Shape optimization using the FWI objective function for salt body segmentation

Taylor Dahlke, Biondo Biondi, and Robert Clapp

ABSTRACT

Level set methods can provide a sharp interpretation of the salt body by defining the boundary as an isocontour of a higher dimensional implicit representation, and then evolving that surface to minimize the Full Waveform Inversion (FWI) objective function. We can take advantage of the fact that the implicit surface update gradient is based on the tomographic update gradient, and utilize it to update the background velocity concurrently with the salt boundary. Using this approach on synthetic examples, we can achieve reasonable convergence both in terms of the residual L2 norm, as well as the evolution of the salt boundary and background velocity towards the true model.

INTRODUCTION

Oil producing regions like the Gulf of Mexico and offshore western Africa are known to have geologically complex salt body formations which can cause difficulties in producing seismic imagery. The velocity of these salt bodies contrasts sharply with that of the background sediment layers. An inaccurate interpretation of the salt boundaries can cause significant errors in the velocity estimation process, because the formations themselves can act as lenses which focus or disperse seismic energy, influencing tomography. This can subsequently impact the imaging results that rely on accurate velocity models. Salt bodies can act as seals trapping hydrocarbons underneath, which are often the targets of such imaging projects. For this reason the interpretation of salt body boundaries can also impact drilling and production activities.

Review

Tomographic approaches to interpreting salt bodies can be less than effective, because the results tend to be too smooth to provide significantly accurate placement of the salt boundaries. Manual and semi-automatic picking of salt boundaries is a common approach to interpreting the desired sharp delineations, but these methods can be time-consuming and tedious since expert input is necessary for either the actual picking, or the oversight and correction. Furthermore, once a model has been

produced, it must be used to generate an image, and then be refined as necessary. A robust method for further automating the salt interpretation procedure would greatly alleviate this bottleneck.

Some previous approaches to performing salt body segmentation use a shape optimization approach for identifying salt body boundaries (Guo and de Hoop (2013), Lewis et al. (2012)). The boundaries of a salt body can be represented as the zero-isocontour of a higher dimensional surface (for example, a 2D boundary as a contour of a 3D surface). A gradient can be derived to evolve this shape / isosurface according to the FWI objective function. Unlike the smooth boundaries produced by tomographic approaches, the isocontour resulting from the shape optimization provides a sharp boundary, which is a more appropriate way to classify most salt-sediment interfaces. Guo and de Hoop (2013) utilize this approach using a frequency domain forward wave operator to evolve a salt boundary and velocity model. However, their approach alternates between updating the background velocity and salt body boundary, which effectively requires twice as many iterations as performing both updates concurrently.

Proposed solution

We propose using a shape optimization approach, with the use of time domain forward wave-propagation. By using time domain forward wave-propagation, we take advantage of using a continuous range of frequencies (rather than discrete frequencies) in each iteration, which allows for sharper delineation of the boundary. Further, we take advantage of the fact that our boundary update gradient is based on the tomographic update gradient, and make updates to both concurrently with the use of scaling parameters. In theory, this approach has the potential to be more efficient than an alternating update approach.

Agenda

In this paper we will begin by discussing the fundamentals of the level set method and its key properties, followed by the derivation of the boundary update gradient. Afterwards, we will demonstrate the application of the boundary-tomography update method on a simple test model. Last, we will discuss the assumptions, limitations, and advantages of the method, as well as future directions for this research.

MATHEMATICAL CONCEPTS

We begin with a brief overview of the level set method and how we apply the evolution scheme it utilizes. The derivation for the shape optimization implementation follows.

Level set fundamentals

In our problem, we are trying to determine the boundary of a two dimensional body. Instead of using an algorithm that operates in this 2D plane directly, we use the level set algorithm which evolves a 3D implicit surface, ϕ. While our algorithm acts directly on this surface instead of the boundary, our solution for the 2D boundary is simply represented by a contour "slice" of this implicit surface where $\phi = 0$, as described in Osher and Sethian (1988) and Burger (2003). While it may seem counterintuitive to add extra dimensionality to our problem, by doing so we gain some advantages. These include the ability to merge and separate bodies as the level set evolution proceeds, as well as the ability to handle sharp corners and cusps in the lower-dimensional (2D) plane that the boundary exists on.

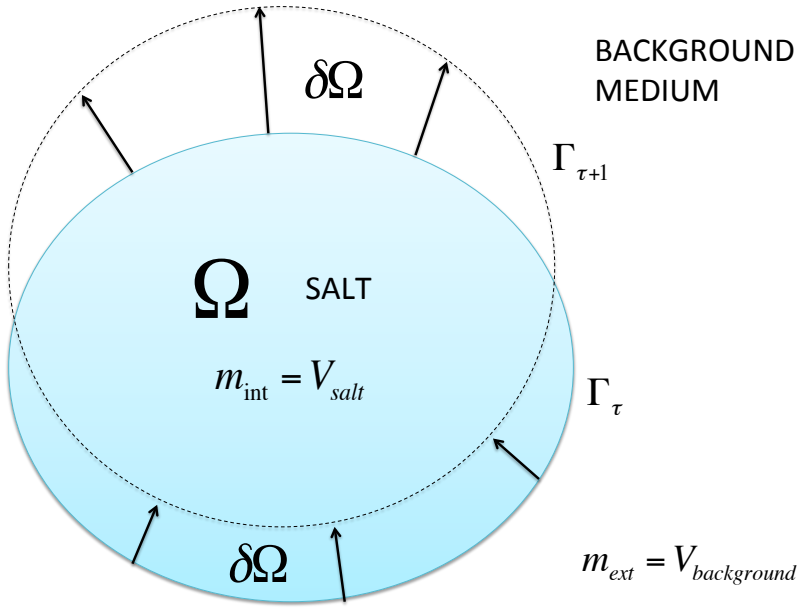

Figure 1: Diagram of domain partitioning. The full inclusive domain is Θ. [**NR**]

Based on this concept, we define a spatial domain $\Theta \subset \mathbb{R}^2$, a (salt) body $\Omega \subset \Theta$, and the salt body boundary Γ such that:

$$\Omega = \{x \mid \phi(x, \tau) > 0\}, \qquad \Gamma = \{x \mid \phi(x, \tau) = 0\}$$

where τ indicates the axis along which the evolution steps progress ($\tau = 0$ is the initial iteration). As such, for a single step along τ, our salt body Ω evolves to Ω' . We define a point along the boundary curve to be:

$x_\Gamma = \{x \in \Gamma\}$.

With this definition of the boundary points, the level set of ϕ that represents the salt body boundary can be described as:

$\phi(x_\Gamma, \tau) = 0$.

By taking the derivative of this equation, the chain rule gives us

$$\frac{\partial \phi}{\partial \tau} + \frac{\partial \phi}{\partial x_\Gamma}\frac{\partial x_\Gamma}{\partial \tau} = 0. \tag{1}$$

This equation can be readily rewritten as:

$$\frac{\partial \phi}{\partial \tau} + \vec{\nabla}\phi \cdot \vec{v}(x_\Gamma, \tau) = 0. \tag{2}$$

We can use $\vec{\nabla}\phi$ defined over all the full domain of x (rather than just x_Γ) since $\vec{\nabla}\phi \cdot \vec{v}(x_\Gamma, \tau)$ is a dot product, and only the terms where $x \in \Gamma$ will contribute to the overall dot product result. This "velocity" term in equation 2 can be defined as having both a "speed" and a normal vector component, $\vec{v}(x_\Gamma, \tau) = V(x_\Gamma, \tau)\vec{n}(x_\Gamma, \tau)$. In complete form there is also a tangential component, but we ignore this part since it doesn't contribute to a change in the surface ϕ.

We know the normal vector is defined as

$\vec{n}(x_\Gamma, \tau) = \frac{\vec{\nabla}\phi(x_\Gamma, \tau)}{|\vec{\nabla}\phi(x_\Gamma, \tau)|}$,

which allows us to restate equation 2 in a more familiar representation:

$$\frac{\partial \phi}{\partial \tau} = -V(x_\Gamma, \tau)\,|\nabla \phi|\,. \tag{3}$$

The scalar speed term $V(x_\Gamma, \tau)$ describes the magnitude of the variation of ϕ that is normal to the boundary Γ. It determines the evolution of the implicit surface, and ultimately the boundary implied by it. The following section describes the derivation of this normal velocity such that the FWI objective function is minimized.

Derivation of the evolution equation

Calculus of variations

To begin, we derive the shape derivative using a formal calculus of variations approach, following the derivation structure outlined in Santosa (1996). The objective is to define the variation of the model m with respect to the variation in the implicit surface, ϕ. The variation in m will only occur between the boundaries of Ω_τ and $\Omega_{\tau+1}$. We define this region $\Omega_\tau \cap \Omega_{\tau+1}$ as $\partial\Omega$.

Let us consider the model parameter variation $m+\delta m$, where the model is binary ($m = m_{int} = V_{salt}$ or $m = m_{ext} = V_{back}$). In the case where the normal vector points outwards from the salt body at a point x_Γ, then an advance of δx_Γ will change the value of $m(x_\Gamma + \delta x_\Gamma)$ from m_{ext} to m_{int} (see figure 2). Therefore, δm at these points can be shown as $\delta m(x_\Gamma + \delta x_\Gamma) = m_{\text{int}} - m_{\text{ext}}$.

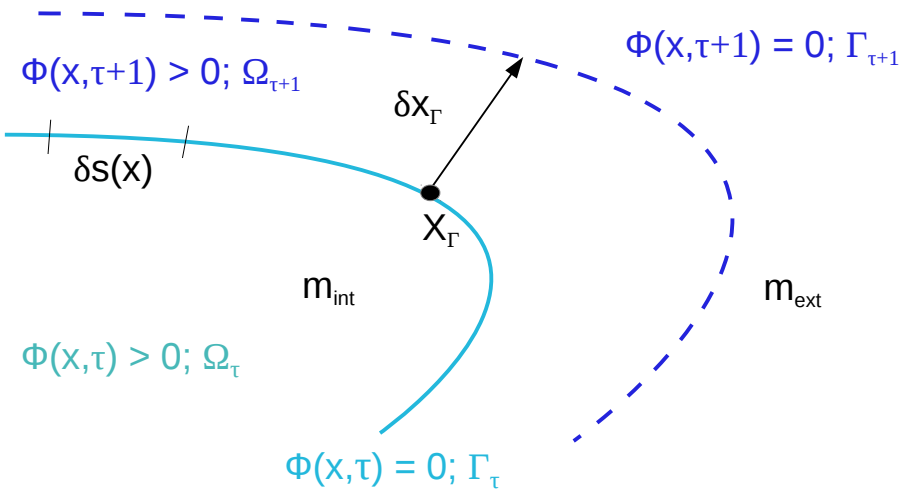

Figure 2: The geometry of the curve $\{x_\Gamma : \phi = 0\}$ for a variation $\delta\phi(x)$ along the evolution axis τ. [**NR**]

Consider an inner product of δm with a test function $f(x)$. Formally, this can be written as,

$$\langle \delta m, f(x) \rangle = \int_{\mathbb{R}^2} \delta m(x) f(x) dx = \int_{\partial\Omega} \delta m(x) f(x) dx. \tag{4}$$

Since the $\delta m(x)$ term equals zero in $\mathbb{R}^2 \setminus \partial\Omega$, it doesn't contribute to the overall inner product when integrating over that domain. For this reason we can justify ignoring this domain region, and only integrate over $\partial\Omega$ where $\delta m(x)$ is non-zero. We know that $\delta m(x)$ will be $\pm(m_{\text{int}} - m_{\text{ext}})$, depending on the relative values of m_{int} and m_{ext} or the direction of the normal vector $\vec{n}$. We only care about the component of δx_Γ that occurs in the normal direction, since a tangential variation of x_Γ doesn't affect

m or ϕ. Furthermore, since δx_Γ is infinitesimal, we can replace dx with $\delta\vec{x_\Gamma} \cdot \vec{n}$ and simplify equation 4 into

$$\langle \delta m, f(x) \rangle = \int_{\partial\Omega} (m_{\text{int}} - m_{\text{ext}}) \delta\vec{x_\Gamma} \cdot \vec{n} f(x) ds(x), \tag{5}$$

where $ds(x)$ is the incremental arc length along the boundary Γ. We can think of $\delta\vec{x_\Gamma} \cdot \vec{n} ds(x)$ as roughly the incremental area over which m varies at x.

We can identify δm from this equation 5. It can be considered a measure over $\partial\Omega$:

$$\delta m = (m_{\text{int}} - m_{\text{ext}}) \delta\vec{x_\Gamma} \cdot \vec{n} \mid_{x \in \partial\Omega}. \tag{6}$$

We will use δm in the shape derivative formulation that we derive next.

Shape derivative formulation

The objective function that we wish to minimize is the full-waveform inversion least-squares objective function:

$$F(m) \doteq \frac{1}{2} \| A(m) - d \|_2^2. \tag{7}$$

In order to minimize this function with each step along τ, we need to derive a solution such that $\frac{\partial F}{\partial \tau} < 0$. $\frac{\partial F}{\partial \tau}$ can be naturally decomposed using derivatives:

$$\frac{\partial F}{\partial \tau} = \frac{\partial F}{\partial m} \frac{\partial m}{\partial \tau} \tag{8}$$

$$\frac{\partial F}{\partial \tau} \triangle \tau = \frac{\partial F}{\partial m} \frac{\partial m}{\partial \tau} \triangle \tau \tag{9}$$

$$\delta F(m) = \frac{\partial F}{\partial m} \delta m. \tag{10}$$

We can make use of the inner-product operator to re-write equation 10 and determine the objective function derivative ($\delta F(m)$) in the direction of δm:

$$\delta F(m) = \left\langle \frac{\partial F}{\partial m}, \delta m \right\rangle. \tag{11}$$

In the following section we will show how the solution to $\frac{\partial F}{\partial m}$ is derived using the adjoint state method.

We remember that in the previous section we stated the goal of this derivation as being a solution of the scalar velocity function $V(x_\Gamma, \tau)$, such that the objective function is minimized. We recognize that the normal component of the variation δx_Γ satisfies:

$$\delta\vec{x}_\Gamma \cdot \vec{n} = V(x_\Gamma, \tau). \tag{12}$$

As a result, we can substitute equation 12 into equation 6 from earlier and get

$$\delta m = (m_{\text{int}} - m_{\text{ext}})V(x, \tau) \mid_{x\in\partial\Omega}. \tag{13}$$

Further, we can use the result in equation 13 and apply it to the integral form of the inner product described in equation 11:

$$\delta F(m) = \int_{\partial\Omega} \frac{\partial F}{\partial m}(m_{\text{int}} - m_{\text{ext}})V(x, \tau)ds(x). \tag{14}$$

In the previous section, $f(x)$ was used as a test function in the inner product example. In this instance, $V(x_\Gamma, \tau)$ can be considered the test function that we are trying to solve for, such that equation 14 is true. Since $\delta F(m)$ must be negative in order to minimize the objective function described in equation 7, we choose $V(x_\Gamma, \tau)$ such that it is:

$$V(x, \tau) \mid_{x\in\partial\Omega} = -(m_{\text{int}} - m_{\text{ext}})\frac{\partial F}{\partial m} \mid_{x\in\partial\Omega}. \tag{15}$$

We keep in mind that any $V(x, \tau)$ that satisfies equation 15 will produce a $\delta\phi$ that reduces $F(m)$, and as a result we choose the same equation defined over a larger space,

$$V(x, \tau) \mid_{x\in\Theta} = -(m_{\text{int}} - m_{\text{ext}})\frac{\partial F}{\partial m} \mid_{x\in\Theta}. \tag{16}$$

where Θ is the entire domain of x. This can be justified since $\partial\Omega$ is a subset of Θ, so the inner product result will not change. We can combine this result into our formulation of the level set update equation from earlier (equation 3) to get a final derivation of the levelset evolution equation:

$$\frac{\partial\phi}{\partial\tau} = (m_{\text{int}} - m_{\text{ext}})\frac{\partial F}{\partial m}\left|\vec{\nabla}\phi\right|. \tag{17}$$

In the following section we describe how the adjoint state method is used to derive $\frac{\partial F}{\partial m}$, which can be shown to be equivalent to $J(m)^T(A(m) - d)$. Since our case uses the FWI objective function (equation 7), this term can be interpreted as least squares migration, more specifically as reverse time migration.

Least squares migration term from adjoint state method

The adjoint state method is an approach that allows us to find the derivative of a functional that is subject to constraining equations without having to compute the (computationally expensive) Fréchet derivative term. In our case, the functional is the FWI objective function (based on data space misfit):

$$J(m) = \tfrac{1}{2} \left\| P(S(m))f - d_f \right\|^2_{L^2(\Sigma)} .$$

Here, T is the recording time, $S_{s,r}$ is the restriction operator onto the receiver positions (which depends on spatial coordinates). The model parameter is the squared slowness, $m = \sigma^2$. Our constraining equations are the forward time-domain wave propagation boundary condition equations,

$$\begin{cases} \left(m(x)\dfrac{\partial^2}{\partial t^2} - \Delta \right) a_s = f_s & \text{if } x = \Theta \setminus \bar{\Omega} \\ \left(m(x)\dfrac{\partial^2}{\partial t^2} - \Delta \right) a_s = 0 & \text{if } x = \Omega \\ \dfrac{\partial a_s}{\partial x} = 0 & \text{if } t = 0 \\ a_s = 0 & \text{if } t = 0 \end{cases}$$

The solution to this set of equations is the forward propagated wavefield, a. The solution of a also solves the following variational equation, which we use as our constraining equation, substituting for the set of equations above:

$$b(\Omega; a, w) - s(w) = 0$$

where

$$b(\Omega; a, w) = \int_{\Theta\setminus\bar{\Omega}} \int_0^T (\nabla a \cdot \nabla w - m(x)\frac{\partial^2 a}{\partial^2 t}\bar{w})dtdx + \int_{\Omega} \int_0^T (\nabla a \cdot \nabla w - m(x)\frac{\partial^2 a}{\partial^2 t}\bar{w})dtdx \tag{18}$$

and

$$s(w) = \int_{\Theta\setminus\bar{D}} \int_0^T f\bar{w}dtdx. \tag{19}$$

The first step in the adjoint state method is to create an augmented Lagrangian functional that combines both the objective and constraining functions. We build this functional using the Lagrange multiplier w:

$$L(\Omega_\tau; a, w) = \tfrac{1}{2} \left\| P(a - d_f) \right\|^2_{L^2(\Sigma)} + b(\Omega_\tau; a, w) - s(w) .$$

If $a = u_\tau$ is the solution to variational (constraining) equation 18, then:

$$L(\Omega_\tau; u_\tau, w) = \tfrac{1}{2} \left\| P(u_\tau - d_f) \right\|^2_{L^2(\Sigma)} .$$

Taking the derivative of this augmented Lagrangian functional gives us:

$$\begin{aligned} \frac{\partial}{\partial \tau} L(\Omega_\tau; u_\tau, w)\Big|_{\tau=0} &= \int_\Sigma (P^* P(S(\Omega_\tau) f - d_f) \frac{\partial}{\partial \tau} S(\Omega_\tau) f \Big|_{\tau=0} d\sigma \\ &+ b(\Omega; \frac{\partial}{\partial \tau} S(\Omega_\tau) f \Big|_{\tau=0}, w) \\ &+ \frac{\partial}{\partial \tau} b(\Omega_\tau; u_\tau, w)\Big|_{\tau=0} . \end{aligned} \tag{20}$$

If w solves:

$$b(\Omega_\tau; a, w) = -\int_\Sigma (P^* P(S(\Omega_\tau) f - d_f) a d\sigma \tag{21}$$

then equation 20 simplifies to:

$$\frac{\partial}{\partial \tau} L(\Omega_\tau; u_\tau, w)\Big|_{\tau=0} = \frac{\partial}{\partial \tau} b(\Omega_\tau; u_\tau, w)\Big|_{\tau=0} . \tag{22}$$

We can write equation 21 in the form of a direct problem, where $\bar{w}$ is the adjoint state,

$$b(\Omega_\tau; \bar{w}, u_\tau) = \textstyle\int_\Sigma (P^* P(S(\Omega_\tau) f - d_f) \bar{u}_\tau d\sigma$$

and $\bar{w}$ is subsequently the solution to:

$$\begin{cases} (m(x) \frac{\partial^2}{\partial t^2} - \Delta) w_s = (P^* P(S(\Omega_\tau) f - d_f) \delta_\Sigma & \text{if } x = \Theta \setminus \bar{\Omega} \\ (m(x) \frac{\partial^2}{\partial t^2} - \Delta) w_s = 0 & \text{if } x = \Omega \\ \frac{\partial w_s}{\partial x} = 0 & \text{if } t = T \\ w_s = 0 & \text{if } t = T \end{cases} \tag{23}$$

We can perform a change of variables such that $h(t) = w(T - t)$. This transformation changes our adjoint state system of equations to read:

$$\begin{cases} (m(x)\frac{\partial^2}{\partial t^2} - \Delta)h_s = (P^* P(S(\Omega_\tau)f - d_f)\delta_\Sigma & \text{if } x = \Theta \setminus \bar{\Omega} \\ (m(x)\frac{\partial^2}{\partial t^2} - \Delta)h_s = 0 & \text{if } x = \Omega \\ \frac{\partial h_s}{\partial x} = 0 & \text{if } t = 0 \\ h_s = 0 & \text{if } t = 0 \end{cases} \tag{24}$$

This makes h_s the reverse time propagation of the residual wavefield at shot s. We complete the derivative described in equation 22, which simplifies to the following:

$$\partial J(m) = -\sum_s \int_0^T \int_{x\in\Gamma} h_s(x,t)\frac{\partial^2 u_s(x,t)}{\partial t^2} d\sigma dt. \tag{25}$$

RESULTS

My demonstration of the shape optimization algorithm was performed on a 2D model, with the implicit surface evolved being a 3D surface. For the forward wave propagation, a wavelet with a 15.0 Hz central frequency was propagated using a time domain forward operator, so that continuous frequency information would be forward modeled in a single iteration.

Shape optimization evolution

We begin with an initial background velocity, and a signed distance function as the initial implicit surface ϕ. Since we assume a constant salt velocity, we use both of these inputs to create a full initial-guess velocity model (m_o). Using this m_o, we forward model to get our d_{syn} which we use to get a residual. The residual is used to calculate both a tomographic and a boundary update gradient, as described in the derivation section previously. We then perform adjoint operations on these gradients so that we can do a linear plane search (in residual space) for the scaling parameters, that we then rescale for use in the forward Euler scheme that updates the implicit surface (ϕ) and the background velocity V_{back}:

$$\phi^{j+1} = \phi^j + \beta\frac{\partial\phi}{\partial j} \tag{26}$$

$$V_{back}^{j+1} = V_{back}^j + \alpha\frac{\partial V_{back}}{\partial j}. \tag{27}$$

where β and α are the step sizes and j is the current iteration point. As the implicit surface is evolved, it is important to maintain stability of the evolution. One relevant

aspect of maintaining stability is keeping the implicit surface update step size (β) small enough to satisfy the Courant-Friedrich's-Levy (CFL) condition, which is stated by Chaudhury and Ramakrishnan (2007) (when applied to level set evolution) as being:

$$G_{max} \cdot \beta \leq min(h_x, h_y) \tag{28}$$

where h_x and h_y are the grid spacing in the x and y directions, and G_{max} is the maximum value of the update gradient. While later we describe how a plane search is used to determine the scaling parameters α and β, my program adjusts these scaling parameters (while maintaining their ratio) when necessary to satisfy the constraint in equation 28 when β is higher than the limit.

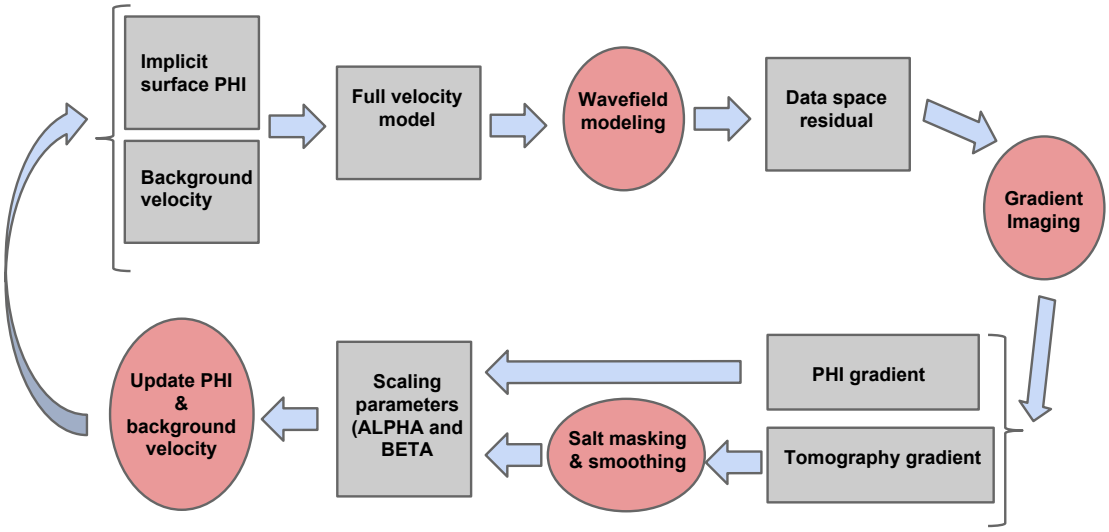

Figure 3: The work flow used for shape optimization. [**NR**]

One additional measure taken to ensure that the evolution is stable is the use of a regularization term that is scaled and added to the boundary gradient before each update is applied. In this case, a distance regularization term was used which drives the shape of the implicit surface towards that of a signed-distance function. When irregularities begin to occur in the implicit surface during level set evolution, numerical errors start to occur which can lead to instability. By driving the gradient of the implicit surface to equal one, we minimize irregularities and are able to continue evolution without having to reinitialize a signed-distance function to the salt boundary contour. An excellent reference on this type of regularization is the paper by Li et al. (2010) regarding the implementation of distance regularized level set evolution.

We apply our algorithm on a simple velocity model, using an acquisition geometry of 17 shots spaced 80 [m] apart, and 65 receivers spaced 20 [m] apart. In the example

shown in Fig. 4, the initial and true background velocity models matched. A bottom reflector and positive velocity gradient was used to help provide better illumination along the bottom and flanks of the circular salt body, which had a velocity of 4500 [m/s]. As hoped for, even while background velocity updates were applied each iteration as described, the final model resulted in virtually zero change to the background model. This is due to the scaling parameter search correctly weighting the α parameter such that successive updates would cancel each other out. More importantly, the boundary of the salt body expanded during evolution to very closely match the true salt body boundary.

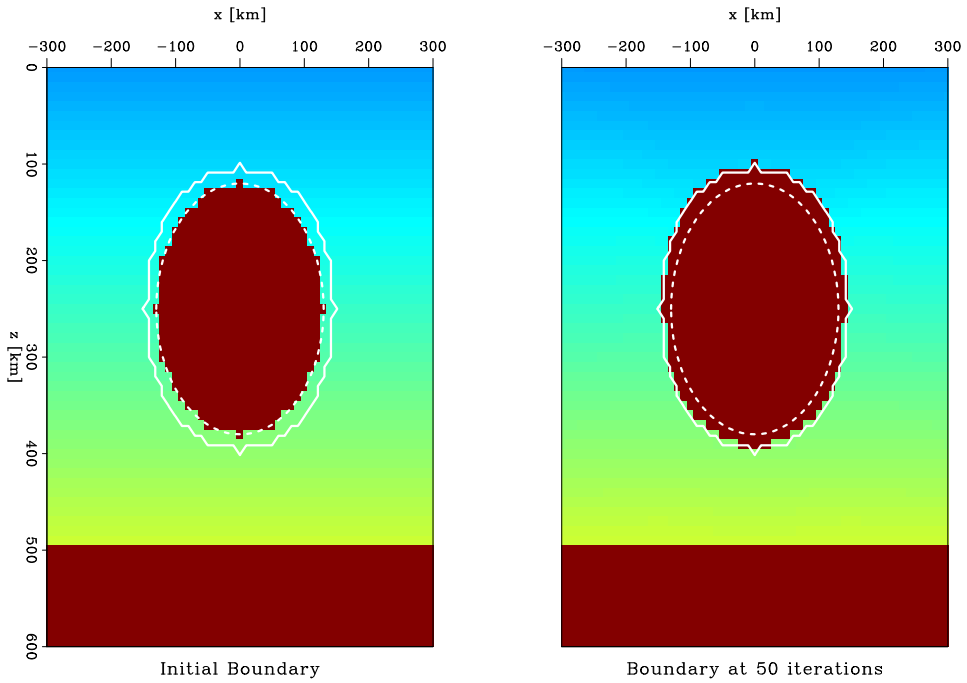

Figure 4: Initial velocity model (left) and final velocity model after 50 iterations (right). True model boundary indicated (solid line); Initial boundary guess (dashed line). [**CR**]

DISCUSSION

No realistic scenario uses a binary velocity model, and any salt interpretation also relies on an accurate determination of the background velocity. Guo and de Hoop (2013) describe an approach to performing general tomographic updates by using an alternating method of updating the background velocity and the salt body boundary. These steps alternate between freezing the background velocity and making a salt boundary update, and then freezing the salt body boundary, smoothing the entire velocity model, and then calculating an update gradient based on this. The smoothing mitigates any reflectivity caused by sharp velocity transitions. While this approach can be effective, it is also slow.

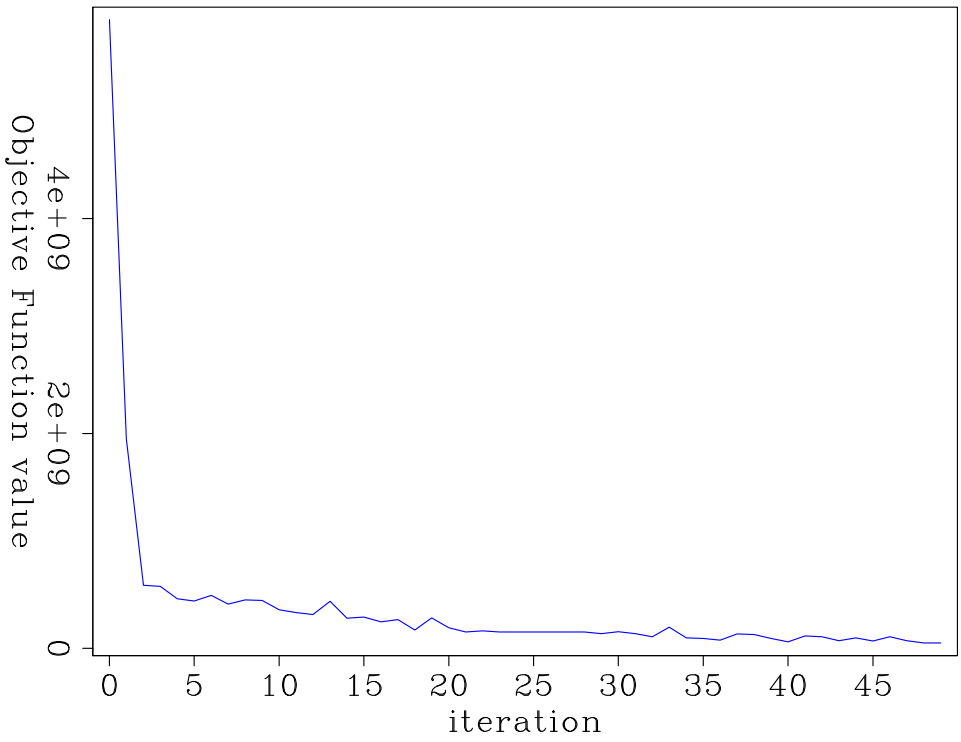

Figure 5: Objective function for evolution shown in Fig. 4 [**CR**]

Scaling parameter optimization

As shown previously, the salt boundary gradient is based on the adjoint of the linearized-Born operator, which is the tomographic update gradient. Since the gradient for both a tomographic and boundary update are calculated in each step regardless, we attempt to take advantage of this by finding scaling parameters to apply to these gradient updates such that we minimize the residual space objective function:

$$\left\| G_{tomo}^T \alpha + G_\phi^T \beta - (d_{calc} - d_{obs}) \right\|. \tag{29}$$

where G_{tomo}^T and G_ϕ^T are the update gradients for the background velocity and implicit surface ϕ respectively.

Minimizing this objective function gives us parameters that are scaled to the residual space, not the gradient space where they are actually applied. Since the adjoint operator that we use creates a scaling difference between the residual and gradient (data and model) spaces, we must rescale α and β once they are found so that they can be effectively applied to the gradients.

Two approaches can be used for this. The first is to fix the β parameter such that it is set to the maximum step size allowed by the stability conditions, and then rescale α while maintaining the ratio between them. This can be effective, but the

(general) decrease in the objective function is not very smooth, since the assumption of the maximum stable step size as being the most optimal choice for β is not always true. The second approach is to rescale α and β according to a γ parameter which is found using a non-linear line search. This technique is much cheaper than performing a full non-linear search for α and β, but allows for β to be variable (unlike the first approach).

One thing to note when observing the trend in Fig. 5 is that the objective function does not decrease monotonically. This may be due to the applied gradient not being positive-definite, which is possible considering the summation of the boundary gradient with a regularization term before application. We expect the objective function to not monotonically decrease as long as a tomographic update and a boundary update are applied together. This is because both gradients intrinsically contain unseparated tomographic and boundary information. When they are applied in the non-linear scheme that we use, this unseparated information can effectively be applied redundantly, causing an update that is not guaranteed to have a lower residual. For example, the tomographic gradient is applied incompletely, since any inter-salt velocity update is ignored (because we assume a constant salt velocity in this work flow). In this case, the tomographic gradient update and the actual final update may conflict in areas within the salt boundary. For this reason, we are not surprised by the non-monotonic decrease that we observe.

Tomographic gradient masking

In this work we assume a constant velocity throughout the salt bodies we model. Because of this, there is no application of the tomographic gradient update in the regions where salt exists. If we calculate G_{tomo}^T without first masking out G_{tomo} in areas overlapped by salt regions, then we introduce bias into the objective function (equation 29), since it will optimize for an update that will not be entirely applied.

We further assume that the salt boundary change will not undergo significant shifts. With this in mind, we apply the masking based on the salt body delineation that was created from the last iteration. Another approach would be to dynamically update the salt boundary based on the scaling parameter β, as β is being solved for. While theoretically producing a more accurate update, this method is also far more expensive, since numerous applications of the forward linearized-Born operator are necessary. For this reason we make the approximation of masking based on the previous iteration of the salt boundary.

Tomographic gradient smoothing

When the masking is performed and the salt boundary shrinks, then an area of the background velocity is exposed which contains a sharp boundary between the newly "exposed" region and the region that was previously exposed and updated. This can

create false (albeit weak) reflectors around the edge of the salt, causing errors as the evolution of the salt boundary continues. For this reason, immediately after masking is performed on the tomographic gradient, it passes through a smoothing operator which removes these sharp discontinuities in the velocity update.

FUTURE WORK

One aspect that is currently in development is a demonstration of the tomographic update being applied to a case where there is a velocity discrepancy between the initial guess and the true background model. In this work, we successfully demonstrate the tomographic update on the null-case (where the true model and initial background velocity models match), but this example is not realistic. Additionally, there are a few areas where the method described can be improved including:

- Extending the algorithm to perform evolutions on 4D surfaces in order to achieve segmentation on 3D isocontours
- Integrating stopping / freezing criteria into the algorithm structure such that convergence can occur
- Integrating expert knowledge into the initialization and evolution of the implicit surface.

3D model extension

The shape optimization algorithm that I use is based on level set theory that is easily extended to application on 3D models. The theoretical difference is simply an evolution of a 4D implicit isosurface, with the zero-level isocontour being the (3D surface) boundary of interest. This task requires a reorganization of the code base more than a rethinking of the theory that the algorithm is based on.

Convergence criteria

As it stands, no implementation of a stopping criteria is used in the algorithm described. In theory, when the residual is equal to zero, there should be no update expressed in the gradients that are calculated. On this basis, the algorithm should eventually reach convergence, assuming that the objective function decreases monotonically. In reality, the objective function is very non-linear, and the linear approximations used to evolve the model play a role in causing the objective function to vacillate. A decrease in the value of the objective function may indicate an update to a more accurate model overall, but is not useful for determining local accuracy; some regions may locally converge, but will continue to be updated since the gradient is

applied in equal weight (spatially) across the model space. A method to address this issue is based on freezing the model locally when it has met a convergence criteria. One approach is a "cooling" method, where the spatial weighting of points on the implicit surface is reduced as points remain defined as inside / outside the salt body for an increasing number of iterations. Eventually, all the weights are equal to zero, halting the updates. Another similar approach is based on measuring the length of the zero-contour boundary and stopping the iterations when its variability decreases below a threshold (Chaudhury and Ramakrishnan (2007)). Incorporating a method like one of these into the algorithm may result in more robust convergence.

Expert knowledge integration

While the semi-automated aspects of shape optimization can be very useful in generating models that fit our recorded data, there can be significant advantages to incorporating expert knowledge into the work flow such that the space of solutions is further constrained, allowing convergence to be reached more quickly. The shape optimization work flow described takes advantage of this at the initialization stage, where a salt boundary must be chosen. One possible way to extend the inclusion of expert input into the work flow is by allowing the user to not only set an initial guess, but also supply a confidence map of the boundary that is chosen. The sensitivity of the boundary to modification by the evolution update can be theoretically set by correlating the initial gradient of the implicit surface to the confidence of the initial boundary that was supplied. In areas where the gradient of the implicit surface is steep, the zero-level set crossing is less sensitive to updates as evolution progresses. One difficulty towards implementing this will be maintaining a regular implicit surface (such that numerical stability can be achieved) while still allowing for the gradient to vary based on confidence of the boundary.

CONCLUSION

In this work we described the derivation of the level set method as applied to the minimization of the FWI objective function. We demonstrated the application of this evolution algorithm and its incorporation with a background velocity tomographic update on a simple model. We consider the limitations of this approach in regards to numerical stability, as well as the assumptions of linearity that we use to find our scaling parameters. Last, we consider the possible pathways of future development of this approach, including the extension to three dimensions, convergence criteria, and especially the inclusion of expert input into the work flow.

ACKNOWLEDGEMENTS

I would like to thank Adam Halpert, Ali Almomin, Musa Maharramov, Sjoerd de Ridder, and Stew Levin for their insight into this work. Further, I thank the SEP sponsors for their generous financial support of this research work.

REFERENCES

Burger, M., 2003, A framework for the construction of level set methods for shape optimization and reconstruction: Interfaces and Free boundaries, **5**, 301–330.

Chaudhury, K. N. and K. R. Ramakrishnan, 2007, Stability and convergence of the level set method in computer vision: Pattern Recogn. Lett., **28**, 884–893.

Guo, Z. and M. de Hoop, 2013, Shape optimization and level set method in full waveform inversion with 3d body reconstruction: SEG Technical Program Expanded Abstracts, 1079–1083.

Lewis, W., B. Starr, D. Vigh, et al., 2012, A level set approach to salt geometry inversion in full-waveform inversion: Presented at the 2012 SEG Annual Meeting.

Li, C., C. Xu, C. Gui, and M. Fox, 2010, Distance regularized level set evolution and its application to image segmentation: Image Processing, IEEE Transactions on, **19**, 3243–3254.

Osher, S. and J. A. Sethian, 1988, Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations: Journal of computational physics, **79**, 12–49.

Santosa, F., 1996, A level-set approach for inverse problems involving obstacles: ESAIM Controle Optim. Calc. Var, **1**, 17–33.