

Using level set based inversion of arrival times to recover shear wave speed in transient elastography and supersonic imaging

Joyce McLaughlin and Daniel Renzi

Mathematics Department, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA

E-mail: mclauj@rpi.edu and renzid@rpi.edu

Received 2 March 2005, in final form 14 November 2005

Published 27 March 2006

Online at stacks.iop.org/IP/22/707

Abstract

Transient elastography and supersonic imaging are promising new techniques for characterizing the elasticity of soft tissues. Using this method, an ‘ultrafast imaging’ system (up to 10 000 frames s^{-1}) follows in real time the propagation of a low-frequency shear wave. The displacement of the propagating shear wave is measured as a function of time and space. Here we develop a fast level set based algorithm for finding the shear wave speed from the interior positions of the propagating front. We compare the performance of level curve methods developed here and our previously developed (McLaughlin J and Renzi D 2006 Shear wave speed recovery in transient elastography and supersonic imaging using propagating fronts *Inverse Problems* **22** 681–706) distance methods. We give reconstruction examples from synthetic data and from data obtained from a phantom experiment accomplished by Mathias Fink’s group (the Laboratoire Ondes et Acoustique, ESPCI, Université Paris VII).

1. Introduction

In this paper, we significantly improve the speed of the distance methods algorithm, [28], for shear wave speed recovery in transient elastography or supersonic imaging; see also [38]. As a background for our targeted application, we recall that the long range goal is to create shear stiffness images in tissue thereby extending, quantifying, and making a visual representation of the tissue mechanical property felt in doctors’ palpation exams. In each of the experiments proposed so far the tissue is displaced, and the displacements on a dense grid *interior* to the tissue are obtained either from ultrasound- [4, 5, 7, 13, 14, 21–23, 31–35, 39–41, 43–45] or MR-based [6, 16, 24–27, 30] experiments. Static displacements, harmonic oscillations, or transient pulses are used to create displacements.

Here we target the transient experiments developed in the laboratory of Mathias Fink, where a line source on the surface of the body [7, 14, 39–41, 43], or on a line orthogonal to

the surface [4, 5], creates a propagating shear wave with an identifiable front in the downward or axial component of the elastic vector displacement. Our current goal is to recover the shear wave speed from the times when the moving front of the shear wave arrives at each point in the medium. These arrival times are determined from the *interior* displacement measurement of the axial component.

Prior work of the authors to show uniqueness, to create images from single frequency data content, and to establish the equation satisfied by the arrival time is contained in [17–20, 29].

For the reasons described in [28], we assume that the measured displacement component satisfies the following wave equation:

$$\nabla \cdot (\mu(x)\nabla u(x, t)) = \rho(x)u_{tt}(x, t) \quad \text{in } \Omega \times (0, T), \quad (1.1)$$

where $0 < \mu, \rho \in C^1(\bar{\Omega})$. Consistent with the transient elastography experiment, we assume that the medium is initially at rest satisfying the homogeneous initial condition,

$$u(x, 0) = u_t(x, 0) = 0 \quad \text{on } \Omega, \quad (1.2)$$

and one of the following boundary conditions,

$$u(x, t) = f(x, t) \quad \text{on } \partial\Omega \times (0, T), \quad (1.3)$$

or

$$\mu(x)\nabla u(x, t) \cdot \nu(x) = g(x, t) \quad \text{on } \partial\Omega \times (0, T), \quad (1.4)$$

where ν is the outward normal to $\partial\Omega$. The goal is to find the shear wave speed, $\sqrt{\mu(x)/\rho(x)}$ from the first arrival times, $\hat{T}(x)$, of the propagating shear wave. These first arrival times are defined as

$$\hat{T}(x) := \inf\{t \in (0, T) : |u(x, t)| > 0\}, \quad (1.5)$$

and are related to the shear wave speed via the Eikonal equation:

$$|\nabla \hat{T}| \sqrt{\mu/\rho} = 1. \quad (1.6)$$

So, determining the wave speed requires a two-part procedure. First, the arrival times of the propagating shear wave have to be determined from the *interior* displacement data. Then the wave speed can be calculated from the arrival times using the Eikonal equation. In [28], we

- described a cross-correlation technique to determine the arrival times;
- demonstrated that simple procedures that use the Eikonal equation can lead to large outliers in the recovered wave speed;
- developed a more robust scheme called the distance method.

As we pointed out in [28], while the distance method does a very good job of reducing the problem of outliers, it can be a little slow. Here we will show the equivalence of the distance method and taking time derivatives on the zero level set of a function $\phi(x, t)$ which has the following properties:

- $\phi(x, \hat{T}(x)) = 0$;
- $\phi(x, t) = \pm \inf_{\hat{x}} \{|\hat{x} - x| : \hat{T}(\hat{x}) = t\}$.

What we have done here is borrow the idea of extending $\phi(x, t)$ using a signed distance function from the level set algorithm literature [36, 42]. The result is that $\phi(x, t)$ satisfies a beautiful, simple linear equation,

$$\phi_t(x, y) = \sqrt{\mu/\rho}. \quad (1.7)$$

In this paper, we show that using (1.7) instead of the distance method significantly improves the speed of our inverse algorithm. To do this, we compute the signed distance

function in a narrow band about the surface $\hat{T}(x) = t$. The level set community has been computing time slices of the signed distance function for many years [1, 9] and have coined the term ‘reinitialization’. Typically, this is accomplished by solving a constant coefficient Eikonal equation. A straightforward extension to computing the signed distance function, at all times which we will need for our inverse algorithm, involves solving an Eikonal equation in every time slice. However, before these Eikonal equations can be solved, each problem has to be set up correctly because (1) the Eikonal equation is discretized on a regular grid, but the boundary condition is given on the curve $\hat{T}(x) = t$; and (2) the appropriate narrow band has to be determined for each time slice. This process is called ‘initialization’. Unfortunately, quickly initializing all the problems in each time slice is quite a bit more complicated and requires a great deal of bookkeeping. Here we present our procedure where we use a hashing function to separate the grid points into ‘reinitialization’ problems. This allows us to simultaneously do all of the initializations. This procedure is described in section 1, and numerical tests of the convergence rate are given. Finally, the complete algorithm is tested on both synthetic data and experimental data from the laboratory of Mathias Fink (Laboratoire Ondes et Acoustique, ESPCI, Université Paris VII) and results are compared with the results given in [28] for the slower distance method.

The remainder of this paper is composed as follows. We will give short summaries of the cross-correlation technique and the distance method in sections 2 and 3. Then we describe how to speed up the distance method in section 4. We test the complete algorithm on both synthetic and experimental data in section 5. Finally, we give our conclusions and suggestions for future work in section 6.

2. Finding arrival times from displacement: a brief summary

In [28], we investigated techniques for finding the first arrival times, $\hat{T}(x)$, $x \in R^n$, $n = 2, 3$ given the displacement $u(x, t)$. Here we give a brief summary of those investigations. The basic idea is that the overall shape of the shear wave pulse is preserved as the wave propagates through the medium. So, parts of the displacement data containing the propagated initial pulse should be highly correlated. We take advantage of this by using a cross-correlation technique. The displacement time history $u(x, t)$ at each spatial point x is cross-correlated against the displacement time history $u(x_{\text{ref}}, t)$ at a single reference point x_{ref} , chosen near the source. We compute the biased cross-correlation for each x using the following formula:

$$C(x, \delta t) := \frac{1}{T} \int_0^T u(x_{\text{ref}}, t) \tilde{u}(x, t - \delta t) dt,$$

$$\tilde{u}(x, t) = \begin{cases} u(x, t) & \text{if } 0 \leq t \leq T, \\ u(x, t - T) & \text{if } t > T, \\ u(x, t + T) & \text{if } t < 0. \end{cases}$$

Now we estimate the arrival times by $\hat{T}(x) \approx \delta t_{\text{max}}$, where

$$\delta t_{\text{max}} := \operatorname{argmax}_{\delta t \in [0, T]} C(x, \delta t).$$

So $\hat{T}(x)$ is estimated by the time delay δt that maximizes the correlation between the signals $u(x_{\text{ref}}, t)$ and $\tilde{u}(x, t - \delta t)$. In a later paper we will improve this technique and the result will be to eliminate some of the artefacts we discuss below.

3. Distance method summary

In this section, we summarize solution techniques developed in [28] for the inverse Eikonal equation: given arrival times that satisfy

$$|\nabla \hat{T}| \sqrt{\mu/\rho} = 1, \quad (3.1)$$

find the wave speed, $\sqrt{\mu/\rho}$. Note also that in [20], we established that Lipschitz continuous first arrival times, \hat{T} , satisfy (3.1). In [28], we showed that because the wave speed, $\sqrt{\mu/\rho}$, is inversely proportional to $|\nabla \hat{T}|$, noise in the first arrival times can drive the denominator close to zero, leading to large outliers in the recovered wave speed if we simply calculate the derivatives in $|\nabla \hat{T}|$ using a forward Euler or a centred difference numerical scheme. We also developed the distance method, which in its second-order form estimates the shear wave speed, $\sqrt{\mu/\rho}$, as

$$\sqrt{\frac{\mu(x)}{\rho(x)}} \approx \frac{1}{2\Delta t} (\inf_{\hat{x}^+} |x - \hat{x}^+| + \inf_{\hat{x}^-} |x - \hat{x}^-|) : \hat{x}^\pm \text{ satisfies } \hat{T}(\hat{x}^\pm) = \hat{T}(x) \pm \Delta t. \quad (3.2)$$

4. Level set formulation

We begin by describing the level set formulation (see [37]). The basic idea is to represent each level curve of \hat{T} , i.e. $\hat{T}(x) = t$, as the zero level set of a higher dimensional Lipschitz continuous function $\phi(x, t)$. That is, $\phi(x, \hat{T}(x)) = 0$. There are many ways to do this. We will use the signed distance function

$$\phi(x, t) = \pm \inf\{|x - \hat{x}| : \hat{x} \text{ satisfies } \hat{T}(\hat{x}) = t\}, \quad (4.1)$$

where the plus (minus) sign is chosen if $t > \hat{T}(x)$ ($t < \hat{T}(x)$), respectively. So, at each point (x, t) , $\phi(x, t)$ is the distance to the level curve $\hat{T}(x) = t$. This is illustrated in figure 1. Furthermore, if we fix $t = t_0$, then the zero level set of $\phi(x, t_0)$ is the contour $\hat{T}(x) = t_0$. We note that this is not exactly the standard level set notation. Usually, the contours of \hat{T} are assumed to be closed curves. These curves segment R^2 into two pieces, a piece inside the curve and a piece outside the curve. The sign of $\phi(x, t)$ is then chosen positive (negative) if x is outside (inside) the contour $T(x) = t$. In our case, the contours of \hat{T} are not necessarily closed curves. Consider the curve shown in figure 1. It is an arrival time contour generated from a plane wave source. Even though it is not closed, it does segment Ω into two pieces, a piece that is still at rest, and a piece that has experienced wave propagation. So the domain of ϕ and its sign convention have been changed to accommodate our particular domain segmentation. That is, $\phi(x^1, t_0) \leq 0$ when (x^1, t_0) is still at rest and $\phi(x^2, t_0) > 0$ when (x^2, t_0) has experienced wave propagation.

Now we derive the level set equation. The level set function, ϕ , is Lipschitz continuous on $\Omega \times [0, T]$ since \hat{T} is Lipschitz continuous and ϕ is the distance function to a continuous curve. This implies that $(\nabla_x \phi, \phi_t)$ exists a.e. Furthermore, we can apply the chain rule (see [11]) to $\phi(x, \hat{T}(x)) = 0$ to get

$$\nabla_x \phi = -\phi_t \nabla_x \hat{T} \quad \text{a.e. on } \{(x, t) : \phi(x, t) = 0\}. \quad (4.2)$$

By our sign convention for ϕ we have $\phi_t \geq 0$ on the zero level set of ϕ . Taking the norm of both sides of (4.2) and using the Eikonal equation, $\sqrt{\mu/\rho} |\nabla_x \hat{T}| = 1$, to eliminate $|\nabla_x \hat{T}|$ leads to the level set equation

$$\phi_t = \sqrt{\mu/\rho} |\nabla_x \phi| \quad \text{a.e. on } \{(x, t) : \phi(x, t) = 0\}. \quad (4.3)$$

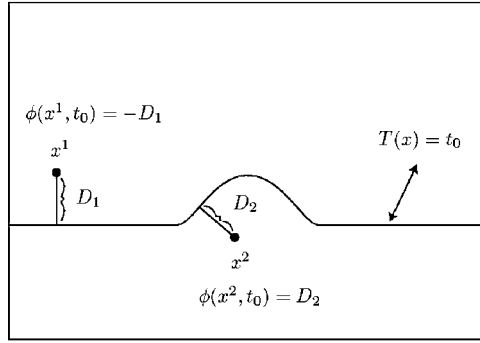


Figure 1. There is a plane wave source at the bottom of this figure. At the time t_0 , the wave has reached the curve $T(x) = t_0$. This curve segments the domain into two segments: the top and bottom. In the top segment, where the medium is still at rest, $\phi(x, t_0) = -D_1$, where D_1 is the distance to the curve $T(x) = t_0$ from the point x . In the bottom segment, where the medium has experienced wave propagation, $\phi(x, t_0) = D_2$, where D_2 is the distance to the curve $T(x) = t_0$ from the point x . For more information on the distance function see [36, 42].

Note that this equation is defined only on the zero level set of ϕ . However, ϕ is defined in a narrow band about the surface $S_{\hat{T}} = \{x | \phi(x, \hat{T}(x)) = 0\}$ so ϕ_t can be approximated on $S_{\hat{T}}$. Furthermore, since ϕ is the signed distance function, $|\nabla_x \phi| = 1$.

Now we see the advantage of this formulation. Recall that if one solves the Eikonal equation for the wave speed, $\sqrt{\mu/\rho} = 1/|\nabla \hat{T}|$, then the $|\nabla \hat{T}|$ term appears in the denominator. Now, we have the elegant formula

$$\phi_t(x, t) = \sqrt{\mu/\rho}. \quad (4.4)$$

Since ϕ is Lipschitz continuous, we know ϕ_t exists a.e. Using a forward Euler approximation scheme for the time derivative in equation (4.4) is equivalent to finding the distance from x to the level curve $\hat{T}(\hat{x}) = \hat{T}(x) + \Delta t$ and dividing by Δt as in the first-order distance method; see [28]. That is, if our point (x, t) is on the zero level set, the method above is equivalent to

$$\begin{aligned} \sqrt{\frac{\mu(x)}{\rho(x)}} &\approx \frac{\phi(x, \hat{T}(x) + \Delta t) - \phi(x, \hat{T}(x))}{\Delta t} = \frac{\phi(x, \hat{T}(x) + \Delta t)}{\Delta t} \\ &= \phi(x, \hat{T}(x) + \Delta t) = \inf\{|x - \hat{x}| : \hat{x} \text{ satisfies } \hat{T}(\hat{x}) = \hat{T}(x) + \Delta t\}. \end{aligned} \quad (4.5)$$

Clearly the accuracy using a forward Euler approximation scheme for ϕ_t when ϕ_t is Lipschitz continuous is $\mathcal{O}(\Delta t)$ which we have also shown for the first-order distance method, in [28], using a different argument.

Higher order approximations are obtained in the obvious way when ϕ_{tt} is also Lipschitz continuous on the zero level set using higher order discretization schemes for ϕ_t . For example, using a centred difference approximation scheme for ϕ_t , and writing the difference equations on the zero level set would result in the following $\mathcal{O}((\Delta t)^2)$ accurate equation:

$$\sqrt{\frac{\mu(x)}{\rho(x)}} \approx \frac{\phi(x, \hat{T}(x) + \Delta t) - \phi(x, \hat{T}(x) - \Delta t)}{2\Delta t}. \quad (4.6)$$

We easily see that (4.6) is just

$$\sqrt{\frac{\mu(x)}{\rho(x)}} \approx \left\{ \frac{1}{2\Delta t} (\inf_{\hat{x}^+} |x - \hat{x}^+| + \inf_{\hat{x}^-} |x - \hat{x}^-|) : \hat{x}^\pm \text{ satisfies } \hat{T}(\hat{x}^\pm) = \hat{T}(x) \pm \Delta t \right\}, \quad (4.7)$$

which is the second-order distance method.

While equations (4.6) and (4.7) are identical, they suggest two different algorithms. To solve (4.7) we find all the contours of T , and then calculate the distance from each point on each contour to its neighbouring contour. On the other hand, to solve (4.6) we first build the signed distance function $\phi(x, t)$, and then interpolate ϕ to get the values in the discretization stencil of (4.6). We refer to equations based on this approach as *inverse level curve methods*. Specifically, we will refer to equation (4.6) as the second-order level curve method. The distance methods are easy to implement with a standard contour plotter in 2D and are $\mathcal{O}(n^3)$, where n is the number of points in time. The level curve methods are much faster, but the implementation is much more involved.

So for the level curve method we start with $\{x : \hat{T}(x) = t\}$. For each fixed time, t , then we find $\phi(x, t)$ (see [1, 9]), by solving the following Eikonal equation:

$$\begin{aligned} |\nabla\phi(x, t)| &= 1 \\ \phi(x, t) &= 0 \quad \forall \{x : \hat{T}(x) = t\}. \end{aligned} \quad (4.8)$$

In level set terminology solving equation (4.8) (see [1, 9]) is often referred to as ‘reinitialization’. One approach to solve this ‘reinitialization’ problem is to first find the distance from the contour $\hat{T}(x) = t$ to all of the mesh points that neighbour this contour. This first step is usually referred to as initialization. Second, the distance at all other points in the domain to the contour $\hat{T}(x) = t$ is calculated using a forward Eikonal solver. Typically the initialization step requires $\mathcal{O}(m)$ steps, and the forward Eikonal solver takes $\mathcal{O}(m \log m)$ steps, where m is the number of points in space, which we assume is approximately n^2 , where n is the number of points in time. More recently, forward Eikonal solvers have been developed that only require $\mathcal{O}(m)$ steps. For our problem we have to solve equation (4.8) for every time step. Applying an ‘initialization’ procedure to each ‘reinitialization’ problem would take at least $\mathcal{O}(n^3)$ time. Implementing our algorithm this way would have at least the same running time as the distance method. Instead we do the following.

First we note that we do not have to find the distance to each contour at every point in the domain, we only need to make sure that we compute the distance function everywhere it is needed in the discretization stencil of (4.6). In our case this means that we only calculate the values of ϕ on the discretization stencil used for the inverse level curve method. For example, to use a centred difference approximation for ϕ_t , we only need to calculate the distance, $\phi(\hat{x}, t_i)$, to each contour, $T(x) = t_i$, for those points, \hat{x} , between the two neighbouring contours $T(x) = t_{i-1}$ and $T(x) = t_{i+1}$. That is, we only need $\phi(\hat{x}, t_i)$ for $\{\hat{x} | t_{i-1} < T(\hat{x}) < t_{i+1}\}$. This is very similar to the idea of ‘narrow-banding’ for the level set method; see [1, 9]. An advantage here is that for the inverse problem, we know exactly which points are needed in the band, so we can obtain an optimally small band. So, for the above example using a centred difference scheme, each point, x , will only be part of two ‘reinitialization’ problems.

Now, what a narrow banded forward Eikonal solver needs for input to solve (4.8) for one time step t_i is as follows.

- A list of grid points in the domain (i.e. grid points between the contours $T(x) = t_{i-1}$ and $T(x) = t_{i+1}$).
- A list of grid points that neighbour the contour $T(x) = t_i$ and the distance from each of these points to this contour. These are the initial ‘accepted’ points in the fast marching method; see [42].

Of course we need these lists for every time step t_i .

So what is new here is that we extend the ‘initialization’ algorithm in [8] to find the input information that a narrow banded forward Eikonal solver needs for all time steps simultaneously. To aid in the description of our ‘initialization’ algorithm, we first describe the

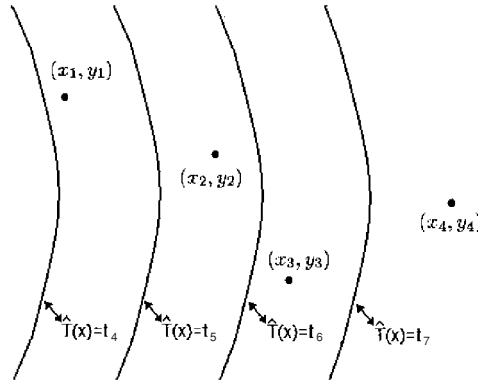


Figure 2. There are four points in this figure. The hashing function g applied to all these points is $g(x_1, y_1) = t_4$, $g(x_2, y_2) = t_5$, $g(x_3, y_3) = t_6$ and $g(x_4, y_4) = t_7$.

way the ‘initialization’ information is stored. For storage we use arrays of stacks as follows: we use an array of stacks labelled ‘domain’. So, for each i , $\text{domain}[i]$ is a stack and it will store the grid points in the domain of equation (4.8) for $t = t_i$. Likewise, we use two additional arrays of stacks labelled ‘neighbour’ and ‘distance’. For each i , $\text{neighbour}[i]$ is a stack that stores the neighbouring grid points of the contour $T(x) = t_i$, and $\text{distance}[i]$ is a stack that stores the distance from the neighbouring grid points to the contour $T(x) = t_i$. At every grid point the algorithm must push this grid point on to the appropriate stack(s). In addition, if the grid point neighbours one of the contours, then the distance to this contour is calculated and pushed onto the appropriate stack.

Essentially, at each mesh point, we accomplish this by performing Chopp’s algorithm [8], on the correct ‘reinitialization’ problem. The goal is to make the running time of the entire method $O(m)$. To achieve this, we will utilize the fact that there exists an $O(m)$ Eikonal solver. Our intermediate goal then is to determine the correct initialization in constant time (i.e. no searching or sorting), resulting in an $O(m)$ running time of the entire method. We accomplish this by using a hashing function, which in two spacial dimensions where now $(x, y) \in R^3$, is

$$g(x, y) = \text{int}((T(x, y) - t_0)/(\Delta t)),$$

where ‘int’ takes a floating point number and truncates it into an integer, and (x, y) is a coordinate pair. This function returns the index of the contour with the largest arrival time that is less than the arrival time at the grid point (x, y) . For example, assume that the grid point (\hat{x}, \hat{y}) lies between the i th contour, $T(x, y) = t_i$, and $(i + 1)$ th contour, $T(x, y) = t_{i+1}$. That is, $t_i \leq T(\hat{x}, \hat{y}) < t_{i+1}$. Now, evaluating the hashing function g at the grid point (\hat{x}, \hat{y}) results in $g(\hat{x}, \hat{y}) = i$, the index of the i th contour, $T(x, y) = t_i$. This is illustrated in figure 2.

Now, to see how the hashing function g helps us determine the correct reinitialization problem at a mesh point (x_i, y_j) , we first consider the rectangle R_{ij} , whose corners are (x_i, y_j) , (x_{i+1}, y_j) , (x_i, y_{j+1}) and (x_{i+1}, y_{j+1}) . It is obvious that if a contour passes through this rectangle, then the mesh points at the corners of the rectangle neighbour the contour. Using the hashing function g , it is easy to determine whether any of the contours pass through this rectangle. To explain how this is done let

$$T1(R_{ij}) = \min(g(x_i, y_j), g(x_{i+1}, y_j), g(x_i, y_{j+1}), g(x_{i+1}, y_{j+1}))$$

$$T2(R_{ij}) = \max(g(x_i, y_j), g(x_{i+1}, y_j), g(x_i, y_{j+1}), g(x_{i+1}, y_{j+1})).$$

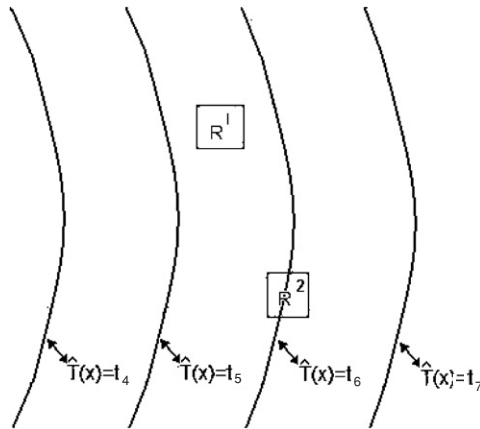


Figure 3. Determining the neighbouring grid points of a contour. Here, the corners of the rectangle R^1 lie between the contours $\hat{T}(x) = t_5$ and $\hat{T}(x) = t_6$, while the corners of rectangle R^2 neighbour the contour $\hat{T}(x) = t_6$. In this case, $T1(R^1) = T2(R^1) = T1(R^2) = 5$ and $T2(R^2) = 6$.

Now, if $T1 = T2$, then no contours pass through R_{ij} . Rectangle R^1 in figure 3 illustrates this. As the corner points of rectangle R^1 must lie between the contours indexed by $T1$ and $T1 + 1$, these points are pushed onto the stacks $\text{domain}[T1]$ and $\text{domain}[T1 + 1]$. If $T1 \neq T2$, then the contour indexed by $T2$ passes through the rectangle. Rectangle R^2 in figure 3 illustrates this. Since all the corner points of rectangle R^2 are neighbouring points of this contour, the distance from each corner point of the rectangle to the contour is calculated and pushed onto the stack $\text{distance}[T2]$, while the corner points of the rectangle R_{ij} are pushed onto the stacks $\text{neighbour}[T2]$, $\text{domain}[T2 - 1]$, $\text{domain}[T2]$ and $\text{domain}[T2 + 1]$. Note that in this explanation we are assuming that there is at least one grid point between contours.

To summarize, here is the complete algorithm for solving the inverse Eikonal equation in two dimensions:

- (1) Discretize time into $N+1$ equally spaced points $t_k, k = 0, 1, 2, \dots, N$.
- (2) Discretize the spacial domain into a $M + 1 \times M + 1$ equally spaced grid $(x_i, y_j), i, j = 0, 1, 2, \dots, M$.
- (3) FOR $i = 1$ to $M-2$ DO 4–11.
- (4) FOR $j = 1$ to $M-2$ DO 5–11.
- (5) $T1 = \min(g(x_i, y_j), g(x_{i+1}, y_j), g(x_i, y_{j+1}), g(x_{i+1}, y_{j+1}))$.
- (6) $T2 = \max(g(x_i, y_j), g(x_{i+1}, y_j), g(x_i, y_{j+1}), g(x_{i+1}, y_{j+1}))$.
- (7) IF $T1 = T2$ push (x_i, y_j) onto $\text{domain}[T1]$ and $[T1 + 1]$.
- (8) ELSE DO 9–12.
- (9) Calculate the distance from $(x_{ii}, y_{jj}), ii, jj = 1, 2$ to the contour $\hat{T} = t_{T2}$ and push these values onto $\text{distance}[T2]$ (see [8]).
- (10) Push $(x_{ii}, y_{jj}), ii, jj = 1, 2$ onto $\text{neighbour}[T2]$ and $\text{domain}[T2]$.
- (11) IF $T2 > 0$ push $(x_{ii}, y_{jj}), ii, jj = 1, 2$ onto $\text{domain}[T2 - 1]$.
- (12) IF $T2 < N$ push $(x_{ii}, y_{jj}), ii, jj = 1, 2$ onto $\text{domain}[T2 + 1]$.
- (13) FOR $k = 0$ to N DO 14.
- (14) Use fast marching method to find the distance to the contour $T = t_k$.
- (15) FOR $k = 1$ to $N - 1$ solve equation (4.6).

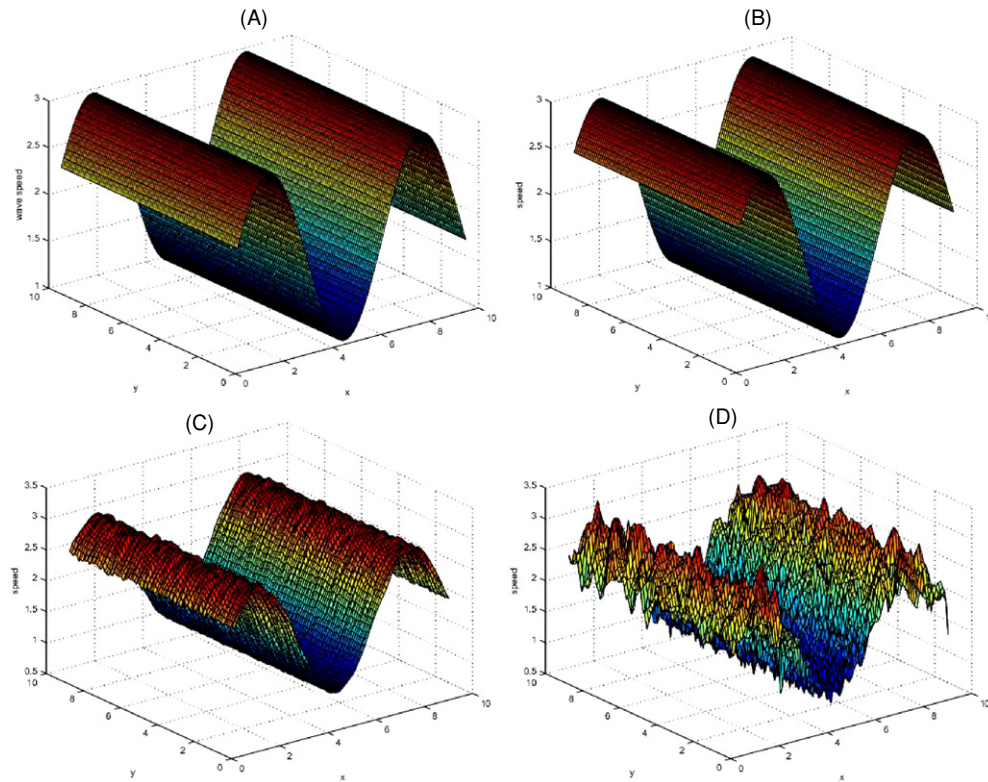


Figure 4. (A) Exact wave speed; (B) wave speed recovery without noise using the second-order level curve method; (C) wave speed recovery with noise ($\gamma = 0.001$) using the second-order level curve method; (D) wave speed recovery with noise ($\gamma = 0.02$) using the second-order level curve method.

Note that we are not solving the problem at the time end points or the grid edges. This is because equation (4.6) requires information outside the time discretization. Likewise, finding the distance from a grid point to a contour would also require information off the grid. It is trivial to modify the above algorithm to utilize lower order methods at the time end points and grid edges. For instance, one could use forward and backward Euler discretizations at the first and last time point, respectively. One could also use the first-order method described in Sethian [42] to find contours, and the distance to neighbouring grid points at the edge of the grid. Alternately, one could use a second-order approximation at the edges using only interior points. For example, the one sided approximation, $\phi_t(x, t_N) = 1.5\phi(x, t_N) - 2\phi(x, t_{N-1}) + 0.5\phi(x, t_{N-2})$, could be used to calculate ϕ_t at the boundary $t = t_N$. Similar approximations could be used at the other edges.

We test this algorithm on the same test problem used to test the distance method in [28]: $\sqrt{\mu(x, y)/\rho(x, y)} = 2 + \sin(x)$, $(x, y) \in [0, 10] \times [0, 10]$, $\hat{T}(0, y) = 0$, (4.9) where \hat{T} satisfies the Eikonal equation. This problem has the exact solution

$$\hat{T}(x, y) = \int_0^x 1/(2 + \sin(x_1)) dx_1. \quad (4.10)$$

The exact wave speed is displayed in figure 4(A). The solution using the level curve method for $\Delta t = 0.1$ is shown in figure 4(B). We repeat these calculations for $\Delta t = 0.05, 0.025$ and

Table 1. L_∞ error for the second-order level curve method.

Δt	L_∞ error second-order distance method
0.2	0.0787
0.1	0.0192
0.05	0.00489

Table 2. Running times for the second-order level curve methods.

Δt	h	Level curve method running time (s)
0.2	0.2	22.8474
0.1	0.1	99.6841
0.05	0.05	393.6220

tabulate the errors in table 1. As expected, the errors appear to be $\mathcal{O}(\Delta t^2)$. To demonstrate that the recovered wave speed with the level curve method does not have problems with outliers we add random Gaussian noise to the arrival times

$$\hat{T}_{\text{noisy}}(x) = \hat{T}(x) + \gamma \text{rand}(x),$$

where $\text{rand}(x)$ is a random number generated in Matlab from the normal distribution with mean zero and variance one, and γ is the noise level. The result for $\Delta t = 0.1$, $\gamma = 0.001, 0.02$ is shown in figures 4(C) and (D). We can see that the fast method has the same good stability properties as the slower distance method given in [28].

Finally, we tabulate the running times of the algorithm for $\Delta t = 0.2, 0.1, 0.05$ in table 2. As before, we halve the grid spacing every time the time step is halved. From these calculations, it appears that the running time is $\mathcal{O}(n^2)$ or $\mathcal{O}(m)$, where n is the number of points in time, and m is the number of points in space.

5. Testing the complete shear wave speed recovery algorithm

5.1. Numerical tests on synthetic data

In this section, we test the second-order level curve method on the same synthetic problems given in [28]: the acoustic wave equation,

$$\nabla \cdot (\mu(x) \nabla u(x, t)) = \rho(x) u_{tt}(x, t) \quad \text{in } \Omega \times (0, T), \quad (5.1)$$

with zero initial conditions and the following boundary conditions

$$\frac{\partial u}{\partial x_1}(0, x_2, t) = \frac{1}{2\sqrt{\pi}c_t} \cos(\tau(t - t_0)) \exp\left\{-\frac{(t - 0.02)^2}{2c_t^2}\right\},$$

$$u(x_1, 6, t) = u(x_1, -6, t) = u(6, x_2, t) = 0.$$

In [28], the wave speed is chosen to be

$$\sqrt{\frac{\mu(x_1, x_2)}{\rho(x_1, x_2)}} = \left(300 + 900 \exp\left\{-\frac{(x_1 - 3.2)^2}{0.25^2} - \frac{(x_2 - 0)^2}{0.5^2}\right\}\right)^2 \text{ cm s}^{-1},$$

and the displacement is obtained for three values of (c_t, τ) , which are $(c_t, \tau) = (0.0018, 250/(2\pi))$, $(0.009, 125/(2\pi))$, $(0.0045, 62.5/(2\pi))$. Note that $\tau/(2\pi)$ is the central frequency of the experiment.

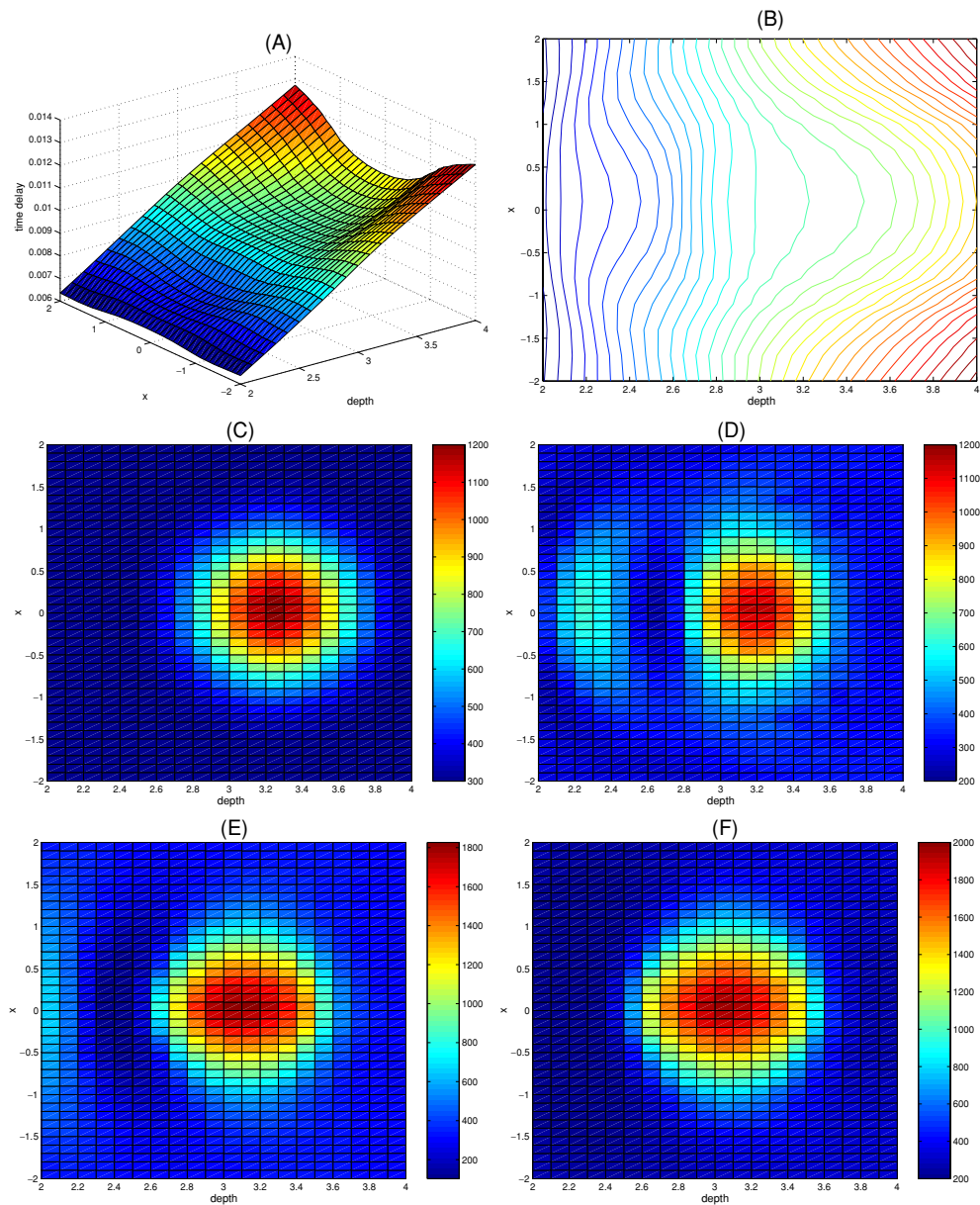


Figure 5. (A) Arrival time for 250 Hz synthetic data; (B) level curves of the arrival time surface for 250 Hz synthetic data; (C) exact wave speed; (D) wave speed recovery with arrival times for 250 Hz synthetic data using the second-order level curve method; (E) wave speed recovery with arrival times for 125 Hz synthetic data using the second-order level curve method; (F) wave speed recovery for 62.5 Hz synthetic data using the second-order level curve method; note the change in the colour bars.

Before we can test the level curve method, we first have to calculate the arrival times of the wave, \hat{T} . This is accomplished by the correlation method described in section 2; for details see [28]. The arrival times, when the central frequency is $\tau/(2\pi) = 250$, are shown in figure 5(A). There is a flat spot in the arrival time surface centred at (3.2, 0) indicating a high

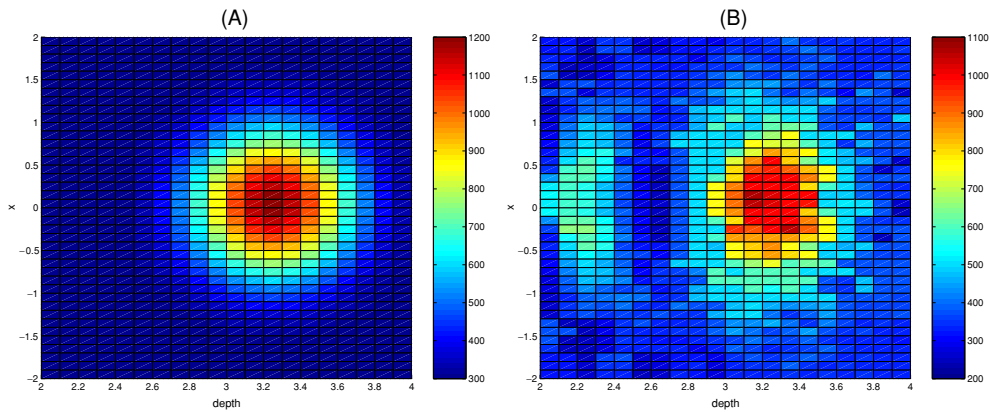


Figure 6. (A) Exact wave speed; (B) wave speed recovery for 250 Hz synthetic data with noise using the second-order level curve method.

speed region. This phenomenon is more clearly observed in the level curves of the arrival time surface, shown in figure 5(B), which bend as they pass through the high speed inclusion. Next, we recover the wave speed from the arrival times using the second-order level curve algorithm. The exact wave speed is shown in figure 5(C) and the recoveries for $\tau/(2\pi) = 250, 125, 62.5$ are shown in figures 5(D)–(F). Note that the colour bars in these figures are not the same. These results are visually almost indistinguishable from the distance algorithm recoveries; see [28]. At high frequencies both the magnitude and support of the high speed region are recovered very well. At lower frequencies the recovery tends to overshoot the fast change, and the support of the high speed region is blurred. There is an artefact about half a wavelength directly in front of the high speed region at all frequencies. This is where the forward and backscattered waves are mixing, causing difficulty for the arrival time identification.

To test the stability of the second-order level curve algorithm, we add Gaussian random noise to the $\tau/(2\pi) = 250$ frequency displacement data

$$u_{\text{noisy}}(x, t) = u(x, t) + \gamma u_{\text{max}} \text{rand}(x, t),$$

where u_{max} is the maximum displacement, γ is the noise level, and $\text{rand}(x, t)$ is a random number generated in Matlab from the normal distribution with mean zero and variance one. We repeat the calculations above to get recoveries with 7% noise ($\gamma = 0.07$). Note that, because of backscattering and wave spreading, behind the inclusion the displacement is more than five times less than the maximum displacement, so the signal-to-noise ratio behind the inclusion is quite small (about 2–3). As in [28], we apply a low pass filter to the cross-correlation function before calculating the arrival times. The recovered wave speed using the level curve method is shown in figure 6(B). The exact wave speed is again shown in figure 6(A). Our noisy recovery compares favourably to the recovery using exact synthetic data given above, and is again almost indistinguishable from the distance method recovery obtained in [28].

5.2. Numerical tests on experimental data

Here we test the level curve algorithm on data measured from a phantom experiment in the laboratory of Mathias Fink. The experiments are performed by exerting a force simultaneously with two parallel bars on the boundary of the phantom. Both bars have the same central frequency. The downward displacement, that is perpendicular to the phantom surface where

the two-bar force is applied, is measured in the plane equidistant between the two bars. The phantom contains a stiff cylindrical inclusion where the wave speed is twice that of the background. The cylinder has a radius 5 mm, and the axis of the cylinder is parallel to the boundary and perpendicular to the image plane. We present results from the experiment performed at 50, 60 and 70 Hz.

The units used here are different than those we use for the synthetic data experiments. Here, we use the units seconds for time, millimetres for length, and m s^{-1} for speed, when we describe our measured data recovery. In the imaging plane, the domain is $(0, 70.492) \times (-20.955, 20.955)$ and the inclusion for the high speed region is $B_p(5)$ where the centre point is $p = (36.746, 0)$. Before we can test the level curve method, we first have to calculate the arrival times of the wave, \hat{T} . The first step is to truncate the data in time, keeping only the data that contains the shear wave travelling through the image plane. This step is necessary because the faster compression wave travels through the image plane before the shear wave and there is some initial bulk motion of the phantom material due to the two-bar push. Also, after the shear wave travels through the image plane the data contains reflections from the sides of the phantom; see [28] for details. The arrival times are identified with the cross-correlation procedure described in section 2, with one small adjustment. Because the wave pulse decorrelates as it travels through the medium (see [28]), we experience split peaks in the cross-correlation function. This is problematic because, at different locations, which split peak is the largest may change, leading to discontinuities in the reconstructed arrival times. To make peak identification of the cross-correlation function easier, we again low pass filter the cross-correlation function, $C(x, \delta t)$. The arrival times are only calculated to the nearest time step, as we see no improvement in accuracy of the wave speed recovery by interpolating to a subgrid with real experimental data.

For illustration, we show the cross-correlation function at two different locations in the medium. The first location is near the source, and is shown in figure 7(A). The cross-correlation function has a clear maximum. In figure 7(B), we show the same cross-correlation function after applying a low pass filter. The low pass filtered cross-correlation function has broadened, implying a loss of resolution of the peak. This will lead to blurring in the final image. Unfortunately, this step is necessary because of the split peak problem discussed above. To illustrate we show the cross-correlation function at a point deep in the medium in figure 7(C). The peak of the cross-correlation function has split due to wave decorrelation. It is obviously quite difficult to find the arrival time from this cross-correlation function. The same cross-correlation function after applying a low pass filter is shown in figure 7(D). The low pass filtered cross-correlation function has a clear maximum.

The reconstructed arrival time surface is shown in figure 8(A). The slight stair stepping effect is because we have truncated the arrival times to the nearest time step. If one looks carefully at the arrival time plot, there is a small flat region centred at about (37, 0) indicating a high wave speed inclusion. The effects of the increased wave speed are more pronounced by looking at the level curves of the arrival time surface, shown in figure 8(B), which bend when travelling through the high-speed region which is circled.

Prior to implementing the level curve algorithms we follow the same preprocessing step given in [28]. We eliminate the small closed level curves in the arrival time data by keeping only the longest connected portion of any given level curve. We know *a priori* that these small closed curves are noise because the arrival times should be increasing in the normal direction. These eliminated curves are shown in red in figure 8(B). We note that this preprocessing step is $\mathcal{O}(m^{3/2})$ and is not yet as fast as the level curve method.

Before we show the results with experimental data, we explain a last step taken in our inverse problem. For laboratory data we generate an initial guess $(\sqrt{\mu/\rho})_0$ using the

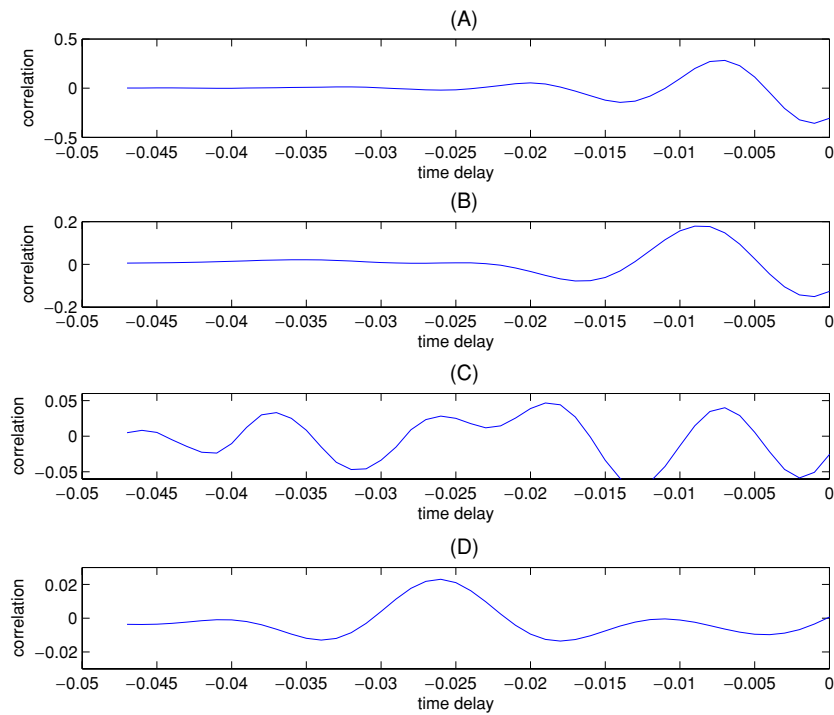


Figure 7. (A) Cross-correlation function at a point near the source; (B) low pass filtered cross-correlation function near the source; (C) cross-correlation function deep in the medium; (D) low pass filtered cross-correlation function deep in the medium.

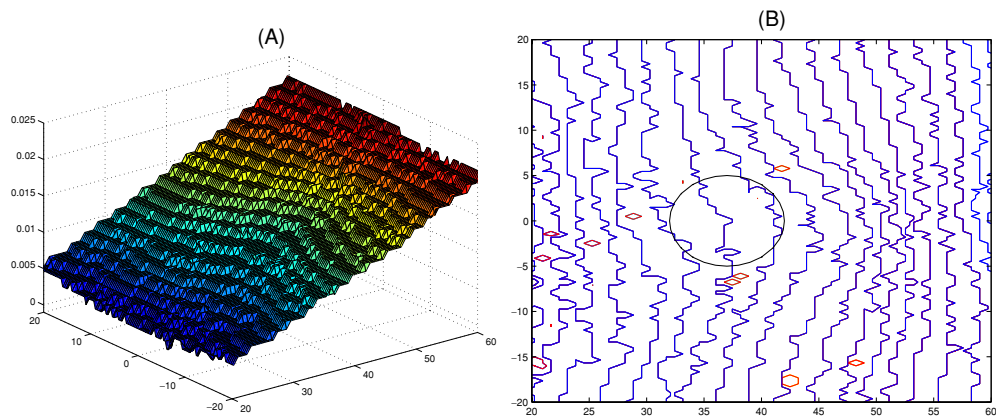


Figure 8. (A) Arrival times of 60 Hz experimental data using correlation; (B) level curves of the arrival time surface generated with correlation.

second-order distance or second-order level curve method. We then add a total variation minimization step; see [28] for details. The final wave speed reconstruction using the second-order level curve method is shown in figure 9(B). For comparison, we repeat these calculations with the second-order distance method, and display the final wave speed

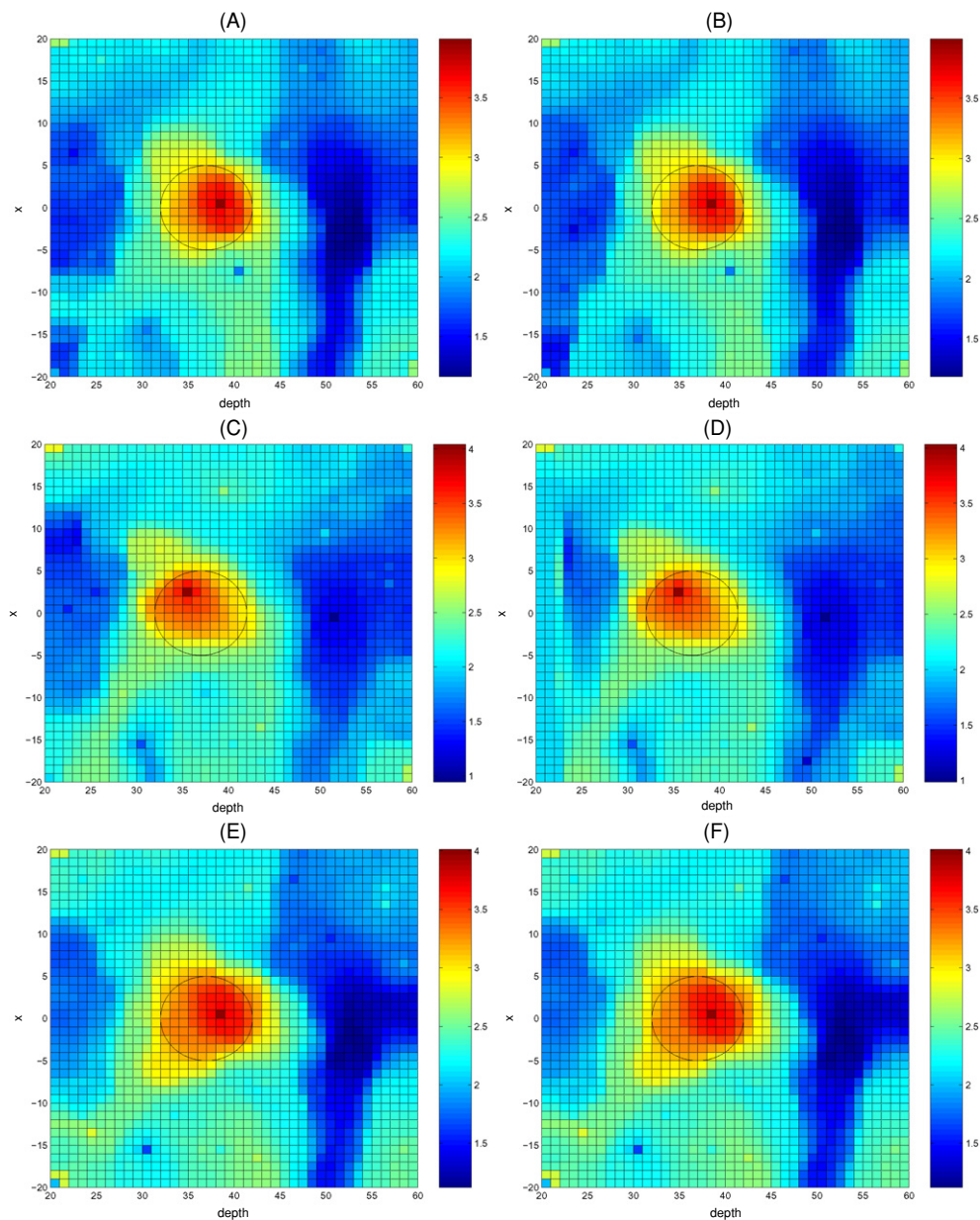


Figure 9. (A) Wave speed recovery for 50 Hz experimental data using the second-order distance method; (B) wave speed recovery for 50 Hz experimental data using the second-order level curve method; (C) wave speed recovery for 60 Hz experimental data using the second-order distance method; (D) wave speed recovery for 60 Hz experimental data using the second-order level curve method; (E) wave speed recovery for 70 Hz experimental data using the second-order distance method; (F) wave speed recovery for 70 Hz experimental data using the second-order level curve method; in all figures the wave speed is given in units of m s^{-1} .

reconstruction in figure 9(A). There is very little difference between the level curve and distance method calculations. As described in [28], both of these wave speed recoveries focus

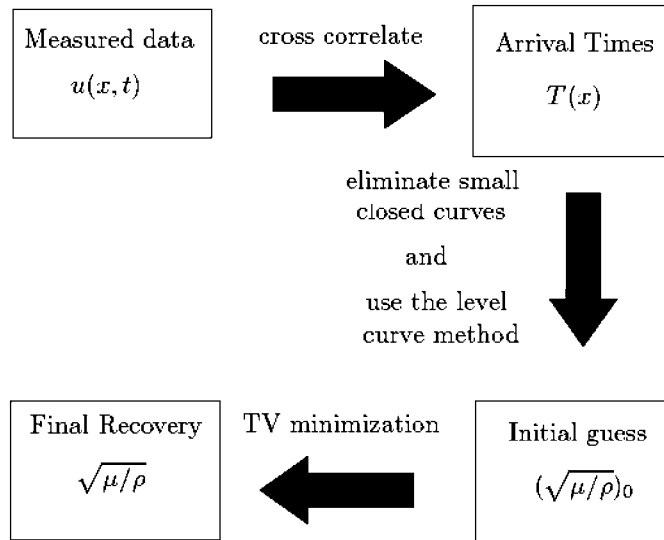


Figure 10. Flowchart for our complete algorithm to determine the shear wave speed $\sqrt{\mu/\rho}$ using the arrival time from the measured displacement data.

very well on the high speed region, but have two artefacts. There are low-speed artefacts both before and after the high speed inclusion. The artefact before the inclusion is very similar to that in the synthetic data recoveries, and is caused by the shear wave speed reflecting off of the inclusion. The artefact after the inclusion is caused by wave pulse decorrelating as it propagates through the medium, and in the future we will reduce or eliminate this artefact by correctly modelling the wave pulse as it propagates.

Now we do all of the above calculations with the 60 Hz and 70 Hz experiment. The final wave speed reconstructions using the second-order level curve method (distance method) is shown in figure 9((D) and (F)) and ((C) and (E)). As before, there is very little difference between the level curve and distance method calculations. Similar to the 50 Hz recoveries, the correlation recoveries focus very well on the high speed region, but have artefacts directly above and below the high speed region.

6. Conclusions and future work

We have presented algorithms for recovering shear wave speed when *interior* space and time-dependent displacement data for a single solution is given. This algorithm takes advantage of the propagating front, by using the Eikonal equation for the arrival times of the shear wave. To summarize, our complete algorithm to determine the shear wave speed, $\sqrt{\mu/\rho}$, by the level curve method using the arrival time from the measured displacement data u is (see figure 10 for the flowchart) as follows:

- (1) Choose arrival time $\hat{T}(x) = -\delta t_{\max}$ at each x to maximize the correlation between $u(x_{\text{ref}}, t)$ and $u(x, t - \delta t)$.
- (2) For each level set of $\hat{T}(x)$, eliminate occasional small closed level curves.
- (3) Generate initial guess $(\sqrt{\mu/\rho})_0$ using the second-order level curve method.
- (4) Minimize the total variation of $\sqrt{\mu/\rho}$.

In [28], we first developed a family of stable methods, which we named distance methods, to accomplish this wave speed inversion. In this paper, we use level sets to create a much faster family of inverse Eikonal solvers that we name level curve methods. We have shown numerically that the level curve method has the same nice stability properties as the distance method and have demonstrated that the running time for this improvement algorithm is $O(m)\log(m)$, where m is the number of grid points in space. The running time of the algorithm could be further improved to $O(m)$ by using an $O(m)$ Eikonal solver.

7. Future work

Here are several problems we plan to address in the future:

- (1) Our proof that the arrival times for the acoustic wave equation satisfy the Eikonal equation requires that the arrival times be Lipschitz continuous, and a result establishing the Lipschitz continuity is needed.
- (2) We will justify the front propagation algorithm for the isotropic linear equations of elasticity. So far it has been justified for the acoustic equation. We note that this algorithm has already been successfully applied to real measured data. The arguments for the acoustic wave equation are easily extended to show that the compression wave satisfies an Eikonal equation with the compression wave speed, but our focus is the front of the shear wave. Shear wave arrivals are no longer first arrivals of the full elastic wave so new techniques are needed. We plan on addressing this problem using microlocal techniques.
- (3) Additional terms need to be added to the model to account for visco-elastic effects such as damping, dispersion and hysteresis. We must establish how these properties affect the arrival times.
- (4) Theorems justifying the use of arrival times for the model that includes visco-elastic effects will have to be developed.
- (5) Currently we are only using one component of the displacement data. In Tanter *et al* [43], the axial and lateral displacements are measured. In addition, future experiments using an array of transducers may be used to measure a third component of the displacement [12]. In principle, using additional components of the displacement data should lead to a better wave speed reconstruction. However, only one component of the displacement is needed to obtain the arrival time surface. We do plan on developing a related algorithm that will take advantage of additional measured displacement data.
- (6) The regularization technique can be improved. It is important for the algorithm to be capable of capturing sharp changes in the shear sound speed. For this reason, we are concentrating on total variation based methods. The method we currently employ has two terms: a smoothing term and a data matching term. The data matching term matches to the shear wave speed recovered from the distance and level curve methods. However, the actual data for the inverse Eikonal equation are the arrival times. A total variation technique that minimizes the total variation of the wave speed, but matches the arrival time data should result in an improvement.
- (7) Currently we are using isotropic models, which assume that wave speed is independent of direction. In many kinds of tissue, especially tissue with fibres, like muscle, the wave speed depends quite heavily on direction. In this case an anisotropic model would be more appropriate. Using our current method, we can only determine the speed orthogonal to the wave front. We plan on developing a related technique that uses multiple experiments to determine all the parameters that govern the directionally dependent shear wave speed.

- (8) The wave pulse decorrelates as it propagates through the medium. This causes the peak of the correlation function to split. We will develop an improved cross-correlation method to find the arrival times by modelling the wave pulse as it travels through the medium.

Acknowledgments

We have benefitted from discussions with J R Yoon, L Ji, A Maniatty, E Park, P Barbone, S Catheline, M de Hoop, M Fink, B Isakov, S Kurylev, M Lassas, C Nolan, S Osher, J Sethian, M Sini, M Tanter and D Tataru. We would like to thank members of M Fink's laboratory for sharing the measured data. JM is partially supported by NSF Focus Research Group Grant No DMS 0101458, ONR Grant No N000 14-96-1-0349 and DR is partially supported by NSF VIGRE Grant No DMS 9983646.

References

- [1] Adalsteinsson D and Sethian J A 1995 A fast level set method for propagating interfaces *J. Comput. Phys.* **118** 269–77
- [2] Barbone P and Nachiket G 2004 Elastic modulus imaging: on the uniqueness and nonuniqueness of the elastography inverse problem in two dimensions *Inverse Problems* **20** 283–96
- [3] Barbone P E and Bamber J C 2002 Quantitative elasticity imaging: what can and cannot be inferred from strain images *Phys. Med. Biol.* **47** 2147–64
- [4] Bercoff J, Tanter M, Sandrin L, Catheline S and Fink M 2001 Ultrafast compound imaging for 2D displacement vector measurements: application to transient elastography and color flow mapping *IEEE Ultrason. Symp. Proc.: An Int. Symp.* vol 2 pp 1619–22
- [5] Bercoff J, Tanter M, Chaffai S, Sandrin L and Fink M 2002 Ultrafast imaging of beam formed shear waves induced by the acoustic radiation force: application to transient elastography *IEEE Ultrason. Symp. Proc.* **2** 1899–902
- [6] Braun J, Buntkowsky G, Bernarding J, Tolxdorff T and Sack I 2001 Simulation and analysis of magnetic resonance elastography wave images using coupled harmonic oscillators and Gaussian local frequency estimation *Magn. Reson. Imaging* **19** 703–13
- [7] Catheline S, Thomas J-L, Wu F and Fink M 1999 Diffraction field of a low frequency vibrator in soft tissues using transient elastography *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **46** 1013–19
- [8] Chopp D L 2001 Some improvements of the fast marching method *SIAM J. Sci. Comput.* **23** 230–44
- [9] Peng D, Merriman B, Osher S, Zhao H and Kang M 1999 A PDE-based fast local level set method *J. Comput. Phys.* **155** 410–38
- [10] Dular P and Geuzaine C *GetDP: A General Environment for the Treatment of Discrete Problems* <http://www.geuz.org/getdp>
- [11] Evans L C and Garipey R F 1992 *Measure Theory and Fine Properties of Functions* (Boca Raton, FL: CRC Press)
- [12] Fink M Private conversation
- [13] Gao L, Parker K J and Alam S K 1995 Sonoelasticity imaging: theory and experimental verification *J. Acoust. Soc. Am.* **97** 3875–80
- [14] Gennisson J L, Catheline S, Chaffai S and Fink M 2003 Transient elastography in anisotropic medium: application to the measurement of slow and fast shear wave speeds in muscles *J. Acoust. Soc. Am.* **114** 536–41
- [15] Geuzaine C and Remacle J-F Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities <http://www.geuz.org/gmsh/>
- [16] Greenleaf J F, Muthupillai R, Rossman P J, Smith J, Manduca A and Ehman R L 1996 Direct visualization of strain waves by magnetic resonance elastography (MRE) *IEEE Ultrason. Symp. Proc.* **1** 467–72
- [17] Ji L and McLaughlin J R 2002 Using a Hankel function expansion to identify stiffness for the boundary impulse input experiment *AMS Contemporary Mathematics (CONM) Book Series: Proc. Conf. on Inverse Problems and Applications (Pisa, Italy)* ed G Alessandrini and G Uhlmann
- [18] Ji L and McLaughlin J R 2003 Recovery of the Lamé parameter μ in biological tissues *Inverse Problems* **20** 1–24
- [19] Ji L and McLaughlin J R 2003 Shear stiffness identification in biological tissues: the full elastic model (draft)

- [20] Ji L, McLaughlin J R, Renzi D and Yoon J-R 2003 Interior elastodynamics inverse problems: shear wave speed reconstruction in transient elastography *Inverse Problems* **19** S1–S29
- [21] Konafagou E E, Harrigan T and Ophir J 2000 Shear strain estimation and lesion mobility assessment in elastography *Ultrasonics* **38** 400–04
- [22] Konafagou E E and Ophir J 1998 A new elastographic method for estimation and imaging of lateral displacements, lateral strains, corrected axial strains and Poisson's ratios in tissues *Ultrasound Med. Biol.* **24** 1183–99
- [23] Konafagou E E 2000 Precision estimation and imaging of normal and shear components of the 3D strain tensor in elastography *Phys. Med. Biol.* **45** 1553–63
- [24] Kruse S A, Smith J A, Lawrence A J, Dresner M A, Manduca A, Greenleaf J F and Ehman R L 2000 Tissue characterization using magnetic resonance elastography: preliminary results *Phys. Med. Biol.* **45** 1579–90
- [25] Manduca A, Oliphant T E, Dresner M A, Mahowald J L, Kruse S A, Amromin E, Felmlee J P, Greenleaf J F and Ehman R L 2001 Magnetic resonance elastography: non-invasive mapping of tissue elasticity *Med. Image Anal.* **5** 237–54
- [26] Manduca A, Oliphant T E, Lake D S, Dresner M A and Ehman R L 2002 Characterization and evaluation of inversion algorithms for MR elastography *Proc. SPIE—Int. Soc. Opt. Eng.* **4684** (pt 1–3) 1180–5
- [27] Manduca A, Lake D S and Ehman R L 2003 Improved inversion of MR elastography images by spatio-temporal directional filtering *Proc. SPIE—Int. Soc. Opt. Eng.* **5032** 445–52
- [28] McLaughlin J and Renzi D 2006 Shear wave speed recovery in transient elastography and supersonic imaging using propagating fronts *Inverse Problems* **22** 681–706
- [29] McLaughlin J R and Yoon J R 2004 Unique identifiability of elastic parameters from time dependent interior displacement measurement *Inverse Problems* **20** 25–45
- [30] Muthupillari R, Lomas D J, Rossmann P J, Greenleaf J F, Manduca A and Ehman R L 1995 Magnetic resonance elastography by direct visualization of propagating acoustic strain wave *Science* **269** 1854–57
- [31] Nightingale K R, Soo M S, Nightingale R W and Trahey G E 2001 Investigation of real-time remote palpation imaging *Proc. SPIE—Int. Soc. Opt. Eng.* **4325** 113–19
- [32] Nightingale K R, Palmeri M L, Nightingale R W and Trahey G E 2001 On the feasibility of remote palpation using acoustic radiation force *J. Acoust. Soc. Am.* **110** 625–34
- [33] Fahey B J, Nightingale K R, McAleavey S A, Palmeri M L, Wolf P D and Trahey G E 2005 Acoustic radiation force impulse imaging of myocardial radiofrequency ablation: initial in vivo results *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **52** 631–41
- [34] Oberai A A, Gokhale N H and Feijoo G R 2003 Solution of inverse problems in elasticity imaging using the adjoint method *Inverse Problems* **19** 297–313
- [35] Ophir J, Cespedes I, Ponnekanti H, Yazdi Y and Li X 1991 Elastography: a quantitative method for imaging the elasticity of biological tissues *Ultrason. Imaging* **13** 111–34
- [36] Osher S J and Fedkiw R 2002 *Level Set Methods and Dynamic Implicit Surfaces* (Berlin: Springer)
- [37] Osher S J and Sethian J A 1988 Front propagation with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations *J. Comput. Phys.* **79** 12–49
- [38] Renzi D 2004 Interior elastodynamics inverse problems: shear wave speed recovery in transient elastography using level set based inversion of arrival times *PhD Dissertation* Department of Mathematics, Rensselaer Polytechnic Institute, Troy, NY
- [39] Sandrin L, Tanter M, Catheline S and Fink M 2002 Shear modulus imaging with 2-D transient elastography *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **49** 426–35
- [40] Sandrin L, Tanter M, Gennisson J L, Catheline S and Fink M 2001 Shear elasticity probe for soft tissues with 1-D transient elastography *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **49** 436–46
- [41] Sandrin L, Tanter M, Catheline S and Fink M 2002 Time-resolved 2D pulsed elastography: experiments on tissue-equivalent phantoms and breast *in vivo* *Proc. SPIE—Int. Soc. Opt. Eng.* **4325** 120–6
- [42] Sethian J A 1999 *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science* (Cambridge: Cambridge University Press)
- [43] Tanter M, Bercoff J, Sandrin L and Fink M 2002 Ultrafast compound imaging for 2-D motion vector estimation: application to transient elastography *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **49** 1363–74
- [44] Taylor L S, Porter B C, Rubens D J and Parker K J 2000 Three-dimensional sonoelastography: principles and practices *Phys. Med. Biol.* **45** 1477–94
- [45] Wu Z, Taylor L S, Rubens D J and Parker K J 2002 Shear wave focusing for three-dimensional sonoelastography *J. Acoust. Soc. Am.* **111** 439–46