Second-order updating in shape optimization for salt segmentation

Taylor Dahlke, Biondo Biondi, and Robert Clapp

ABSTRACT

Interpretation of sharp salt boundaries can be achieved by using level sets to define the boundary as an isocontour of a higher dimensional implicit surface. Using shape optimization, we can evolve this surface and the boundary it represents. We derive an update for the implicit surface that uses second-order information in the Hessian of the FWI objective function, taking into account the effects of the acquisition, as well as scattering and transmission energy. This approach helps us avoid local minima and more effectively converges on the true model, both in terms of the data and model residual norms. We demonstrate this idea using a Gauss-Newton approximation of the Hessian on synthetic examples.

INTRODUCTION

In hydrocarbon producing regions like the Gulf of Mexico and offshore Brazil, there is interest in exploring for resources that exist close to and below salt bodies. Seismic imaging is the primary method of identifying these resources, but unfortunately this technology runs into difficulties imaging around salt bodies (Leveille et al., 2011). Because salt has a higher velocity compared to the surrounding sediments, it is quite reflective. Salt often has complex shapes and high reflectivity, so the energy that strikes it is scattered before it reaches the targets of interest nearby (Etgen et al., 2009). Additionally, this energy may not be properly captured by the acquisition geometry, even with long offsets. If the boundaries of the salt are placed improperly, it becomes especially hard to identify targets along the flanks and base of salt. Furthermore, if a well is inadvertently placed through salt, this could complicate drilling, or the well could fail. For these reasons, it is important to correctly identify the boundaries of salt bodies in our earth models.

Tomographic approaches for 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 are common approaches for 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. Models are usually refined iteratively, which means manual adjustment of the salt bodies must be continuously revisited, causing a bottleneck in the overall work flow. A robust method for further automating the salt interpretation procedure during inversion would prove to be very useful in practice.

Some previous approaches to segmenting salt bodies use a shape optimization approach for evolving the boundaries (Guo and de Hoop, 2013; Lewis et al., 2012). These boundaries can be represented as the zero-isocontour of a higher dimensional surface (for example, a 2D boundary as a contour of a 3D surface). An updating step can be derived to evolve this shape / isosurface according to the Full Waveform Inversion (FWI) objective function. Unlike the smooth boundaries produced by tomographic approaches, the isocontour resulting from shape optimization provides a sharp boundary, which is a more appropriate way to classify many 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. Their approach creates and applies a steepest-descent update, which can create problems updating the base-of-salt (BOS) once the top-of-salt (TOS) has gotten close to convergence. This issue has been observed in recent work (Guo and de Hoop, 2013), and is an inherent problem with the steepest-descent update approach. In (Dahlke, 2015), we decomposed the domain of the model so that a line search was performed for both TOS and BOS gradient, allowing the BOS to continue updating after the TOS converged. However, splitting the domain into more regions in order to get better resolution of updating conflicts with the added cost from the line search that each new domain requires. Further, this approach can never take into account the relationships between model points; its application is ultimately a diagonal matrix constrained to n unique values (in the case of n domain regions).

To address the problem with steepest-descent updating, we utilize the second-order information provided by the Hessian of the objective function, in order to choose better search directions in our inversion and avoid local minima. We use a Gauss-Newton approximation to the Hessian in our inversion. Further, we suggest the potential of using the scattering and transmission components of the Hessian in order to further improve our updating. We also take advantage of the sparsity of our final update to reduce the I/O and computation necessary for performing the Born and Gauss-Newton Hessian operators.

In this paper we will begin by discussing the fundamentals of the level set method, followed by the derivation of the second-order boundary update. Next we will describe and demonstrate the algorithm used, and discuss the assumptions and fundamental limitations of this approach. Following, we will compare the first order updating provided by a steepest-descent approach to a second order updating, and demonstrate the improved results that this approach offers. Last we will discuss the compression applied to the operators used and the future opportunities related to that.

THEORY

While it may seem counter-intuitive to add an extra dimension to our problem, by doing so, we gain the advantage of easily merging/separating bodies as the evolution proceeds, as well as the ability to handle sharp corners and cusps in the lower-dimensional (2D) plane on which the boundary exists.

Osher and Sethian (1988) and Burger (2003) describe the level set of ϕ that represents the salt body boundary as $\phi(x_\Gamma, \tau) = 0$, where x_Γ is the set of points along a boundary Γ, and τ is the iteration count. By taking the derivative of this equation with respect to τ (to find the $\delta\phi$ between iterations), applying the chain rule, and re-arranging terms we can get:

$$\frac{\partial \phi}{\partial \tau} = -V(x_\Gamma, \tau)\left|\nabla \phi\right|. \tag{1}$$

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. We derive this normal velocity such that the FWI objective function is minimized

$$\psi = \min \left\| F(m) - d \right\|_2^2, \tag{2}$$

where $F(\cdot)$ is the forward wavefield modeling operator, m is the velocity model, and d is the observed data.

Calculus of variations

The shape derivative we use is based on a formal calculus of variations outlined in Santosa (1996). The objective is to define the variation of the model m with respect to the boundary variation (represented implicitly by the surface, ϕ).

Figure 1 illustrates that $\delta m(x_\tau)$ will be $\pm(m_{\text{int}} - m_{\text{ext}})$, depending on the relative values of m_{int} and m_{ext} or the sign (direction) of the normal vector $\vec{n}$. We only care about the component of δx_Γ that occurs in the normal direction, because a tangential variation of x_Γ does not affect m or ϕ. Because of this, we can express $\delta m(x)$ as

$$\delta m = (m_{\text{int}} - m_{\text{ext}})\delta \vec{x_\Gamma} \cdot \vec{n} \mid_{x \in \partial\Omega}. \tag{3}$$

which can be considered a measure over $\partial\Omega$.

We consider an inner product of velocity model perturbation δm with a test function $f(x)$. Formally, this can be written as,

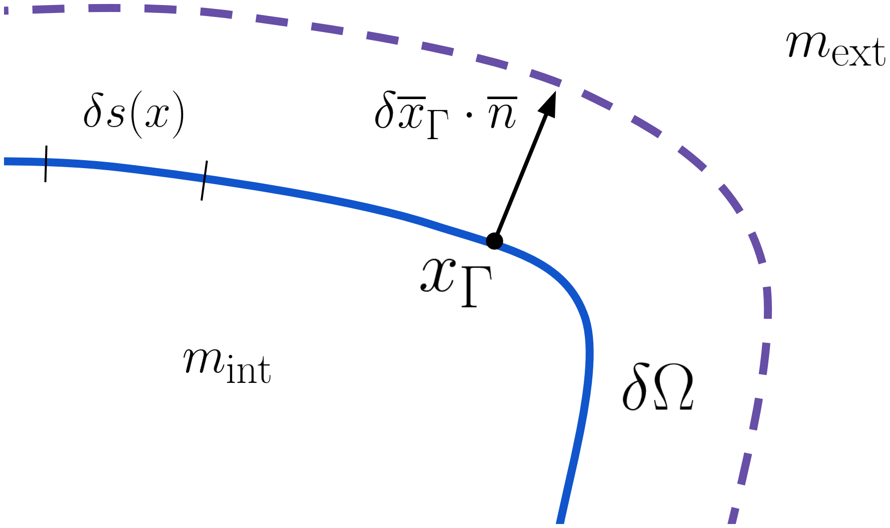


Figure 1: The geometry of the curve $\{x_\Gamma : \phi = 0\}$ for a variation $\delta\phi(x)$ for an evolution step τ. $\delta\Omega$ is the perturbation of the salt body Ω. [**NR**]

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

Because the $\delta m(x)$ term equals zero in $\mathbb{R}^2 \setminus \partial\Omega$, it does not contribute to the overall inner product when integrating over that domain; therefore, we only integrate over $\partial\Omega$ where $\delta m(x)$ is non-zero.

We want to decrease our objective function (2), so we choose a second-order Newton step such that $\triangle m = -H^{-1} g$. In the case of the FWI objective function, we can use the Gauss-Newton approximation of the Hessian such that

$$\triangle m = -[B^T(m_0)B(m_0)]^{-1} g, \tag{5}$$

where $B(m_0)$ is the linearized Born operator at $m = m_0$, and $g = B^T(m_0)r$ is the adjoint Born operator applied to the data space residuals, as described in Plessix (2006). Since this is the best search direction to decrease our objective function (2) in the quadratic sense, we substitute it into $f(x)$ from (4) to get

$$\langle \delta m, f(x) \rangle = \int_{\partial\Omega} \delta m \left(\left[B^T(m_0) B(m_0) \right]^{-1} B(m_0) r \right) dx. \tag{6}$$

Because we are interested in the projection of this search direction on the constraining equation for δm that we outline in equation (3), we make our substitution for δm yielding to:

$$\langle \delta m, f(x)\rangle = \int_{\partial\Omega} (m_{\text{int}} - m_{\text{ext}}) \delta\vec{x_\Gamma} \cdot \vec{n} \left(\left[B^T(m_0) B(m_0) \right]^{-1} B(m_0) r \right) ds(x). \tag{7}$$

Because δx_Γ is infinitesimal, we replace dx with $\delta\vec{x_\Gamma} \cdot \vec{n}$ when we substitute into (7). We call $ds(x)$ 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 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:

$$\delta\vec{x_\Gamma} \cdot \vec{n} = V(x_\Gamma, \tau). \tag{8}$$

In order for the inner product that we have defined in (7) to represent a decrease in the objective function (2), we need to choose a $V(x_\Gamma, \tau)$ such that $\langle \delta m, f(x)\rangle < 0$. The choice of $V(x_\Gamma, \tau)$ that give us the most negative value is opposite to the other terms, i.e;

$$V(x_\Gamma, \tau) = -\int_{\partial\Omega} (m_{\text{int}} - m_{\text{ext}}) \left(\left[B^T(m_0) B(m_0) \right]^{-1} B(m_0) r \right) ds(x) \tag{9}$$

When we substitute (9) into (1) we get our final update equation for our implicit surface

$$\frac{\partial \phi}{\partial \tau} = (m_{\text{int}} - m_{\text{ext}}) \left(\left[B^T(m_0) B(m_0) \right]^{-1} B(m_0) r \right) \left| \vec{\nabla} \phi \right|. \tag{10}$$

General evolution algorithm

Our algorithm takes the following form:

```
Initialize $\phi$, $m_{ext}$
$r \leftarrow$ calculate residual
$\psi_{new} \leftarrow \|r^2\|$
**for** $it = 1, \text{niter}$ **do**
    **if** $\psi_{new} < \psi_{old}$ **then**
        $g_{vel} \leftarrow B^T r$
        $g_\phi \leftarrow K(g_{vel})$
        $\beta_{max} \leftarrow$ find max step size
        $g_{DRLSE} \leftarrow$ calculate DRLSE
        $\phi_{it+1} = \phi_{it} + \beta_{max} \cdot g_\phi + g_{DRLSE}$
        $r \leftarrow$ calculate residual
    else
        $\beta \leftarrow$ line search
        $\phi_{it+1} = \phi_{it} + \beta \cdot g_{\phi} + g_{DRLSE}$
        $r \leftarrow$ calculate residual
    end if
end for
```

First we initialize our implicit surface, background velocity, and subsequently our full velocity model. Next, we compute synthetic data based on that full velocity model. After this operation we find our data residual and calculate our objective function value. We check to make sure the objective function value is decreasing for following iterations, but not the first. Then we compute a gradient using the adjoint Born operator. We use the conjugate gradient method to compute the application of the inverse of the Gauss-Newton Hessian on the FWI gradient calculated previously. Next we compute the search direction of the implicit surface (the K operator described in the algorithm above), as well as the maximum step size for our line search in a manner that satisfies the Courant-Friedrichs-Lewy (CFL) condition. By default, we take a step using the maximum β step size. But if that step size lowers our objective function value, we proceed to calculate a new gradient. If it fails, then we undo our update such and perform a line search to find the optimal β value. We then use that value to scale the implicit surface search direction. We also add the DRLSE (Distance Regularized Level Set Evolution) term in order to stabilize the evolution of the implicit surface (Li et al., 2010). The β value is already calculated in a way that accounts for this DLRSE term. Last, the update is applied to the model, new synthetic data and residuals are made, and a new FWI gradient is calculated for a new iteration.

Compressed imaging

One aspect of the problem that lends itself well to improvements in efficiency is the relatively narrow region set of implicit surface model points that need to be updated in order to impact the velocity model itself. Because of the nature of level set theory, typically only points that are close to $\phi = 0$ will actually change the boundary of the salt. It is possible for model points that are far from the current boundary to create a new body (or inclusion), but this behavior might be assumed to happen less frequently. If we make that assumption, then we are really only interested in doing update calculations for the narrow region around the existing boundary. We can take advantage of this to reduce the number of computations we perform in our algorithm.

One particular way that this idea is incorporated is by saving wavefields. For the Born and Gauss-Newton computations we require the source wavefield for the imaging steps of these operators. When we calculate the synthetic data initially, we can save the wavefield for later use, and avoid recomputing it. Depending on the model size though, reading this saved wavefield from disk can be equally or more expensive than recomputing it again. However, we really only need this wavefield for the imaging condition, and we only need to perform the imaging condition on the narrow region set. For this reason, we can compress the source wavefield to only the narrow region set of points, making it much faster to later read it from disk. In the case of the model shown in 2(b), we are able to compress the wavefield to about 10% of its original size, which can be read from disk much more quickly than it can be recomputed. Furthermore, we gain a smaller advantage from the imaging condition having to only be performed on 10% of the model points.

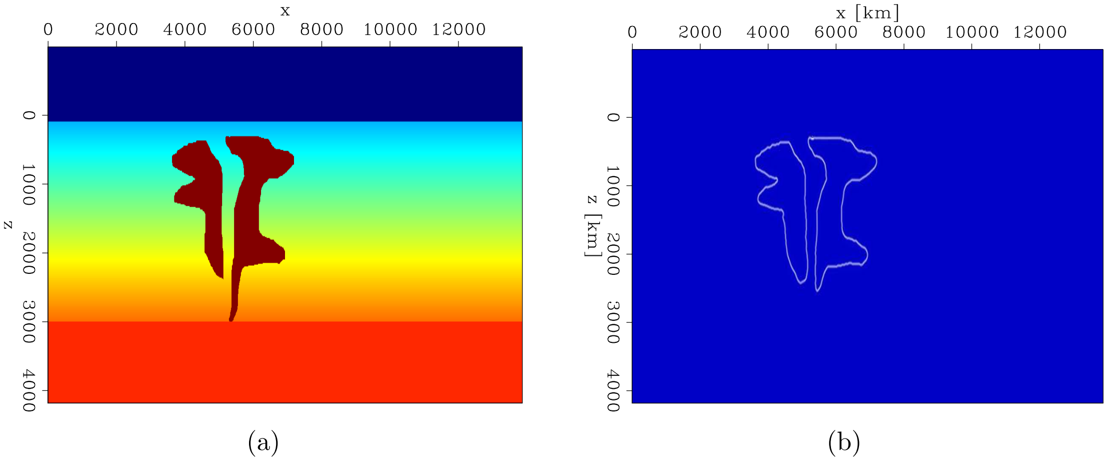


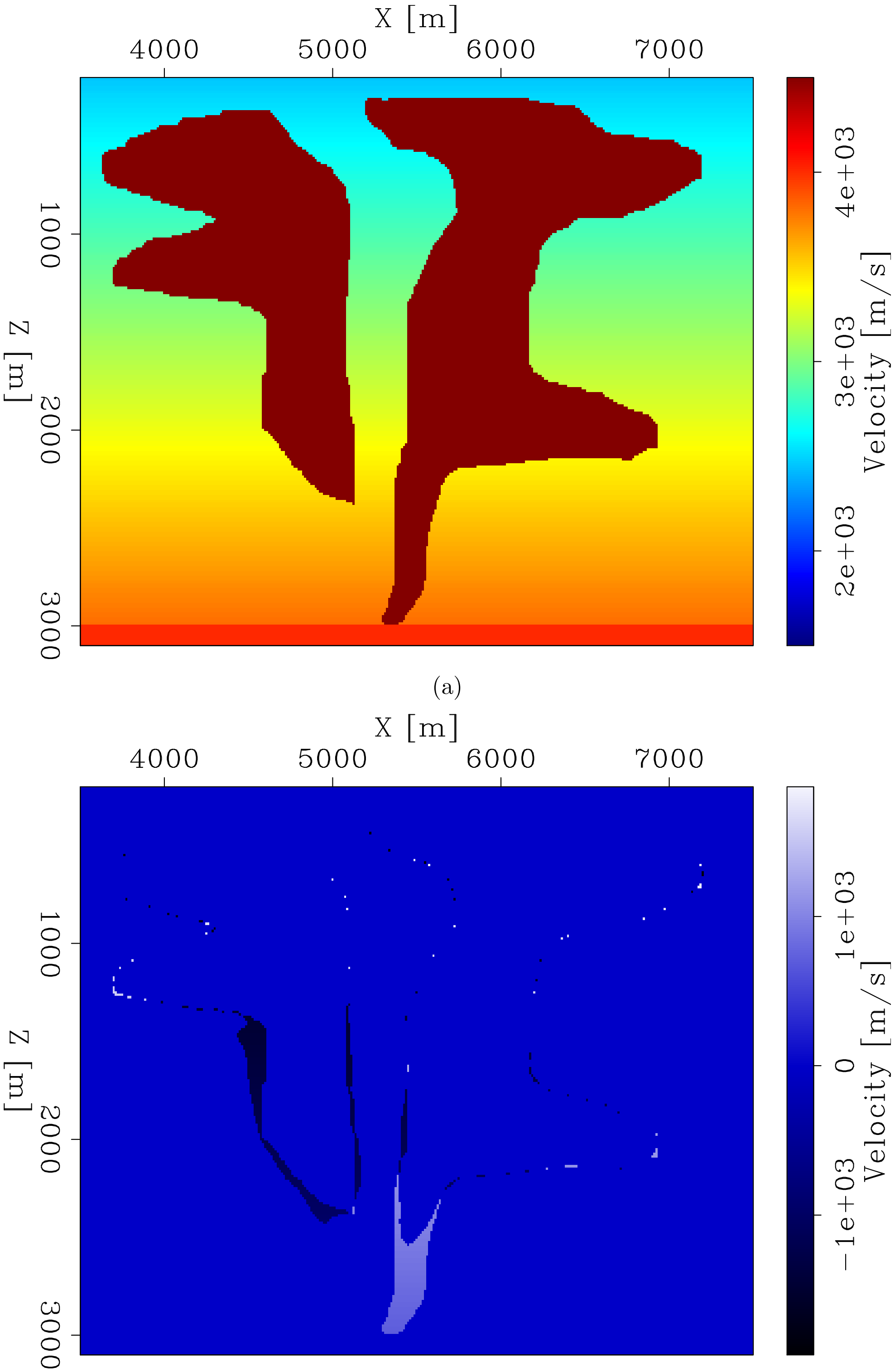
Figure 2: (a) True model (full extent); (b) Boundary zone that was captured in compression step. [**ER**]

APPLICATION

We begin by using a perturbed starting model, and then using a fixed step-size steepest descent approach to get a reasonably close convergence to the true model. It is the result of this step that is the starting model used for the following examples. This initial attempt at convergence gives us a model where the more difficult features (like overhangs, or steep salt flanks) are still in need of correction (see Figure 3), and where the top of salt regions are established, as is often the case an exploration imaging project.

We use a trailing line acquisition with 59 shots, each with 240 receivers. The receiver spacing is 25 [m], and the shot spacing is 100 [m], giving us 6100 [m] of offset. We use a shallow water bottom at 100 [m], making our model similar to a North Sea marine environment.

We can see the specific areas where the two methods differ by looking at the differences between the updates that have been made to the model using either method. This is shown in 4. These changes are actually quite small, but show that the Hessian updating method does a better job at model convergence on some of the areas that



(a)

(b)

Figure 3: (a) True model (zoomed in); (b) Difference between full true model and initial guess. [**ER**]

are otherwise poorly illuminated, particularly at the base of salt body on the left side, as well as under the overhang.

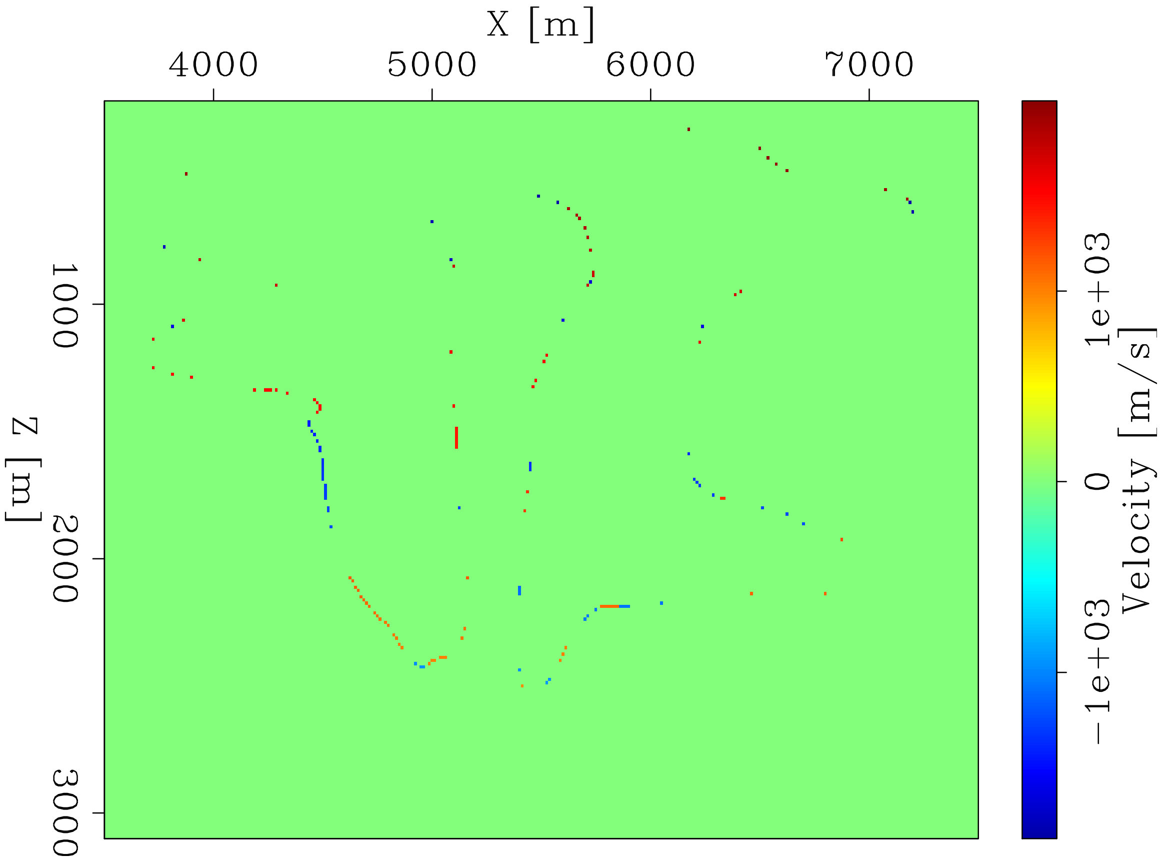


Figure 4: The difference between the absolute value of the changes made by each method, i.e: (ABS(Hessian_result - true_model) - ABS(Steepest_descent_result - true_model)). This plot can be interpreted as red areas being regions where the Gauss-Newton method does a better job, while blue areas are regions where the Steepest Descent method does a better job. [**CR**]

When we look at the decrease in the objective function with iteration, we can see that the Gauss-Newton method correctly decreases faster than our steepest descent approach, for both the model residual norm (5) and for the data residual norm (6). We used the same line search algorithm (quadratic interpolation) for both cases. However, in this comparison, the steepest descent norm curves make a notable jump at iteration 13. This rise occurs for a case where the alogithm reaches a local minima such that the line search chooses a step size of zero. Since making this step size is pointless, we choose a small step size that will attempt to get us out of the local minima and continue descent. We find that in this case the approach is successful, and we continue to decrease both norms accordingly. We choose to use the largest stable step size for each iteration, and then check to see if a reduction in the objective function is achieved. When it is not, then we discard that update and redo it using a line search for the best step size parameter. Portions of the residual norm figures where the curve is flat (no change) are iterations where the maximum step size is not sufficient, and where the line search is performed in its place. This is done for efficiency, as a line search for many of the iterations would otherwise choose the maximum step size anyway.

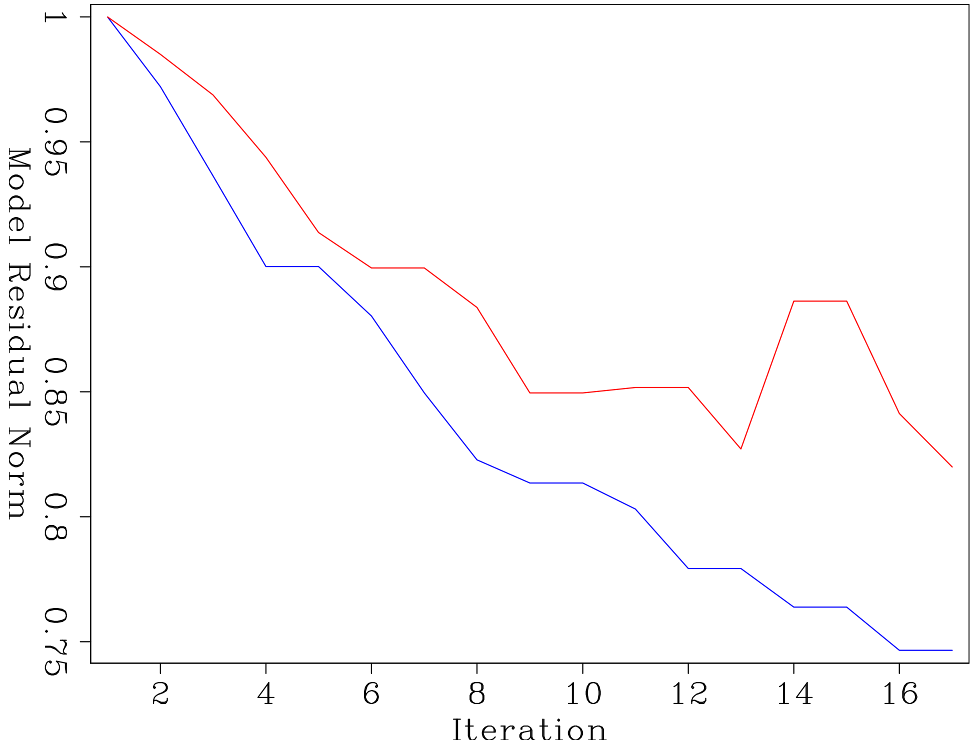


Figure 5: Comparison between the norm of the model residual for each method. Red is steepest descent method, while blue is the Gauss-Newton Hessian method. [**CR**]

CONCLUSIONS

In conclusion, we find that by using second order information in our updating of the implicit surface, we gain improved convergence of our model, both in terms of the model residual norm and the data residual norm. However, one aspect that we must contend with is the increased cost of inverting the Hessian, and whether that cost is worth the improvement that we see. Using compression methods on our model domain for the boundary that we are interested in may be useful for making this Hessian inversion cheaper, and is an area to be actively explored moving forward.

ACKNOWLEDGMENTS

Taylor would like to acknowledge the constructive conversations that I've had with my colleagues in SEP, especially Guillaume Barnier, Ettore Biondi, and Musa Ma-

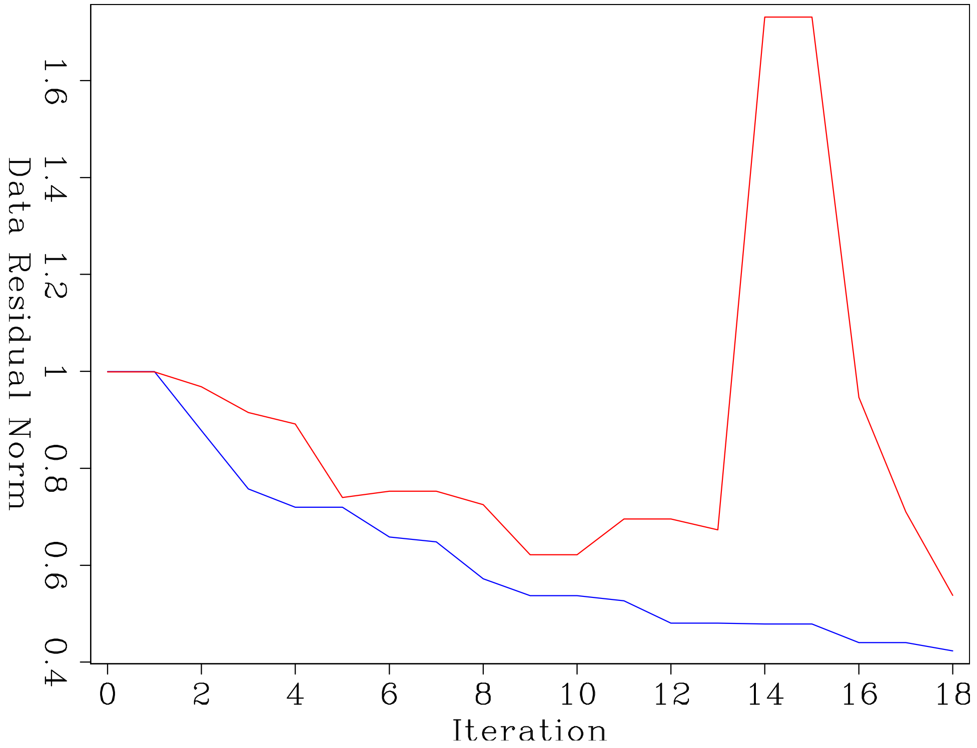


Figure 6: Comparison between the norm of the data residual for each method. Red is steepest descent method, while blue is the Gauss-Newton Hessian method. [**CR**]

harromov. I am very grateful to the sponsors of the Stanford Exploration Project for both their technical advice and financial support.

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.

Dahlke, T., 2015, Domain decomposition in shape optimization for segmenting salt bodies: SEP-158, 51–66.

Etgen, J., S. H. Gray, and Y. Zhang, 2009, An overview of depth imaging in exploration geophysics: Geophysics, **74**, WCA5–WCA17.

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.

Leveille, J. P., I. F. Jones, Z.-Z. Zhou, B. Wang, and F. Liu, 2011, Subsalt imaging for exploration, production, and development: A review: Geophysics, **76**, WB3.

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.

Plessix, R.-E., 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503.

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