,

$$Q(t_0)=\cosh\omega t_0, \qquad \omega=\sqrt{\frac{ak^2(1+a^2)}{1+a}}.$$

Pick $k=\frac{1}{a}$ and let a tend to zero. Then $\frac{1}{\sqrt{2a}}<\omega<\sqrt{\frac{2}{a}}$, and

$$f(x_0=0,t_0)=\frac{1}{\cosh\omega t_0}<\frac{1}{\cosh\frac{t_0}{\sqrt{2a}}}.$$

Hence

$$||\tilde{f}(x_0,t_0)-f(x_0,t_0)||_\infty>1-\frac{1}{\cosh\frac{t_{max}}{\sqrt{2a}}}>\frac{1}{2}$$

for a small enough. Thus, we have shown that arbitrarily small changes in the velocity $v(x,y)$ may lead to significant changes in $f(x_0,t_0)$, i.e., *the direct problem is physically unstable in the max norm.*

2.6 Ill-posedness of the inverse problem

Inverse Problem: Given $f(x_0,t_0)=\frac{v(x_0,t_0)}{|Q(x_0,t_0)|}$, $x_{min}\le x_0\le x_{max}$, $0\le t_0\le t_{max}$, find $v(x,z)$, the velocity inside the domain covered with the image rays arriving to the surface in the interval $[x_{min},x_{max}]$.

Here we shall prove that the corresponding discrete problem is ill-posed: Given $f(x_{0i},t_k)$, $i=0,1,...,n-1$, $k=0,1,...,p-1$, $x_{0i}=x_{min}+i\Delta x$, $t_k=k\Delta t$, where $\Delta x=(x_{max}-x_{min})/(n-1)$, $\Delta t=t_{max}/(p-1)$ respectively, find $v(x_i,z_j)$, $i=0,1,...,n-1$, $j=0,1,...,m-1$.

Let $x_{min}=-L$ and $x_{max}=L$ and n be odd so that $x=0$ is one of the grid lines. Suppose we are given the following two discrete arrays: (1) $f(x_{0i},t_k)=1$ and (2) $\tilde{f}(x_{0i},t_k)=1$ if $x_{0i}\neq 0$ and $\tilde{f}(x_{0i},t_k)=b>1$ if $x_{0i}=0$. Then

$$||f(x_{0i},t_k)-\tilde{f}(x_{0i},t_k)||_\infty=b-1. \tag{44}$$

For $f(x_{0i},t_k)=1$ $v(x,y)=1$. Let us find a velocity $\tilde{v}(x,z)$ such that the exact values of $\tilde{f}$ for it coincides with $\tilde{f}(x_{0i},t_k)$ on the mesh. Let the mesh step in x_0 be Δx. We will look for $\tilde{v}(x,z)$ in the following form: pick $0<\alpha\le\Delta x$ and set $\tilde{v}(x,z)=1$ if $|x|\ge\alpha$, and

$$\tilde{v}(x,z)=v(x,t_0(z))=1+(\tilde{v}(0,t_0)-1)\exp\left(1-\frac{1}{1-\left(\frac{x}{\alpha}\right)^2}\right)$$

if $|x| < \alpha$. Here $v(0, t_0)$ is to be found. Note that

$$\tilde{v}_{xx}(0, t_0) = -\frac{2}{\alpha^2}(\tilde{v}(0, t_0) - 1). \tag{45}$$

Since $\tilde{f}(0, t_k) = \frac{\tilde{v}(0,t_0)}{Q(0,t_0)} = b$,

$$Q(0, t_0) = \frac{\tilde{v}(0, t_0)}{b}. \tag{46}$$

Due to the symmetry of our $v(x, z)$, the ray starting at $x_0 = 0$ perpendicular to the surface is straight. Let us write the IVP for Q and P for this ray:

$$\begin{aligned} \frac{dQ}{dt_0} &= v^2 P, \qquad Q(T = 0) = 1, \\ \frac{dP}{dt_0} &= -\frac{v_{xx}}{v} Q, \qquad P(T = 0) = 0. \end{aligned}$$

Here $v(t_0) \equiv \tilde{v}(0, t_0)$. Taking into account relation (46) and using Eqn. (45) we get:

$$\begin{aligned} \frac{dv}{dt_0} &= bv^2 P, \qquad v(t_0 = 0) = b, \\ \frac{dP}{dt_0} &= \frac{2}{\alpha^2 b}(v - 1), \qquad P(t_0 = 0) = 0 \end{aligned} \tag{47}$$

Along with IVP (47) consider the following IVP:

$$\begin{aligned} \frac{dw}{dt_0} &= bu, \qquad w(t_0 = 0) = b, \\ \frac{du}{dt_0} &= \frac{2}{\alpha^2 b}(w - 1), \qquad u(t_0 = 0) = 0. \end{aligned} \tag{48}$$

Solving IVP (48) we find:

$$w(t_0) = 1 + (b - 1) \cosh\left(\frac{t_0\sqrt{2}}{\alpha}\right).$$

Then by a variant of a comparison theorem, on the interval $[0, T_*)$ where the solution to IVP (47) exists, $v(t_0) > w(t_0)$. Hence, $\tilde{v}(0, t_0)$ either blows up, or reaches its maximum at t_{max}. Hence we conclude that

$$||v(x, z) - \tilde{v}(x, z)||_\infty > (b - 1) \cosh\left(\frac{t_{max}\sqrt{2}}{\alpha}\right). \tag{49}$$

Comparing formulae (44) and (49) we see that for any b we can pick $\alpha = \min\{\Delta x, \frac{(b-1)t_{max}\sqrt{2}}{3}\}$ and hence make the left-hand side of Eqn. (49) greater than 1. Thus we have shown that *the inverse problem is numerically unstable in the max norm.*

2.7 Eulerian formulation of the inverse problem

The inverse problem stated in Section 2.4 can be formulated in a different, Eulerian way. Consider the mapping between the Cartesian coordinates (x, z) and the time migration coordinates (x_0, t_0). The functions $x_0(x, z)$ and $t_0(x, z)$ satisfy the following system of equations:

$$|\nabla x_0|^2 = \left(\frac{\partial x_0}{\partial x}\right)^2 + \left(\frac{\partial x_0}{\partial z}\right)^2 = \frac{1}{Q^2(x, z)}, \tag{50}$$

$$\nabla x_0 \cdot \nabla t_0 = \frac{\partial x_0}{\partial x}\frac{\partial t_0}{\partial x} + \frac{\partial x_0}{\partial z}\frac{\partial t_0}{\partial z} = 0, \tag{51}$$

$$|\nabla t_0|^2 = \left(\frac{\partial t_0}{\partial x}\right)^2 + \left(\frac{\partial t_0}{\partial z}\right)^2 = \frac{1}{v^2(x, z)}. \tag{52}$$

Equation (50) follows from the definition of Q. Equation (51) indicates that the curves t_0=const are orthogonal to the image rays, and will be derived in Section 3.1.1 below. Equation (52) is the Eikonal equation.

The input data are

$$v^2_{Dix}(x_0,t_0)=\frac{v^2(x(x_0,t_0),z(x_0,t_0))}{Q^2(x(x_0,t_0),z(x_0,t_0))}. \tag{53}$$

The boundary conditions are:

$$x_0(x,0)=x, \quad t_0(x,0)=0, \quad Q(x,0)=1, \quad v(x,0)=v_{Dix}(x_0=x,t_0=0). \tag{54}$$

3 Numerical algorithms

In this section we will propose three numerical algorithms. We will start with an efficient time-to-depth conversion algorithm. The input for it is $v(x_0,t_0)\equiv v(x(x_0,t_0),z(x_0,t_0))$. The output is $v(x,z)$, $x_0(x,z)$ and $t_0(x,z)$. This algorithm is an essential part of the other two algorithms which produce $v(x,z)$ from $v_{Dix}(x_0,t_0)$. The first of these two, based on the ray tracing approach, creates $v(x_0,t_0)$, the input for the time-to-depth algorithm. The second, based on the level set approach, uses it as a part of its time cycle. Also, if nothing else is available, Dix velocities can be used as the input for our time-to-depth conversion. The main advantage of this time-to-depth conversion algorithm is that it is very fast and robust.

3.1 Efficient time-to-depth conversion algorithm

In this section we will use notation T for t_0 to be consistent with the notations in the Eikonal equation (2). Also, we will deal with the reciprocal of the velocity $s(x,z)$ called *slowness* for convenience.

3.1.1 Eulerian formulation of the boundary value problem

Let (x,z) be a subsurface point (Fig. 9). Let $s(x,z)$ be the slowness at the point (x,z). Let the image ray from (x,z) reach the surface at some point x_0 and let T be the one-way travel time from (x,z) to the surface point x_0.

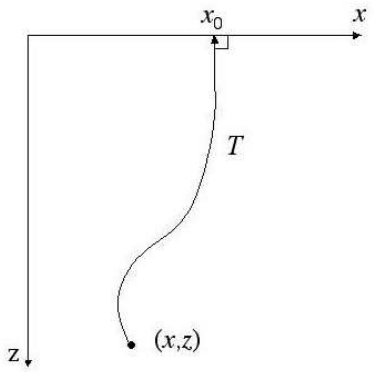


Figure 9: Section 3.1.1. Relation between (x,z), x_0 and T.

Let $x_{min}\le x_0\le x_{max}$, $0\le T\le T_{max}$, $x_{min}\le x\le x_{max}$, $0\le z\le z_{max}$. Given $s(x_0,T)$, our goal is to find $s(x,z)$, $x_0(x,z)$ and $T(x,z)$, i.e., the slowness at each subsurface point (x,z), the

escape location of the image ray from each subsurface point (x, z), and the one-way travel time along each image ray. Thus, the input for this algorithm is given in the time domain (x_0, T), and the desired output is in the depth domain (x, z).

The functions $x_0(x, z)$ and $T(x, z)$ are well-defined in the case if the image rays do not intersect inside the domain in hand. If the image rays intersect, the algorithm will follow the first arrivals to the surface.

The functions $s(x_0, T)$, $x_0(x, z)$ and $T(x, z)$ are related according to the following system of PDE's:

$$|\nabla T|^2 = s^2(x_0, T) \equiv s(x_0(x, z), T(x, z)), \tag{55}$$

$$\nabla T \cdot \nabla x_0 = 0. \tag{56}$$

Equation (55) is the Eikonal equation with an unknown right-hand side. Equation (56) gives a connection between x_0 and T, and indicates that the curves T=const are orthogonal to the image rays. We may derive this relation as follows:

We first note that the escape location x_0 is constant along each image ray. Hence the time derivative of x_0 along each image ray must be zero:

$$\frac{dx_0}{dT} = \frac{\partial x_0}{\partial x}\frac{dx}{dT} + \frac{\partial x_0}{\partial z}\frac{dz}{dT} = 0. \tag{57}$$

Writing the equations of the phase trajectories for the Hamiltonian

$$H = \frac{1}{2}|\nabla T|^2 - \frac{1}{2}s^2(x, z) = 0$$

given by the Eikonal equation, we have that

$$\frac{dx}{dT} = \frac{\partial T}{\partial x}\frac{1}{s^2}, \qquad \frac{dz}{dT} = \frac{\partial T}{\partial z}\frac{1}{s^2}.$$

Substituting this into equation (57) we get:

$$\frac{\partial x_0}{\partial x}\frac{dx}{dT} + \frac{\partial x_0}{\partial z}\frac{dz}{dT} = \frac{1}{s^2}\nabla x_0 \cdot \nabla T = 0.$$

Hence, $\nabla x_0 \cdot \nabla T = 0$ as desired.

We also have boundary conditions for the system (56):

$$x_0(x, 0) = x, \quad T(x, 0) = 0, \quad s(x, 0) = s(x_0 = x, T = 0). \tag{58}$$

3.1.2 Numerical algorithm

The motivation and the main building block or this algorithm is Sethian's Fast Marching Method [Sethian, 1996] designed for solving a boundary value problem for the Eikonal equation with known right-hand-side. This method is a Dijkstra-type method, in that it systematically advances the solution to the desired equation from known values to unknown values without iteration. Dijkstra's method, first developed in the context of computing a shortest path on a network, computes the solution in order $N \log N$, where N is the total number of points in the domain. The first extension of this approach to an Eikonal equation is due to Tsitsiklis [Tsitsiklis, 1995], who obtains a control-theoretic discretization of the Eikonal equation, which then leads to a causality relationship based on the optimality criterion. Tsitsiklis' algorithm evolved from studying isotropic min-time optimal trajectory problems, and involves solving a minimization problem to update the solution. A more recent, finite difference approach, based again on Dijkstra-like ordering and updating, was developed by Sethian [Sethian, 1996, Sethian, 1999A] for solving the Eikonal equation. Sethian's Fast Marching Method evolved from studying isotropic front propagation problems, and involves an upwind

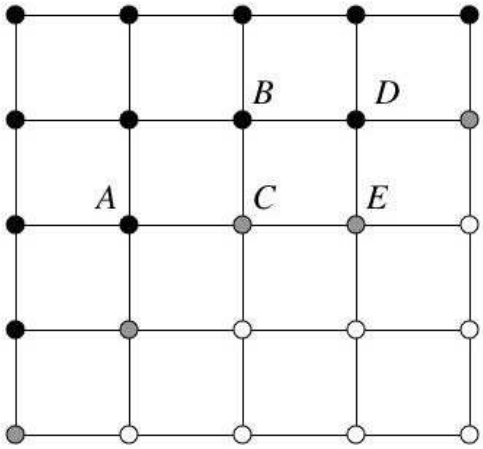


Figure 10: Fast Marching Method. Black, grey and white dots represent "Accepted", "Considered" and "Unknown" points respectively.

finite difference formulation to update the solution. Both Tsitskilis' method and the Fast Marching Method start with a particular (and different) coupled discretization and each shows that the resulting system can be decoupled through a causality property. In the particular case of a first order scheme on a square mesh, the resulting quadratic update equation at each grid point is the same for both methods. We refer the reader to these references for details on ordered upwind methods for Eikonal equations, as well as [Sethian and Vladimirsky, 2003] for a detailed discussion about the similarities and differences between the two techniques. More recently, Sethian and Vladimirsky have built versions a class of Ordered Upwind Methods, based on Dijkstra-like methodology, for solving the more general class of optimal control problems in which the speed/cost function depends on both position and direction, which leads to a convex Hamilton-Jacobi equation. See [Sethian and Vladimirsky, 2003] for details.

We now discuss the Fast Marching Method in more detail, since it will serve as a building block to our algorithm. In order to follow the physical propagation of information, an upwind scheme is used, and the solution is computed at points in order of increase of T. In order to achieve it, the points are divided into "Accepted", where T is computed and no longer can be updated and can be used for estimation of T at its neighbors; "Considered", where T is computed but may be updated in future and cannot be used for estimation of T at other points; "Unknown", where no value of T has been computed yet. At each time step a "Considered" point with the smallest value of T, determined by the heap sort, becomes "Accepted". Sethian used this approach to compute the solution of the Eikonal equation with known right-hand-side in a variety of settings including semiconductor processing, image segmentation, seismic wave propagation and robotic navigation: for details, see [Sethian, 1996, Sethian, 1999A, Sethian, 1999B].

In our case the principal difference is that the right-hand side of the Eikonal equation is unknown. In result we do not know where the information comes from. This creates an issue which we will discuss in Section 3.1.3 below, however, we first outline the time-to-depth conversion algorithm.

Let us discretize and solve the system (55), (56) with boundary conditions (58). The input for the numerical algorithm is the matrix $s(x_{0i}, T_k)$, $i = 0, 1, ..., n-1$, $k = 0, 1, ..., p-1$. Denote the mesh steps in x_0 and T by hx and ΔT respectively. The mesh steps in x and z are hx and hz respectively. We define $s(x_0, T)$ beyond the mesh points by the bilinear interpolation. The output of the numerical algorithm are the matrices $s(x_i, z_j)$, $x_0(x_i, z_j)$ and $T(x_i, z_j)$, $i = 0, 1, ..., n-1$, $j = 0, 1, ..., m-1$. The algorithm is the following:

1. Mark the boundary (surface) points ($x_i = x_{0i}, z = 0$) as "Accepted". Set $s(x_i, z_0 =$

$0) = s(x_0 = x_i, T = 0)$, $x_0(x, z = 0) = x_0$, $T(x, z = 0) = 0$ according to the boundary conditions. Mark the rest of the mesh points (x_i, z_j) as "Unknown".

2. Mark the "Unknown"points adjacent to the "Accepted" points as "Considered". We call two points adjacent (or nearest neighbors) if they are separated by one edge.
3. Compute or update tentative values of $s(x_i, z_j)$, $x_0(x_i, z_j)$ and $T(x_i, z_j)$ at the "Considered" points.

   (a) If a "Considered" point E has only one "Accepted" nearest neighbor D as in Fig. 10, then the values at E are found from the 1-point-update system:

$$\begin{aligned} &x_0(E) = x_0(D), \\ &T(E) - T(D) = hs(x_0(D), T(E)), \\ &s(E) = s(x_0(E), T(E)), \\ &T(E) > T(D). \end{aligned} \tag{59}$$

   Here H is either hx or hz depending on the arrangement of E and D.

   (b) If a "Considered" point has only two "Accepted" nearest neighbors and they are located so that it lies linearly between them, then we compute triplets of tentative values of s, x_0 and T for each of the two "Accepted" points and the "Considered" point from system (59), and then choose the triplet with the smallest value of T.

   (c) If a "Considered" point C has only two "Accepted" neighbors A and B not lying on the same grid line, as in Fig. 10, then the tentative values at C are found from the 2-point-update system:

$$\begin{aligned} &\frac{(T(C) - T(A))^2}{hx^2} + \frac{(T(C) - T(B))^2}{hz^2} = s^2(x_0(C), T(C)), \\ &\frac{(T(C) - T(A))(x_0(C) - x_0(A))}{hx^2} + \frac{(T(C) - T(B))(x_0(C) - x_0(B))}{hz^2} = 0, \\ &s(C) = s(x_0(C), T(C)), \\ &x_0(A) \le x_0(C) \le x_0(B), \\ &T(C) \ge \max\{T(A), T(B)\}. \end{aligned} \tag{60}$$

   We solve the first two equations in system (60) using a Newton solver.

   (d) If a "Considered" point has three or more "Accepted" nearest neighbors then we compute a triplet of tentative values for each possible couple of "Accepted" points forming a right triangle together with the "Considered" point such that the "Considered" point lies at its right angle, and choose the triplet with the smallest value of T.

4. Find a "Considered" point with the smallest tentative value of "T" and mark it as "Accepted". We use a heap sort to keep track of the tentative T values.
5. If the set of "Considered" points is not empty, return to 2.

3.1.3 Causality

At $T = 0$ the wave front is a segment of the straight line from $(x_0, 0)$ to $(x_{n-1}, 0)$. In order to propagate it correctly, we must compute the points in order of increase of T as given in Sethian's Fast Marching Method.

In the above our update principle, the 1-point update (59) *artificially puts point E on the image ray passing through D (Fig. 10) prescribing $x_0(E) = x_0(D)$, while the 2-point-update looks for the correct image ray (the correct value of x_0).*

At the moment when some "Unknown" point becomes "Considered", it has only one "Accepted" nearest neighbor. Therefore the tentative values at it are found from the 1-point update system (59). Then, if it does not become "Accepted" by that time, it gets two "Accepted" neighbors lying on different grid lines. Then the values at it are found from the 2-point-update system (60). We emphasize that *we design our algorithm so that the 2-point-update values replace the 1-point-update values whenever it is possible independently of whether the new tentative value of T is smaller or larger.* Note that in the Fast Marching Method, the 2-point-update value never exceeds the 1-point-update value due to the fact that the slowness is known at each point. In our formulae (59) and (60) for 1- and 2-point-update respectively the slowness s in the right-hand side depends on T. Because of this, we cannot eliminate the situation where the value of T given by 1-point-update is smaller than the one given by 2-point-update. Such a situation is dangerous because the 1-point-update's setting $x_0(E) = x_0(D)$ is correct only if the true velocity (slowness) at E is larger (smaller) then at both of its nearest neighbors in the direction perpendicular to the segment DE (Fig. 10). Thus, in the case where this setting is incorrect, the 1-point-update values must be replaced by 2-point update values before the point gets "Accepted" in order to propagate the front in order of increase of the true values of T. The question is whether we can guarantee it.

We found examples where indeed a smaller tentative value of T from the 1-point-update was replaced by a larger one from the 2-point-update in a small subset of points. However, numerous numerical experiments showed that such points disappear as we refine the mesh of the input data $s(x_{0i}, T_k)$, $i = 0, 1, ..., n-1$, $k = 0, 1, ..., p-1$. Moreover, we did not find any example where the points with 1-point-update values got accepted when they should not be. Thus, although the upwind principle may be violated in theory, we have not found any such example in practice.

3.1.4 Boundary effects

We have input data in the rectangular time domain (x_0, T), and we look for the output in the rectangular depth domain (x, z). We will call the image rays arriving at the end points of the "earth surface" segment of the domain the *boundary image rays.* There are three possible behavior of a boundary image ray:
1) the ray is straight, i.e, lies strictly on the boundary of the domain;
2) the ray escapes from the domain;
3) the ray enters the interior of the domain.
If the boundary image ray is either straight or escapes from the domain then our numerical algorithm computes the values at the boundary mesh points correctly, as the physical domain of dependence of each boundary point lies inside the numerical domain of dependence in these cases. If the boundary ray enters the interior of the domain, then the values at the boundary points are computed by 1-point-updates. The physical domain of dependence for each boundary point lies outside the domain, and hence, cannot be inside the numerical domain of dependence. In this case, our algorithm does not converge in the cone of influence of the boundary points.

3.1.5 Synthetic data example

As a first example, we took the velocity field

$$v(x, z) = 1 + \frac{1}{2} \cos \frac{\pi x}{3} \sin \frac{\pi z}{3},$$

and generated the input data $v(x_0, T)$, $0 \le x_0 \le 12$, $0 \le T \le 5$ for our time-to-depth conversion algorithm on a 200×200 $nx_0 \times nT$ mesh by shooting characteristics. Then we applied the algorithm to these data and computed the velocity $v(x, z)$ on the 200×400 $nx \times nz$ mesh. The

results are presented in Fig. 11-12. The exact velocity is shown in Fig. 11(a); the input data are shown in Fig. 11(b). The velocity found by the algorithm is shown in Fig. 11(c). The relative error, i.e., $(v_{found} - v_{exact})/v_{exact}$ is shown in Fig. 11(d). The maximal relative error is less than 5 percent and is achieved at the points where the image rays collapse. The image rays computed for the exact velocities are shown in Fig. 12. Note that 1) the image rays severely bend, diverge and intersect, and 2) the boundary image rays are straight, which eliminates the errors from the boundary effects.

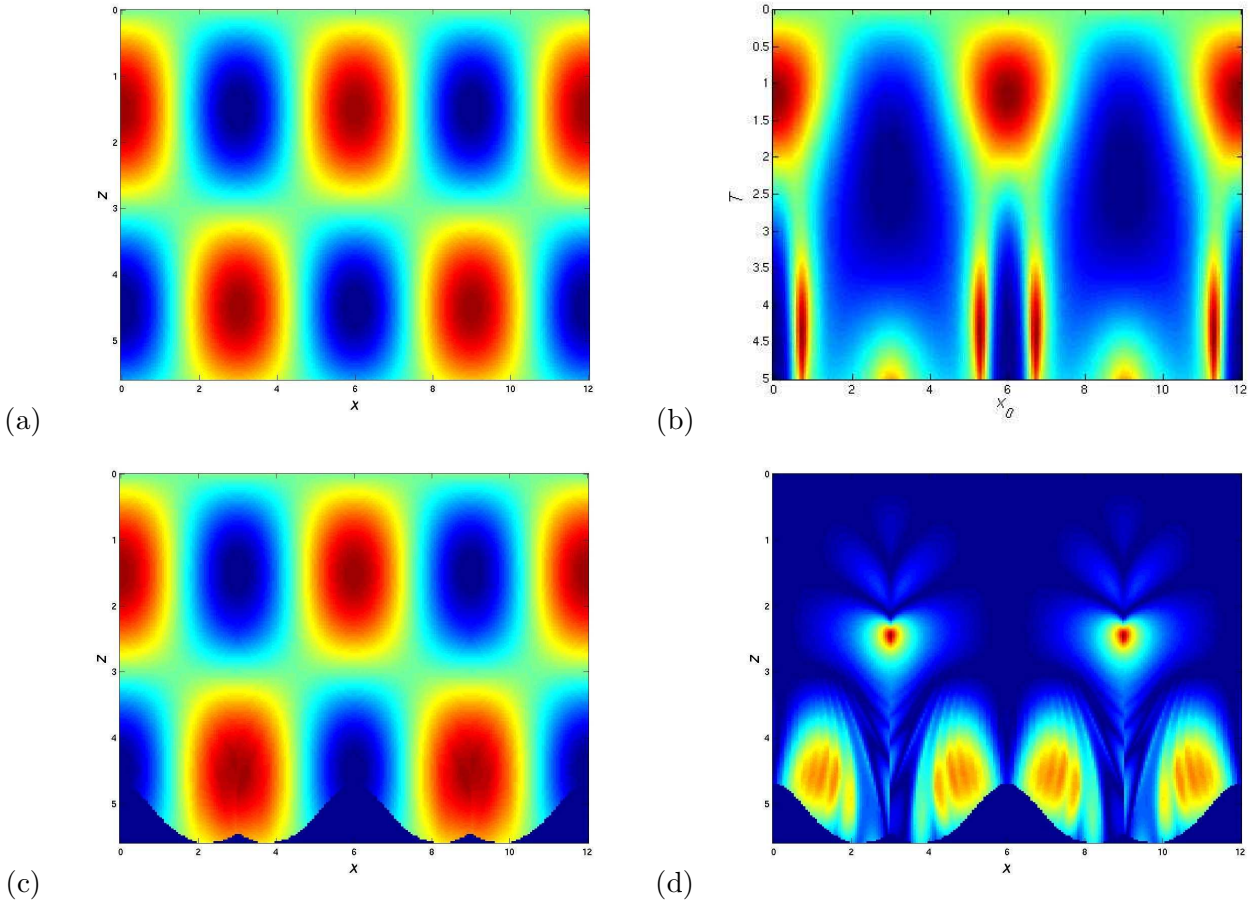


Figure 11: (a): the exact velocity $v(x,z) = 1 + \frac{1}{2}\cos\frac{\pi x}{3}\sin\frac{\pi z}{3}$; (b): the input data $v(x_0,T)$; (c): the found velocity $v(x,z)$; (d) the relative error: its maximus is less than 5 percent.

3.2 Algorithms producing the seismic velocities from the migration velocities

The algorithm introduced in Section 3.1.2 requires the velocities $v(x_0,T)$ as the input; one can use the Dix velocities $v_{Dix}(x_0,T)$ as input. However, Dix velocities are obtained with the assumption that the subsurface structures are horizontal and the velocity depends only on the depth. Theorem 2 gave the relation between the Dix velocities and the true seismic velocities. In this section, we introduce two algorithms which try to construct the true seismic velocities from the Dix velocities and use the algorithm in Section 3.1.2 as their essential part. To be sure, we have just proven that this problem is ill-posed; nonetheless we can develop algorithms

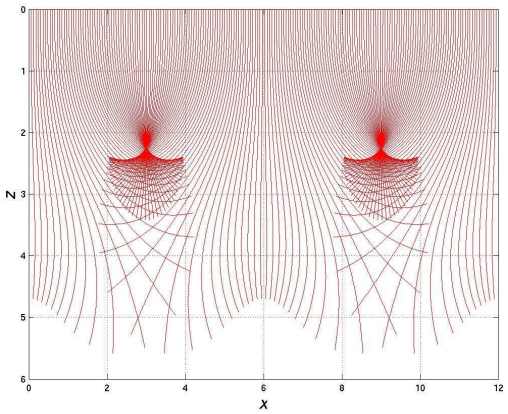


Figure 12: The image rays computed for the exact velocity.

which attempt the smoothed reconstruction. The first one is based on the ray tracing approach, and the second one is based on the level set approach. This is a worthwhile endeavor: our numerical examples below demonstrate that the Dix velocities and the true seismic velocities may significantly differ in the case of lateral velocity variation.

3.2.1 Ray tracing approach

The ray tracing algorithm consists of three steps.
Step 1. Find the image rays.
Step 2. Compute the geometrical spreading $|Q| = |\frac{dl}{dx_0}|$ on the image rays and find $v(x_{0i}, T_k)$. l is the length of the front.
Step 3. Apply the time-to-depth conversion algorithm from Section 3.1.2 to get $v(x_i, z_j)$, $x_0(x_i, z_j)$ and $t_0(x_i, z_j)$ from $v(x_{0i}, T_k)$.

Let us describe Step 1 in more details. The boundary conditions are $v(x_{0i}, T = 0) = f(x_{0i}, T = 0)$, $Q(x_{0i}, T = 0) = 1$, $P(x_{0i}, T = 0) = 0$. The ray tracing system for the i-th ray is the following:

$$\begin{aligned} x_T &= v\sin\theta, \qquad x(0) = x_{0i}, \\ z_T &= v\cos\theta, \qquad z(0) = 0, \\ \theta_T &= -v_n = -v_l, \qquad \theta(0) = 0, \\ Q_T &= v^2 P, \qquad , Q(0) = 1, \\ P_T &= -\frac{v_{nn}}{v} Q = -\left(\frac{v_{ll}}{v} + \frac{\kappa v_T}{v^2}\right) Q, \qquad P(0) = 0. \end{aligned} \tag{61}$$

Here $v_n = v_x\cos\theta - v_z\sin\theta$ is the derivative of v in the direction normal to the ray (note: $v_n \equiv v_q$); v_l is the derivative of v with respect to the arc length of the front; $v_{nn} = v_{xx}\cos^2\theta - 2v_{xz}\cos\theta\sin\theta + v_{zz}\sin^2\theta$ is the second derivative of v in the direction normal to the ray ($v_{nn} \equiv v_{qq}$); v_{ll} is the second derivative of v with respect to the arc length of the front; κ is the curvature of the front. Adalsteinsson and Sethian [Adalsteinsson and Sethian, 2002] derived the following relation between the second derivative of some physical quantity with respect to the arc length of the front and its second derivative along the line tangent to the front:

$$g_{ll} = g_{zz} - (g_x n_x + g_z n_z)\kappa,$$

where n is the unit vector normal to the front. Replacing g with v and noticing that

$$v_x n_x + v_z n_z = v_\tau = \frac{v_T}{v}$$

is the derivative of v with respect to the arc length of the ray, we get the last equation in (61).

We solve system (61) for all of the rays simultaneously by the forward Euler method as follows.

For $k = 0$ to $k = p - 1$ do:

1. Find the least squares polynomials for the set of points $(l_i, v_i(T_k))$ where l_i is the arc length of the front between ray 0 and ray i at the time T_k, and $v_i(T_k)$ is the value of the velocity on the i-th ray at time T_k. Evaluate $v_l(T_k)$ and $v_{ll}(T_k)$ taking the first and the second derivatives of this polynomial. Moreover, replace the values of the velocity $v_i(T_k)$ by the values of this polynomial. Evaluate the curvature $\kappa(T_k)$ as follows. Find the least squares polynomials for the sets of points $(i, x_i(T_k))$ and $(i, z_i(T_k))$ where i is the index of the ray, and x_i and z_i are the x- and z-coordinates of the i-th ray at time T_k. Take the first and the second derivatives of these polynomials p_x and p_z and find

$$\kappa = \frac{p'_x p''_z - p'_z p''_x}{(p'^2_x + p'^2_z)^{3/2}}.$$

   Approximate $v_T(T_k)$ by

$$v_t(T_k) = \frac{v(T_k) - v(T_{k-1})}{\Delta T}$$

   if $k > 0$, and we set $v_T(T_0 = 0) = 0$, since the curvature of the front is zero at $T = 0$.

2. Perform one forward Euler step for each of the rays.
3. For each of the rays find $v_i(T_{k+1}) = f_i(T_{k+1})Q(T_{k+1})$, where $f_i(t_{k+1}) \equiv f(x_{0i}, T_{k+1})$, $i = 0, 1, ..., n - 1$.

Remarks.

- One can see that we find $v(x_{0i}, T_k)$, $i = 0, 1, ..., n - 1$, $k = 0, 1, ..., p - 1$ in the step 1. Hence it is possible immediately go to step 3 to find $v(x_i, z_j)$. However, numerous numerical experiments showed that step 1 computes the image rays $(x(x_{0i}, T_k), z(x_{0i}, T_k))$ significantly more accurately than the velocity $v(x_{0i}, T_k)$. And Step 2 which is very simple, significantly improves the accuracy of $v(x_{0i}, T_k)$.
- As we have shown in Section 2.6 the inverse problem is numerically unstable. The use of the least squares polynomials suppresses the growth of the small bumps which naturally appear in result of computations, and hence, stabilizes the algorithm.
- The main limitation of this algorithm is that it blows up as the image rays come too close to each other or diverge too much.
- One can use the additional output $x_0(x_i, z_j)$ and $t_0(x_i, z_j)$ to convert a time-migrated image to depth rather than perform depth migration with the found velocities $v(x_i, z_j)$.

3.2.2 Level set approach

Level set methods, introduced in Osher and Sethian [Osher and Sethian, 1988], are numerical methods for tracking moving interfaces: they rely in part on the theory of curve and surface evolution given in Sethian [Sethian, 1982, Sethian, 1985] and on the link between front propagation and hyperbolic conservation laws discussed in Sethian [Sethian, 1987]. These techniques recast interface motion as a time-dependent Eulerian initial value partial differential equation. For a general introduction and overview, see Sethian [Sethian, 1999B].

The main idea of a level set method is the representation of a front as the zero level set of some higher dimensional function. In our context, we want to propagate the wave front coinciding with the flat surface at $t = 0$ downward the earth. We embed the wave front into a

2D function $\phi(x,z)$ so that the front is its zero level set. Furthermore, we embed the quantities Q and P defined on the front into 2D functions $q(x,z)$ and $p(x,z)$ so that at each moment of time $Q = q(x,z)\big|_{\{(x,z)|\phi(x,z)=0\}}$ and $P = p(x,z)\big|_{\{(x,z)|\phi(x,z)=0\}}$, i.e., Q and P coincide with q and p on the zero level set of $\phi(x,z)$. Let

$$g_x = \frac{\phi_x}{|\nabla\phi|}, \qquad g_z = \frac{\phi_z}{|\nabla\phi|}.$$

Let us find the system of equations for q and p. First note that

$$v_{nn} = v_{xx}\cos^2\theta - 2v_{xz}\cos\theta\sin\theta + v_{zz}\sin^2\theta.$$

Second, at each point of the the zero level set of ϕ, i.e. at each front point,

$$g_x = \cos\theta, \qquad g_z = \sin\theta.$$

Then we get the following equations for q and p:

$$q_t = v^2 p, \qquad p_t = -\frac{v_{xx}g_x^2 - 2v_{xz}g_xg_z + v_{zz}g_z^2}{v}q. \tag{62}$$

These equations coincide with the equations for Q and P on the front. Here we switch the notation for time from T to t. We will reserve the notation T for auxiliary times in the fast marching parts of our level set algorithm.

Thus, we have to solve the following system of PDE's:

$$\begin{aligned} &\phi_t + v(x,z)|\nabla\phi| = 0,\\ &q_t = v^2(x,z)p,\\ &p_t = -\frac{v_{xx}g_x^2 - 2v_{xz}g_xg_z + v_{zz}g_z^2}{v(x,z)}q. \end{aligned} \tag{63}$$

As before, we have the input data $f(x_0,t) = \frac{v(x_0,t)}{|Q(x_0,t)|}$ given in (x_0,t) space on a $n\times k$ mesh, and we need to obtain $v(x,z)$ in (x,z) space on a $n\times m$ mesh.

Initialization: Set $q(x,z) = 1$, $p(x,z) = 0$, which is correct for the front at $t = 0$. Set $v(x,0) = f(x_0,0)$ and attach labels "x" to the surface points. Set $\phi(x,z) = z$, i.e., make the level set function a signed distance function.

We solve system (63) in the following time cycle: for $k = 0$ to $p-1$ do:

1. Starting with the current "x" points, solve the system

$$q(x,z)|\nabla T| = \frac{1}{f(x_0,T)}, \qquad \nabla x_0\cdot\nabla T = 0$$

   using the Fast Marching time-to-depth conversion algorithm introduced in Section 3.1.2 to find $v(x,z) = f(x,z)q(x,z)$ for the current $q(x,z)$.

2. Attach labels "x" to the accepted points for which T is not greater than the current value of time t_k.

3. Detect the zero level set of ϕ. Find the velocity v at the zero level set of ϕ and build an extension of v solving the system

$$|\nabla d| = 1, \qquad \nabla d\cdot\nabla v_{ext} = 0,$$

   with the boundary conditions $d = 0$ and $v_{ext} = v$ at the zero level set of ϕ, using the Fast Marching Method, as it is suggested in [Sethian, 1996, Sethian, 1999A]. If the extended velocity is built this way, ϕ remains to be the signed distance function if it was it at $t = 0$.

4. Perform a time step: Compute the quantities g_x and g_z for the current ϕ. Find v_{xx}, v_{xz} and v_{zz} by finding least square polynomials for each grid line $x = x_i$ and $z = z_j$ and evaluating their derivatives. Make one forward Euler step for equations (62) to find new q and p. Solve the level set equation

$$\phi_t + v_{ext}|\nabla\phi| = 0$$

from $t = k\Delta t$ to $t = (k+1)\Delta t$ by the forward Euler method with a time step satisfying the CFL condition.

The main advantage of this algorithm in comparison with the ray tracing algorithm is that it can work even if the image rays intersect, since it tracks the first arrival front.

Having obtained the true seismic velocities $v(x_i, z_j)$ one can perform depth migration to obtain an improved seismic image in the Cartesian coordinates. Alternatively, knowing the velocity $v(x_i, z_j)$ one can apply Sethian's fast marching method [Sethian, 1996] to obtain $t_0(x_i, z_i)$ and $x_0(x_i, z_i)$ to convert the time migrated image to depth.

4 Synthetic data examples

4.1 Example 1

The example in this section allows us to compare performances of the ray tracing algorithm and level set algorithm with a somewhat typical approach. One typical approach to seismic velocity estimation is to compute the Dix velocities and then apply image ray tracing. Here we will replace the image ray tracing with our time-to-depth conversion algorithm.

We considered the velocity fields of the form:

$$v(x,z) = 1 + \exp\left(-c(x^2 + (z-1)^2)\right), \qquad x_0 \in [-2,2], \qquad t \in [0, 0.7]. \tag{64}$$

We took $c = 0.5$, $c = 1$ and $c = 1.5$. The larger c, the sharper the Gaussian anomaly. For each of these fields we created the input data $f(x_0, t)$ on a 200×200 $x_0 \times t$ mesh and applied each of the three algorithms to them: the time-to-depth conversion, the ray tracing, and the level set. The output $v(x, z)$ is given on 200×200 $x \times z$ mesh.

The exact velocity, the input data (the Dix velocity, the found velocity and the image rays for the sharpest Gaussian anomaly corresponding $c = 1.5$ are shown in Fig. 13. We see that the Dix velocity qualitatively differs from the exact velocity and the found velocity resembles the exact velocity much more than the Dix velocity.

The results are summarized in Table (1). We see that

Table 1: The maximal relative errors produced by the time-to-depth conversion, the ray tracing and the level set algorithms on the data from the velocity field (64).

Algorithm	*Time-to-depth*	*Ray tracing*	*Level set*
$c = 0.5$	0.31	0.023	0.078
$c = 1$	0.44	0.11	0.079
$c = 1.5$	0.49	0.29	0.20

- the ray tracing and the level set produce significantly more accurate results than the typical approach;
- the ray tracing approach is more accurate than the level set where the image rays diverge moderately, while it becomes less accurate as the divergence of the image rays increases.

Note that if the image rays diverge severely so that the derivative v_{nn} (or, in different notations, v_{qq}) becomes large, both our ray tracing and level set algorithms blow up, while the time-to-depth convergence algorithm produces inaccurate but stable results.

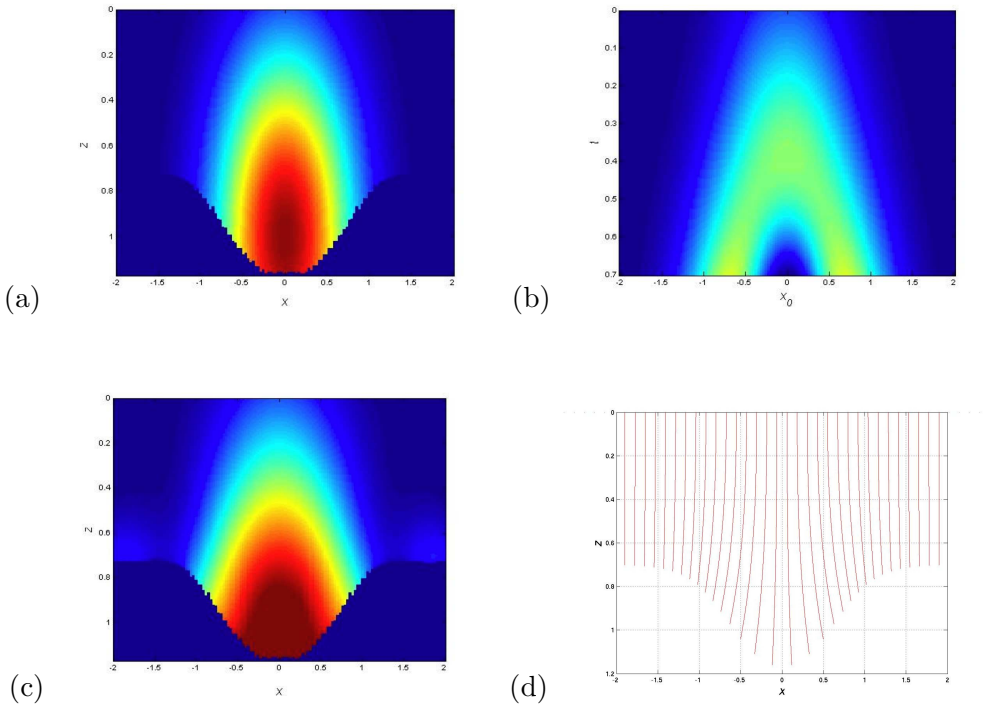


Figure 13: (a): the exact velocity $v(x,z)$; (b): the input data $f(x_0,t) \equiv v_{Dix}(x_0,t)$; (c): the found velocity $v(x,z)$; (d) the image rays.

4.2 Example 2

In this section we also consider an example with a Gaussian anomaly, but with numbers closer to the real seismic numbers:

$$v(x,z) = 2 + 2\exp\left(-(x^2 + (z-2)^2)\right),$$
$$x_0 \in [-3,3], \quad t \in [0,1],$$

The center of the anomaly lies at the depth of 2 km and the background velocity is 2 km/sec. The results (Fig. 14) are produced by the level set algorithm. The found velocity resembles the exact velocity while the Dix velocity and the found velocity differ qualitatively.

5 Field data example

In this section we consider a field data example coming from the North Sea (Fig. 15, left). The main feature in this image is the salt dome. Typically, the velocity inside the salt is higher than it is in the surrounding rock. Salt is light and it pushes the layers up as it comes from inside the earth. The lateral velocity variation here is severe according to the geophysical standards. Note rapidly changing values inside the salt dome, which indicates that the lateral velocity variation is too large for the time migration.

In Fig. 15, right, the time migration velocities chosen in the process of making this image are shown. The Dix velocities were then obtained from them and smoothed (Fig. 16(a)). The level set algorithm was then applied to these Dix velocities to obtain seismic velocities $v(x,z)$ (Fig. 16(b)). The image rays were computed from the found $v(x,z)$ by shooting characteristics.

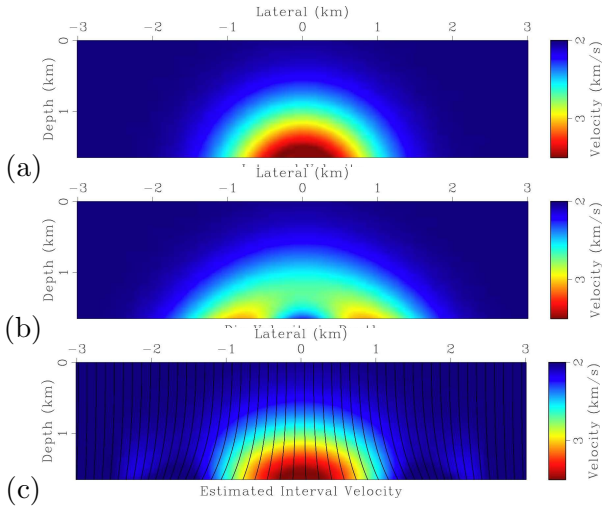


Figure 14: (a): the exact velocity $v(x, z)$; (b) the input data: the Dix velocity converted to depth by "vertical stretch"; (c): the found velocity $v(x, z)$ and the image rays.

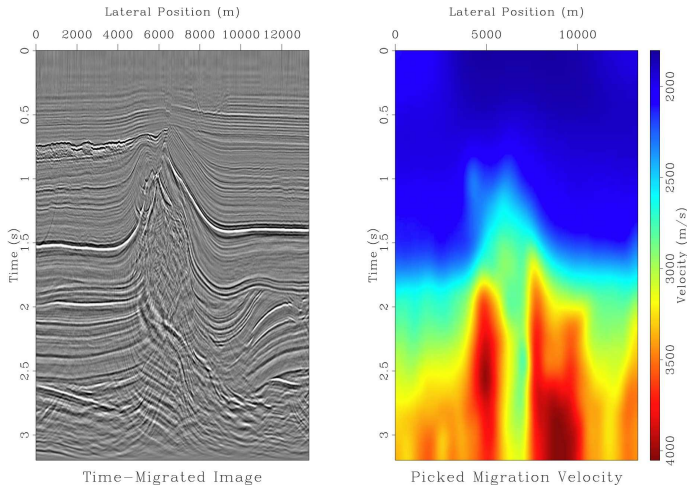
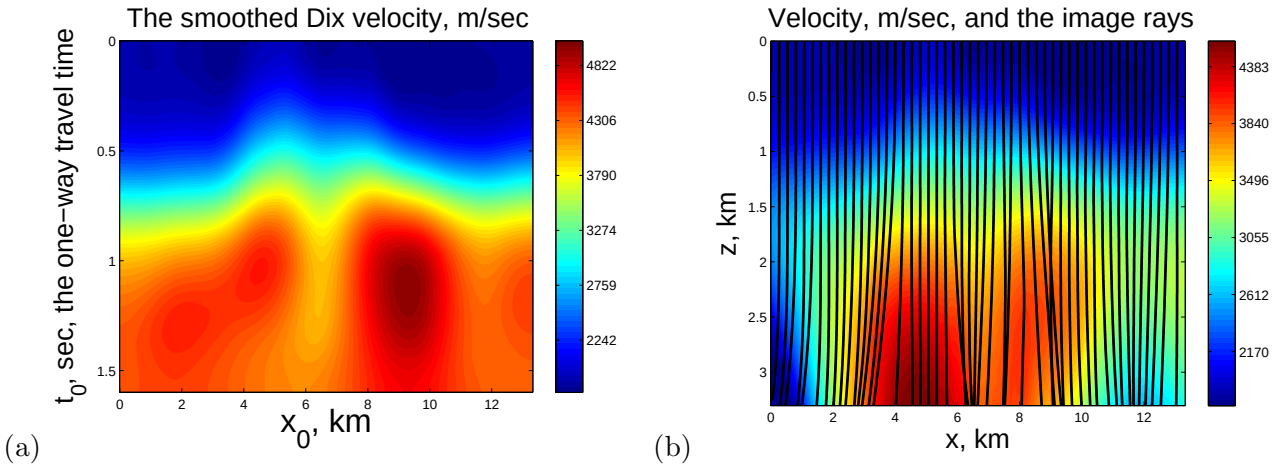


Figure 15: Left: seismic image from North Sea obtained by prestack time migration using velocity continuation [Fomel, 2003]. Right: the corresponding time migration velocity.

The depth domain (x, z) was cut at 3.3 km to make the the found $v(x, z)$ into a rectangular matrix.



(a) (b)

Figure 16: (a) The smoothed Dix velocity $v_{Dix}(x_0, t_0)$; (b) the found seismic velocity $v(x, z)$ and the image rays computed from it.

The depth migrated image, built using the calculated $v(x, z)$, is shown in Fig. 17. The image is in the regular Cartesian coordinates. It shows subsurface structures up to 3.3 km in depth which is quite deep according to geophysical standards. There is a noisy reconstruction inside the salt dome but the surrounding layers are resolved well. Overall, this image looks reasonable.

6 Conclusions

We derived a relation between the Dix velocities and the true seismic velocities in 2D, and a relation between parameters of the emerging wave front and the true seismic velocities in 3D. We stated an inverse problem and showed that it is ill-posed. We introduced three numerical algorithms for obtaining seismic velocities as functions of the regular Cartesian coordinates and tested them on synthetic data examples. We demonstrated that the Dix velocity can be qualitatively different from the true seismic velocity. We applied our algorithms to field data with severe lateral velocity variation and obtained a reasonably looking seismic image in regular Cartesian coordinates up to 3.3 km in depth.

References

[Adalsteinsson and Sethian, 2002] Adalsteinsson, D. and Sethian, J.A., Transport and Diffusion of Material Quantities on Propagating Interfaces via Level Set Methods, J. Comp. Phys, **185**, 1, pp. 271-288, 2002.

[Červený, 2001] Červený, V., Seismic Ray Theory: Cambridge University Press, 2001.

[Dix, 1955] Dix, C. H., Seismic velocities from surface measurements: Geophysics, **20**, pp. 68-86, 1955

[Fomel, 2003] Fomel, S., Time-migration velocity analysis by velocity continuation: Geophysics, **68**, pp. 1662-1672, 2003

[Hubral, 1977] Hubral, P., Time migration - Some ray theoretical aspects: Geophys. Prosp., **25**, pp. 738-745, 1977

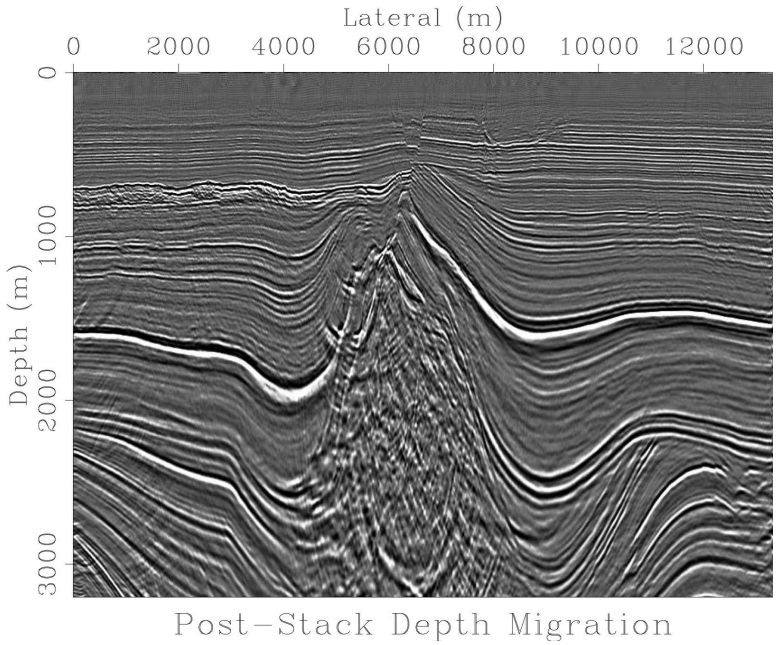


Figure 17: The poststack depth migrated image obtained with the found $v(x, z)$.

[Hubral and Krey, 1980] Hubral, P., Krey, T., Interval velocities from seismic reflection time measurements: SEG, 1980.

[Osher and Sethian, 1988] Osher S., Sethian, J. A., Front propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comp. Phys., **79**, pp. 12-49, 1988.

[Popov, 2002] Popov, M. M., Ray theory and gaussian beam method for geophysicists: Salvador: EDUFBA, 2002.

[Popov and Pšenčik, 1978] Popov, M. M, Pšenčik, I., Computation of ray amplitudes in inhomogeneous media with curved interfaces: Studia Geoph. et Geod., **22**, pp. 248-258, 1978.

[Sethian, 1982] Sethian, J. A., An Analysis of Flame Propagation: Ph.D. Dissertation, Department of Mathematics, University of California, Berkeley, CA, 1982.

[Sethian, 1985] Sethian, J. A., Curvature and the Evolution of Fronts: Commun. in Math. Phys., **101**, pp. 487-499, 1985.

[Sethian, 1987] Sethian, J. A., Numerical methods for propagating fronts in *Variational Methods for Free Surface Interfaces*, (eds. P. Concus & R. Finn), Springer-Verlag, NY, 1987.

[Sethian, 1996] Sethian , J. A., A Fast Marching Level Set Method for Monotonically Advancing Fronts: Proceedings of the National Academy of Sciences, 93, 4, 1996.

[Sethian, 1999A] Sethian, J.A., Fast Marching Methods: SIAM Review, Vol. 41, No. 2, pp. 199-235, 1999.

[Sethian, 1999B] Sethian, J. A., Level set methods and fast marching methods: Cambridge University Press, 1999.

[Sethian and Vladimirsky, 2003] Sethian, J. A., Vladimirsky, A., Ordered Upwind Methods for Static Hamilton-Jacobi Equations: Theory and Algorithms: SIAM J. Numer. Anal., 41, 1, pp. 325-363, 2003

[Shah, 1973] Shah, P. M., Use of wavefront curvature to relate seismic data with subsurface parameters: Geophysics, **38**, 812-825, 1973.

[Tsitsiklis, 1995] Tsitsiklis, J.N., Efficient Algorithms for Globally Optimal Trajectories: IEEE Tran. Automatic Control, 40, pp. 1528-1538, 1995.

[Yilmaz, 2001] Yilmaz, O., Seismic Data Analysis: Soc. of Expl. Geophys., 2001