Application of full Hessian on level set formulation and parameter inversion

Taylor Dahlke, Biondo Biondi, and Robert Clapp

ABSTRACT

Level sets are subsets of a domain that have the same value for a certain function. We can use them as a tool to update discrete boundaries of homogeneous bodies, which makes them particularly useful for updating salt models. Often, salt takes complicated geometries which causes a lack of direct illumination, as well as interactions between boundaries. Deriving a formulation of the Hessian which takes into account the level set parametrization should allow for better search directions than simpler methods. We find that by linearizing the velocity model perturbation with respect to the underlying background and level set parameters, we can derive a Hessian application operator suitable for a linear inversion scheme to get an improved search direction for updating the salt boundaries.

INTRODUCTION

Full Waveform Inversion (FWI) is generally used to update a continuous earth parameter like density or acoustic velocity. However, we often do not have enough frequency information to resolve features that have sharp boundaries, which is generally true for salt bodies in the Eastern Gulf of Mexico and other areas. It has been shown in previous work (see Santosa (1996), Lewis et al. (2012) and Burger (2003)) that level sets can provide a way to update these types of sharp boundaries.

However, this tool is meant to assist in our ability to invert for the entire earth model, which includes regions best approximated by continuous rather than discrete updates. This presents a problem of making updates to two domains; both an implicit surface that represents the sharp boundary, as well as the continuous background velocity field (as we do in traditional FWI). Previous work has demonstrated inversion schemes that update these domains sequentially Guo and de Hoop (2013), and also concurrently Dahlke et al. (2015) by finding global scaling variables for each gradient using line search methods. Both of these methods, while relatively inexpensive, are susceptible to local minima, and are ultimately first order updates that do not account for the interaction between model points, or the effects of acquisition.

By deriving a Hessian on a new model space that contains both level set (salt) and background (non-salt) parameters, we can better address the interaction between the salt and non-salt components of the model, and ultimately find a better update to these parameters.

We address this by first outlining the theory behind a model parametrization that contains both domains. Next we derive an application of the Hessian to this model. Afterwards, we describe the numerical application of this operator in an inversion scheme. Last we show numerical results of using this Hessian operator in our inversion scheme, and compare against results from more simplistic approaches.

DERIVATION

The first step of this derivation is to describe the model space that we are working with. We will call our velocity model m which we define as:

$$m(\phi, b) = H(\phi)(c_{\text{salt}} - b) + b \tag{1}$$

where $H(\circ)$ is the Heaviside function, ϕ is the implicit surface, b is the background velocity model, and c_{salt} is the acoustic velocity of the salt. The dimensions of m, ϕ, and b are all NXxNZ since they exist over the same spatial domain (2D in this case). This means that the implicit surface is greater than zero in the salt body region, and less than zero outside of it. We generalize these parameters for the entire spatial domain (ignoring i, j), and expand this definition with a Taylor series as:

$$m_1 = m_0 + \frac{\partial m}{\partial \phi}\Big|_{m_0} \triangle\phi + \frac{\partial m}{\partial b}\Big|_{m_0} \triangle b + ...$$

By truncating this series and ignoring higher order terms, we can create a linear approximation for the perturbation of the velocity model m with respect to ϕ and b:

$$m_1 - m_0 = \triangle m \approx \frac{\partial m(\phi_o, b_o)}{\partial \phi}\triangle\phi + \frac{\partial m(\phi_o, b_o)}{\partial b}\triangle b. \tag{2}$$

This can be written as a matrix operation:

$$\triangle m \approx \begin{bmatrix} \frac{\partial m(\phi_o, b_o)}{\partial \phi} & \frac{\partial m(\phi_o, b_o)}{\partial b} \end{bmatrix} \begin{bmatrix} \triangle\phi \\ \triangle b \end{bmatrix}.$$

When we define $\triangle p$ as

$$\triangle p = \begin{bmatrix} \triangle\phi \\ \triangle b \end{bmatrix},$$

this gives us:

$$\triangle m \approx \begin{bmatrix} \frac{\partial m(\phi_o, b_o)}{\partial \phi} & \frac{\partial m(\phi_o, b_o)}{\partial b} \end{bmatrix} \triangle p.$$

Where we define operator D as:

$$\begin{aligned} D &= \begin{bmatrix} \frac{\partial m(\phi_o, b_o)}{\partial \phi} & \frac{\partial m(\phi_o, b_o)}{\partial b} \end{bmatrix} \\ &= \begin{bmatrix} \delta(\phi)(c_s - b) & 1 - H(\phi). \end{bmatrix} \end{aligned} \tag{3}$$

This operator D ultimately scales and masks the parameter fields $\triangle\phi$ and $\triangle b$. With this new approximation of the perturbation in our velocity model, the application of our Born operator (B) to our new model parameter space $\triangle p = [\triangle\phi \quad \triangle b]^T$ is:

$$\begin{aligned} \triangle d &= B \triangle m \\ \triangle d &\approx BD \triangle p. \end{aligned}$$

Alternatively, we can find the gradient for our model parameters by applying the adjoint operations:

$$\triangle p \approx D^T B^T \triangle d.$$

Similarly we can find the application of the Hessian of the FWI objective function as:

$$D^T H D \triangle p \approx -D^T B^T \triangle d. \tag{4}$$

In equation 4, we can substitute H with either the full or Gauss Newton Hessian. Previous work by Fichtner (2010) shows that the full Hessian of the FWI objective function can be constructed by summing a WEMVA component with the Gauss-Newton component of the Hessian. It is this formulation of the full Hessian application that we use. The method I propose is to solve equation 4 for $\triangle p$ using a conjugate gradient inversion method.

Previous work (Dahlke et al. (2016)) showed a similar technique, with the transformation to $\triangle p$ being done after an inversion for the $\triangle m$ search direction is completed (without the "constraint" of projection operator D). This means instead of solving 4, we solve:

$$H \triangle m = -B^T \triangle d, \tag{5}$$

followed by:

$$\triangle p = D^T \triangle m. \tag{6}$$

The method using equations 5 and 6 will be referred to as unconstrained inversion, while the proposed method (the solution of equation 4) is constrained by D. We compare the two methods later in this paper.

NUMERICAL APPLICATION

Inside the D operator we have both a Heaviside function $H(\cdot)$ as well as a Dirac-delta function $\delta(\cdot)$. While computing the application of the Heaviside function on our implicit surface ϕ is relatively simple for a discrete case (masking with a threshold), the same cannot be said for the application of $\delta(\cdot)$ to our numerical problem. Most of our boundary is not explicitly represented as points where $\phi = 0$, but simply by the adjacency of positive and negative ϕ values. For this reason, we relax the definition of our operator to use an approximation of the Dirac-delta function. Previous derivations using calculus of variations (Santosa (1996), Dahlke et al. (2015)) have demonstrated that we can reduce our FWI objective function using an update for ϕ defined as:

$$\triangle\phi = |\nabla\phi|(c_{\text{salt}} - b)B^T\triangle d. \tag{7}$$

This is similar to the update we've derived using the linearization of the Taylor series expansion:

$$\triangle\phi = \delta(\phi)(c_{\text{salt}} - b)B^T\triangle d. \tag{8}$$

For this reason, we substitute $|\nabla\phi|$ for $\delta(\phi)$, as it suffices as an approximation of the Dirac-delta function when we keep the implicit surface slope regularized as we update it. In our case, this means using a double-well potential function fomulation of Distance Regularized Level Set Evolution (DRLSE) as described in Li et al. (2010). This step provides a forcing term to help keep the slopes of the implicit surface at either zero or one. When that is maintained, the only place where $|\nabla\phi| > 0$ is immediately surrounding the boundary. Using this substitution, the D operator we apply is:

$$D = \begin{bmatrix} |\nabla\phi|(c_{\text{salt}} - b) & .1 - H(\phi) \end{bmatrix} \tag{9}$$

RESULTS

To test the effectiveness of this linearization, I pose the following question: can I get a first-step update using this new method that is better than other methods? To answer this, I choose to find search directions using the proposed method as well as two other approaches, and then use a line search to find the step-length α for each search direction. These three search directions methods are steepest-descent, unconstrained inversion, and the proposed constrained inversion. I then do a model update and compare the model space residual to see which update performed better.

The first of the two comparison methods is simply steepest descent, where the search direction is my gradient: $\triangle p = D^T B^T \triangle d$. The second method is a little more subtle. With the second method, I use the application of the full Hessian as described in Fichtner (2010) and invert for $\triangle m$, followed by the application of D as described in equations 5 and 6.

To perform the inversion for the search direction in either case, we made use of the out-of-core solver developed by Biondi and Barnier (2017). We implemented a conjugate-gradient (CG) approach to take advantange of the symmetry of the Hessian operator (as opposed to conjugate gradient least-squares (CGLS), which doesn't require a symmetric operator). By using this method we were able to invert at the cost of one forward operation per iteration rather than both the forward and adjoint operations required by CGLS. One further requirement of using CG is that the operator must be positive definite, while for CGLS it need not be. For the Gauss-Newton Hessian we can use either safely, but with the full Hessian, we risk a failure in our inversion, depending on the velocity model that we are using. For the models we use in this paper, we found the CG inversion to be stable, so it was used instead of CGLS since it performs at half the cost. When CG is used, the objective function value will not necessarily converge to zero since it isn't a least squares formulation like CGLS (see Aster et al. (2011) for more). For all cases shown, we start at an objective function value of zero since we begin the inversion with a zero initial model.

We perform this comparison on two different velocity model examples. Both examples have the same true model, but have different initial model states (see Figure 1). In case one, the base of the circular salt is perturbed downwards. In case two, the top of the bottom reflector salt is perturbed upwards.

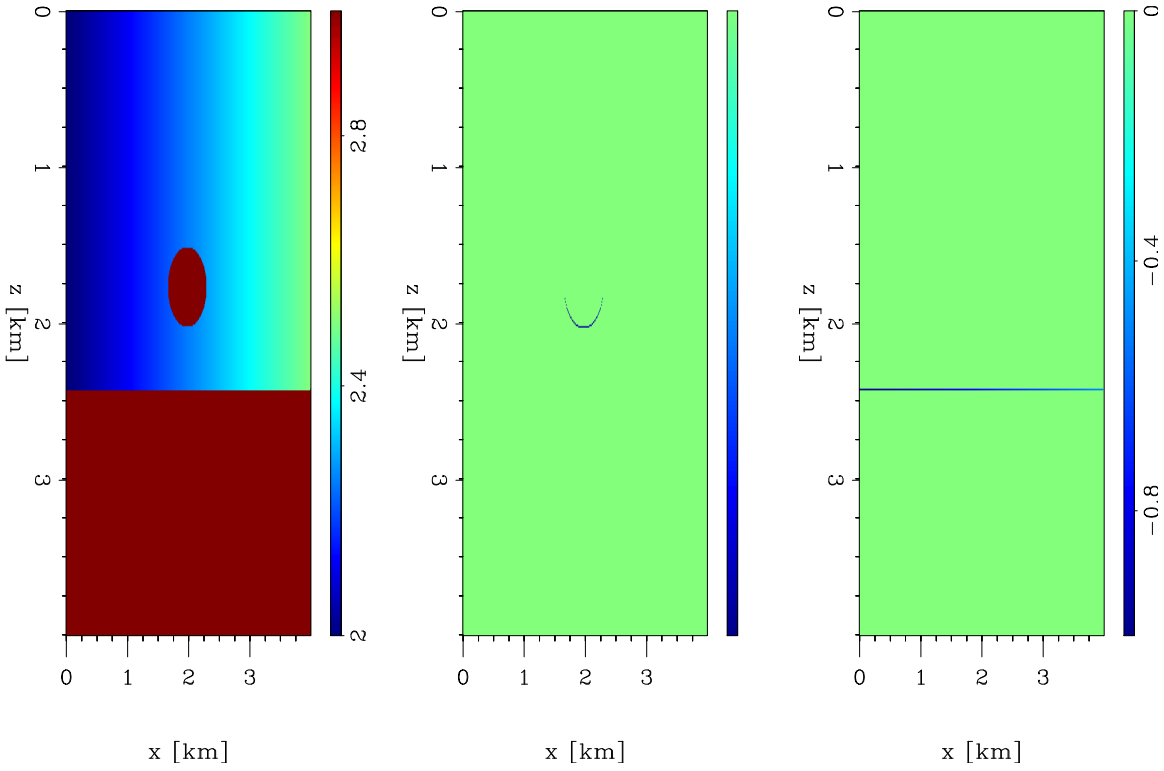

Figure 1: True model (left); the difference between the true and inital model for case one (middle); the difference between the true and inital model for case two (right). [**ER**]

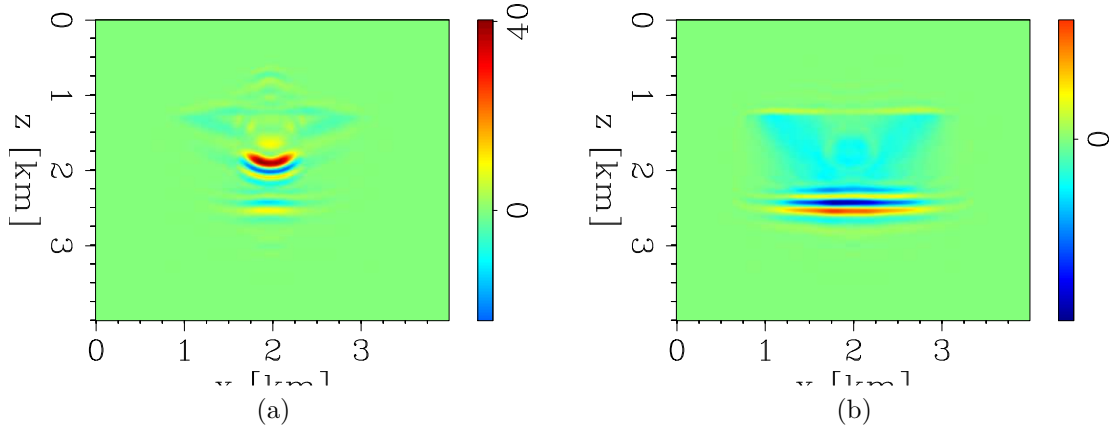

Figure 2: Adjoint Born images for (a) perturbed salt base case; (b) perturbed bottom reflector case; no clipping of amplitudes. [**CR**]

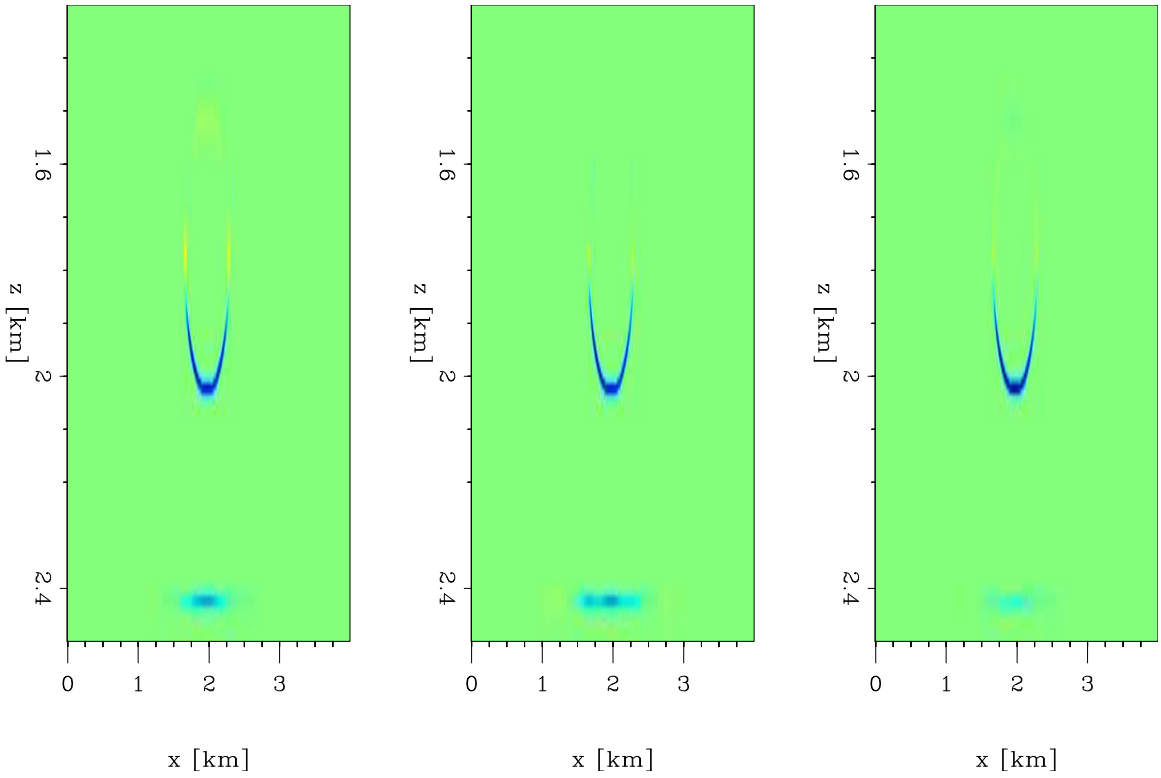

Figure 3: CASE 1: Search directions for the ϕ update resulting from Steepest descent approach (left), unconstrained Hessian inversion (middle), and proposed Hessian inversion scheme (right). [**CR**]

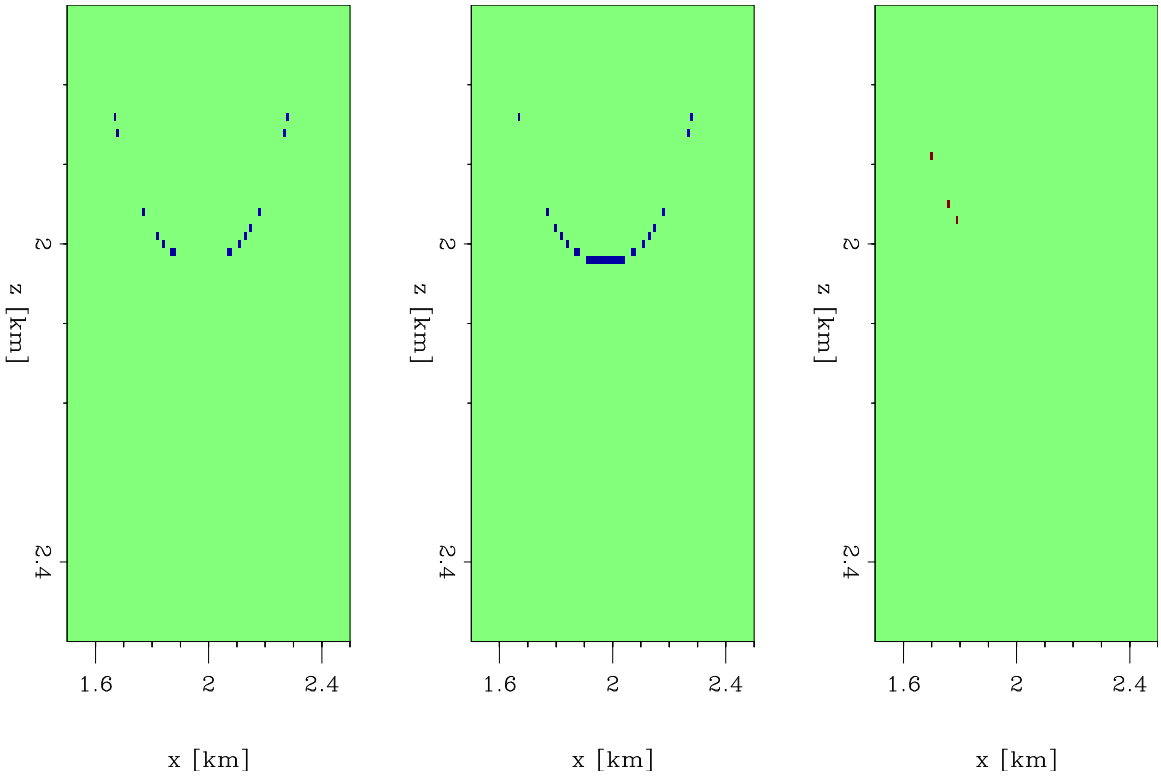

Figure 4: CASE 1: Difference between true model and velocity model applying first update to ϕ using a line search method to find step size α. (Left) based on steepest descent, (middle) based on unconstrained Hessian inversion, (right) based on proposed Hessian inversion incorporating projection operator D as a constraint. [**CR**]

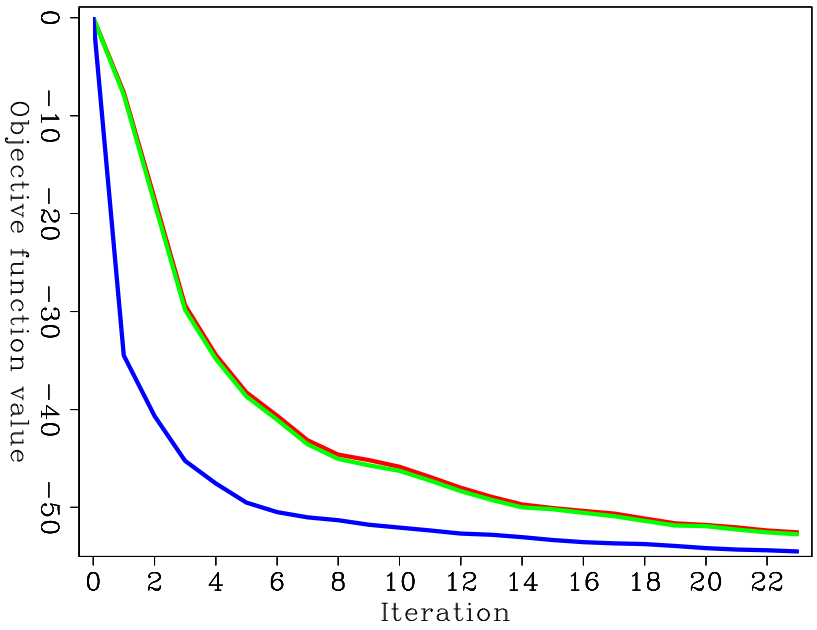

Figure 5: CASE 1: Objective function decrease for the three search direction inversions; unconstrained search direction inversion with no D operator (blue), Proposed method (red), and proposed method with scaling term equal to one (green). [**CR**]

Table 1: Case one: Perturbed base of salt

Update method	Model residual norm
Steepest descent	8.617
Hessian inversion (unconstrained)	16.658
Hessian inversion (proposed)	1.863

For the first case, we find a significant improvement by using the proposed method when compared to the other two methods. We can see from the search directions that the new Hessian inversion result (Figure 3, right) has a more correct focusing of energy on the base of salt rather than on the bottom reflector as with the other two methods (Figure 3 left, middle). This leads to a better update when we perform a line search on these search directions (Figure 4). When we difference the updated model with the true model and take the L2 norm of the residual, we see that the proposed method has an improved update when compared to either of the other approaches (Table 1).

One further question that we have is the effect of either the masking or scaling components inside the D operator. Which of these makes the most difference? In both cases, we have an unperturbed background velocity model where the scaling term inside D varies spatially (note the horizontal gradient in Figure 1, left), so the scaling term $(c_{\text{salt}} - b)$ inside D is non-trivial. To test this, we perform our proposed method as before, with the exception of the scaling term being equal to one across the whole spatial domain, eliminating its effect. We find that the search direction that results from the inversion is similar to the result from the originally proposed

definition of D (Figure 6) in terms of amplitude distribution. This similarity is also reflected in the objective function descent (Figure 5), where the two curves are nearly identical. We do find that the lack of the scaling term increases the magnitude of $\triangle\phi$ in the correct direction by about 16% in the same areas that the original approach tries to update. This improved search direction found by ignoring the scaling requires further investigation.

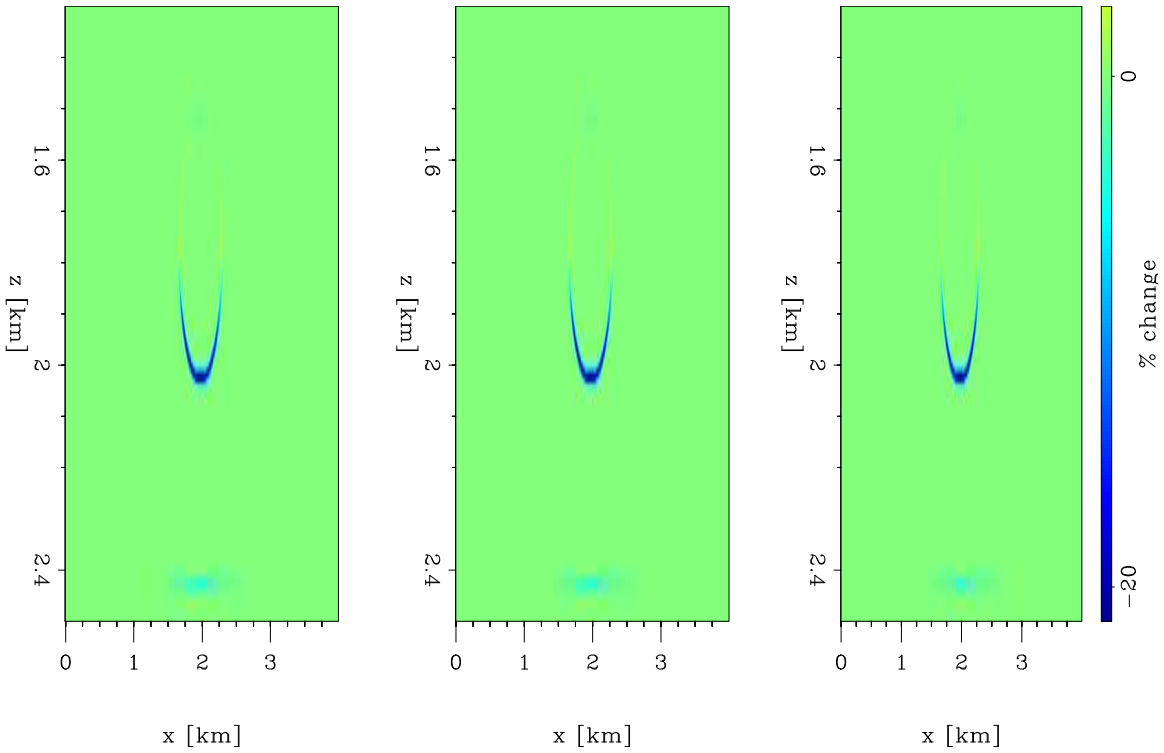

Figure 6: Search direction of proposed method (left), and proposed method without scaling component (middle). Percent difference between (left) and (middle) search directions (right). [**CR**]

For the second case with the perturbed bottom reflector, we find that we still have an improved result, but slightly less obvious. When we compare the objective function decrease for the two inversion approaches (Figure 9), we can see that these curves are far more similar than in the case one example (Figure 5). The steepest descent result (Figure 7, left) correctly has most of its energy on the bottom reflector rather than on the circular salt as the other two inversion methods do. However, the inverison methods both correctly update a larger portion of the perturbation, with the proposed method doing so with slightly less energy incorrectly applied to the circular salt above. This leads us to get an improved update after line search (Figure 8). Looking at the model residual norm again, we find that the proposed method again performs better, in this case with less of a margin than before (see Table 2).

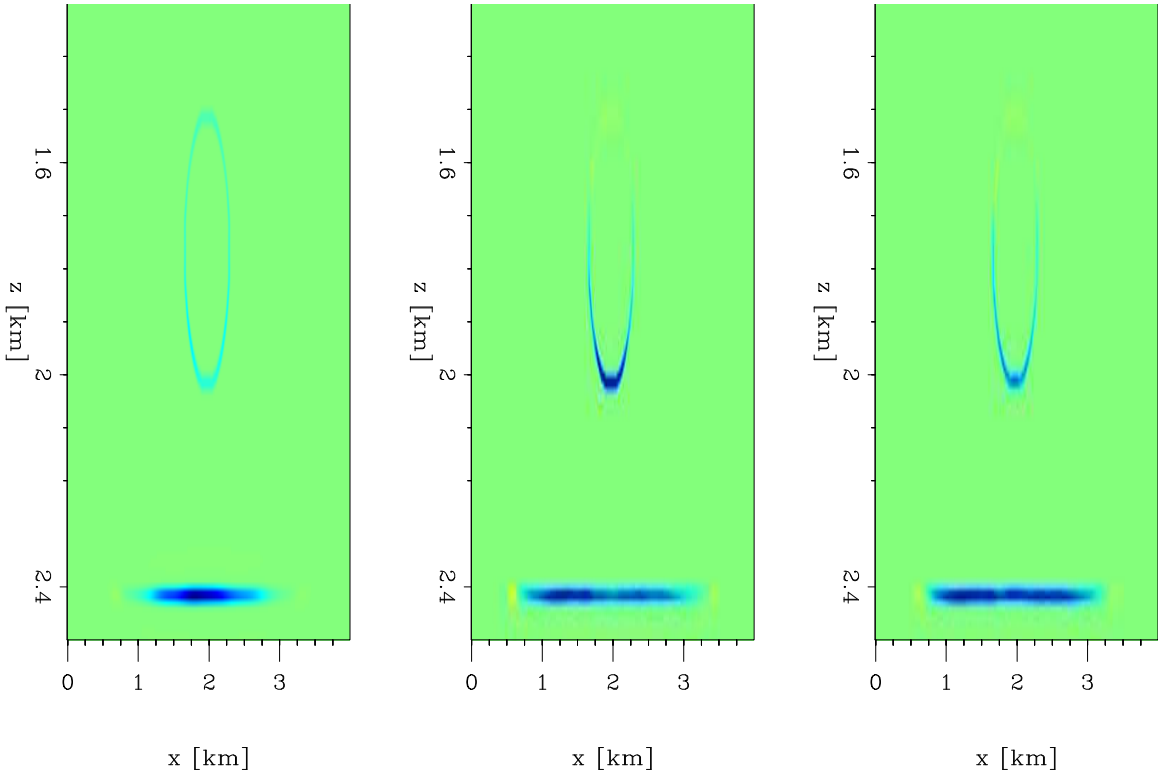

Figure 7: CASE 2: Search directions resulting from steepest descent approach based on gradient (left), from unconstrained Hessian inversion (middle) , and from proposed Hessian inversion scheme (right). [**CR**]

Table 2: Case two: Perturbed bottom reflector

Update method	Model residual norm
Steepest descent	172.683
Hessian inversion (unconstrained)	144.088
Hessian inversion (proposed)	135.161

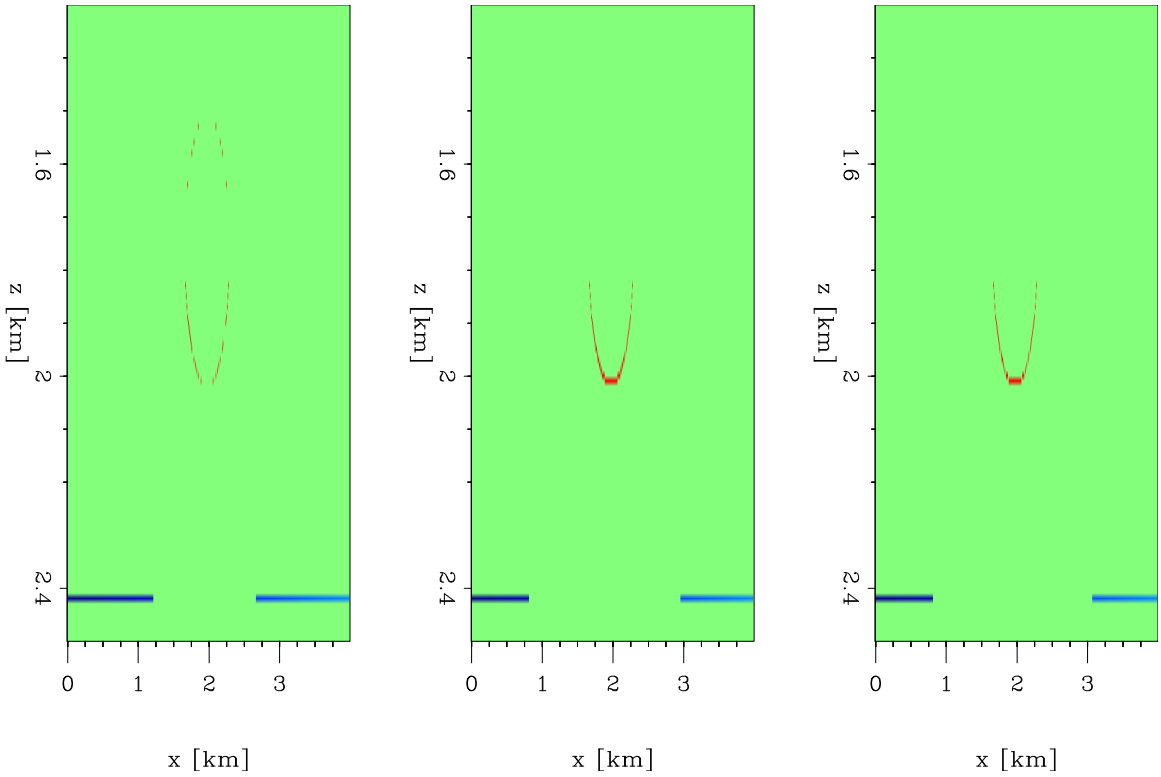

Figure 8: CASE 2: Difference between true model and velocity model for perturbed bottom reflector case after applying first update to ϕ using a line search method to find step size α. (Left) based on steepest descent, (middle) based on unconstrained Hessian inversion, (right) based on new Hessian inversion incorporating D operator directly. [**CR**]

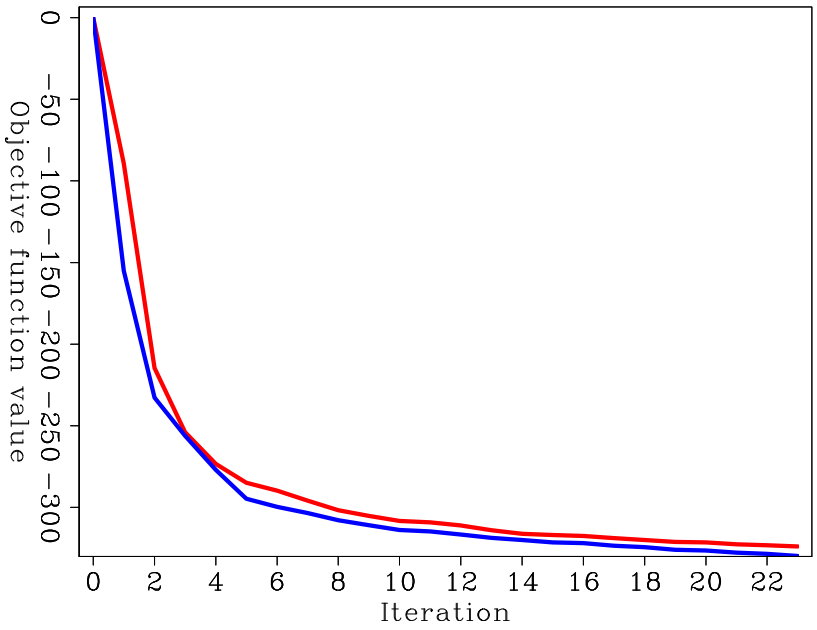

Figure 9: CASE 2: Objective function decrease for the two search direction inversions; Unconstrained search direction inversion (blue) and proposed method (red) using D operator. [**CR**]

CONCLUSIONS

The results of the two simple model cases show that we are able to get an improved search direction for the salt body boundary movement when we invert for our new model parameter updates $\triangle p$ using the new linearized derivative operator D defined earlier. We also show that the while the masking component seems to make the most significant difference when compared to the other two methods, there remains more to be learned about the significance of the scaling component of the D operator.

REFERENCES

Aster, R., B. Borchers, and C. Thurber, 2011, Parameter estimation and inverse problems. International Geophysics: Elsevier Science.

Biondi, E. and G. Barnier, 2017, A flexible out-of-core solver for linear/non-linear problems: SEP-Report, **168**.

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., B. Biondi, and R. Clapp, 2015, Domain decomposition of level set updates for salt segmentation, 1366–1371.

Dahlke, T., R. Clapp, and B. Biondi, 2016, Second-order updating in shape optimization for salt segmentation, 5369–5373.

Fichtner, A., 2010, Full seismic waveform modelling and inversion. Advances in

Geophysical and Environmental Mechanics and Mathematics: Springer Berlin Heidelberg.

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.

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