Chapter 4

Interpreter guidance for inclusion discovery

In this chapter I explain how one can modify the implementation of the level set inversion so that outside input can elegantly guide the inversion by means of expanding the footprint of the gradient update and by intelligently initializing the implicit surface. This improves the overall convergence rate, which is important because the operators used to find the search direction are computationally expensive. I finish by comparing inversion examples that showcase how these types of guidance can improve the convergence rate and allow for model updating that the unmodified level set algorithm is unable to replicate.

MOTIVATION

An advantage of the level set framework is that the implicit surface can be deformed at positions that are not on the current boundary, even to the extent that it 'punctures' the zero level set and creates a 'donut hole' or other topology. This can be useful for modifying salt bodies, since we may begin with a solid salt body, and then later deform the implicit surface such that the salt has inclusions inside it.

However, an underlying assumption of the gradient derived earlier is that the salt model only changes at the boundary of the original shape. This is because the salt is defined using a Heaviside function approximation, and so the gradient contains a $\delta(\phi_0)$ term that (in a practical sense) updates at the boundary only. If we only update along the current boundary in this manner, we stifle the possibility of topology changes, such as the discovery of inclusions that exist away from the current boundary. This is an inherent limitation of the theory I have derived thus far.

Furthermore, I have yet to discuss any methodology for initializing the implicit surface prior to inversion other than stipulating that the zero crossing position correlates to the boundary of the salt body with which we wish to begin. As long as the interior salt region is greater than zero, the Heaviside function indicates it as salt. This means that beyond the requirement of having the proper sign, there is flexibility in how one initializes that surface. I leverage this flexibility to create regions of preferential updating sensitivity.

MODIFICATIONS TO GRADIENT

Because the support of the gradient I previously derived is a subset of the actual objective function support (which is the entire model domain), I can expand the support of the masking term to preserve regions inside the current salt boundaries (possible inclusion regions) without affecting our ability to minimize our objective function. This means I can modify the masking term in our **D** operator to allow for updating outside of the boundary in regions selected by seismic interpreter guidance. I represent this modification with a new term $\hat{\boldsymbol{\delta}}(\boldsymbol{\phi_0}, \mathbf{G})$, which takes into account the interpreter guidance **G**:

$$\mathbf{D} = \left[\hat{\boldsymbol{\delta}}(\boldsymbol{\phi_0}, \boldsymbol{G})(\boldsymbol{c}_{\text{salt}} - \boldsymbol{b_0}) \quad \boldsymbol{I} - \hat{\boldsymbol{H}}(\boldsymbol{\phi_0}) \right].$$

The **G** term defines additional regions of our model where one wishes to allow updating to occur, for example, areas where an interpreter suspects inclusions to exist.

INITIALIZING THE IMPLICIT SURFACE USING INTERPRETER GUIDANCE

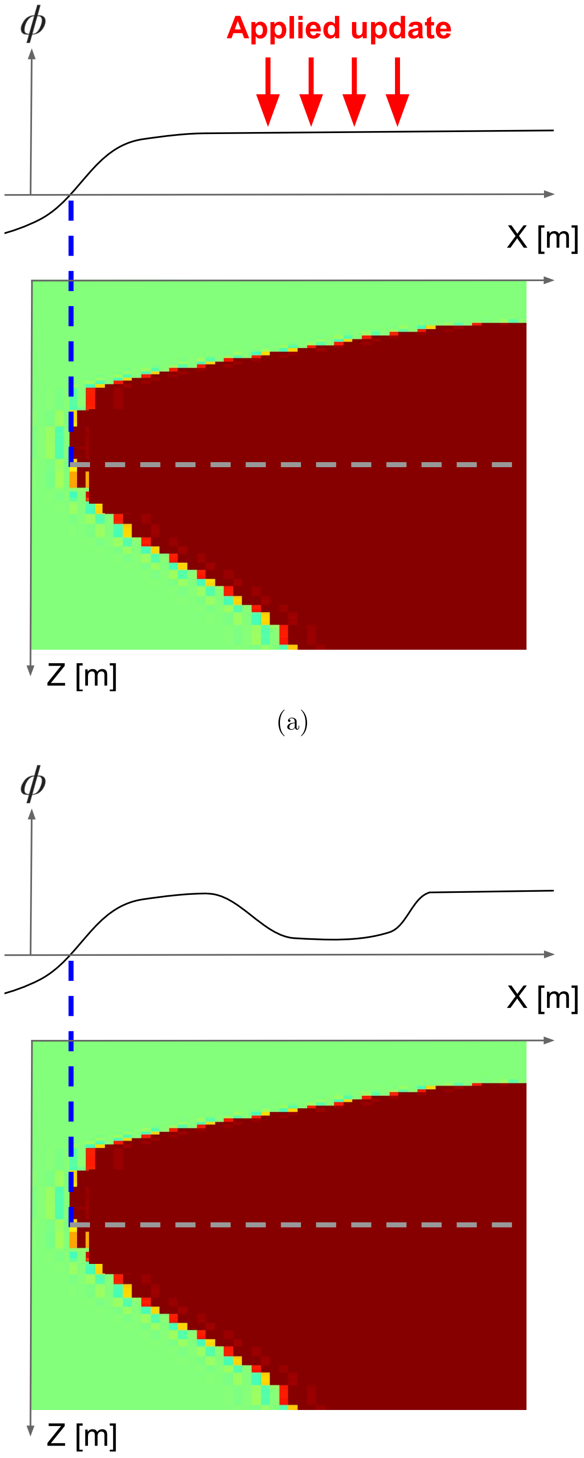
However, even if I expand the footprint of the region where updating is allowed, I can only discover an inclusion if the update perturbs the implicit surface below the zero-level set. This means the initial height of the implicit surface affects sensitivity to updating in different regions. For example, in Figures 4.1(a) and 4.1(b), we can see how an unguided implicit surface can receive an update that suggests an inclusion, but not realize that change in the resulting salt model.

However, I can initialize ϕ as shown in Figure 4.2(a) in order to increase sensitivity in that area of the salt model. This time, the same update is able to create the inclusion in the same iteration (4.2(b)). This is especially important when there are other model regions being updated that have a high impact on the data residual. In the case of Figure 4.1(a), the linesearch that chooses how strongly to apply the update will be most influenced by regions (like the top of salt) to which the objective function is inherently more sensitive, and may never allow for an inclusion update deeper in the model, even after many iterations.

The opportunity here is to set the height of the implicit surface according to how likely we believe that an inclusion is present at that position. This allows us to input a probabilistic mapping of inclusion likelihood into our initialization of ϕ. The same interpreter input G from before can be scaled and used to modify the implicit surface height appropriately. For the Sigsbee model example, Figure 4.3 shows the implicit surface with interpreter guidance applied, and Figure 4.4 shows it without. In Figure 4.3, we can see light-colored regions of the implicit surface that are not present in Figure 4.4, and have a lower value than the rest of the interior salt region. This means updates in this area will be more likely to break past the zero-level set contour and change the value of $\hat{H}(\phi_o)$ to create an inclusion in the model (if the data suggests that). In the case of erroneous interpreter guidance, the gradient may not create an inclusion, or if it does, it may fill it back in during later iterations.

DEMONSTRATION ON A GULF OF MEXICO MODEL

In this 2D synthetic example, I use a model where the inclusion is close enough to the boundary (Figure 4.5(a)) that there is a reasonable chance of the unmodified level set approach finding the inclusion without any of the interpreter guidance methods already described. Beginning with a model that has no inclusion (Figure 4.5(b)), I compare the unmodified method against the partially guided inversion (expanded gradient only) and the fully guided inversion (expanded gradient and implicit surface initialization). For both of the guided approaches, I expand the gradient by modifying



(a)

(b)

Figure 4.1: a) Example of salt model before update applied, and b) after update applied to simple ϕ surface with no sensitivity preferences. Note that the update applied in Figure 4.1(a) does not decrease the implicit surface below zero in Figure 4.1(b), resulting in no salt boundary change. [**NR**] chapter4/. inclusion-discovery1a,inclusion-discovery1b

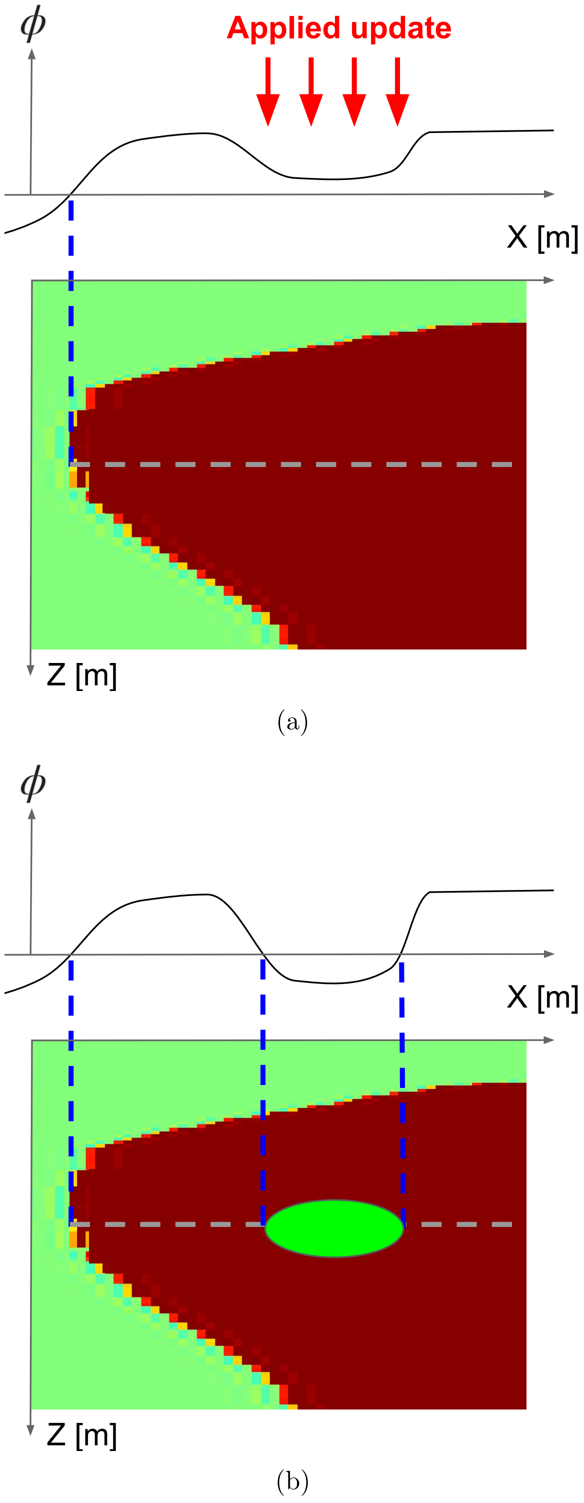


Figure 4.2: a) Example of salt model before update applied, and b) after update applied to ϕ surface with sensitivity preferences already incorporated into initial ϕ surface. Note that the update applied in Figure 4.2(a) decreases the implicit surface below zero in Figure 4.2(b), resulting in a salt boundary change. [**NR**]
chapter4/. inclusion-discovery2a,inclusion-discovery2b

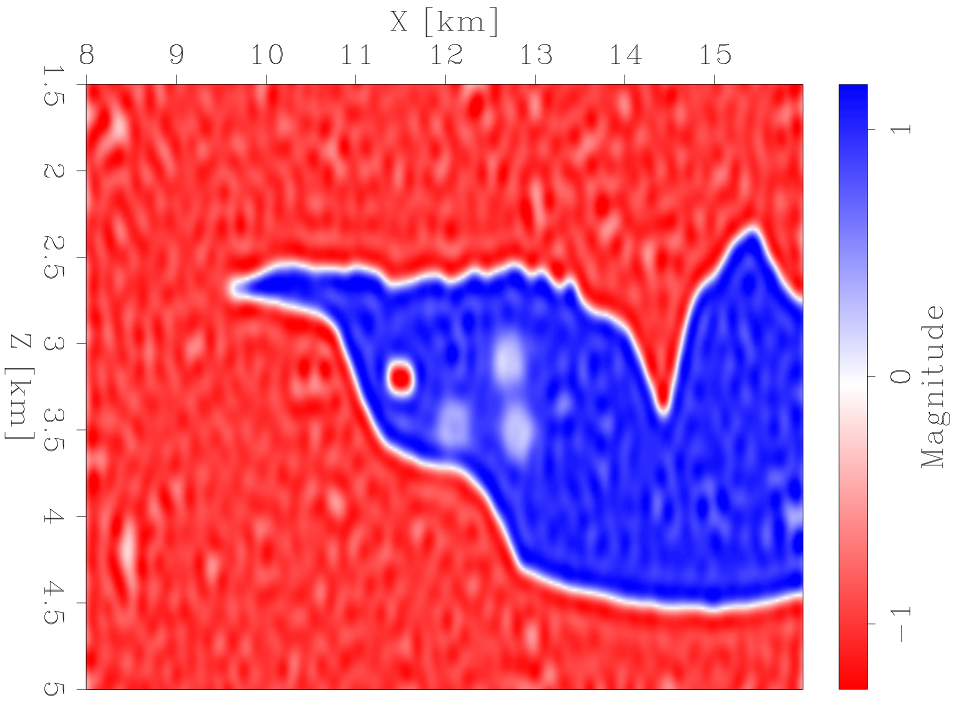


Figure 4.3: Initial implicit surface (guided). Note the three light colored regions where the implicit surface value has been reduced. [**ER**] chapter4/. initialphi-fullyGuided-sigsbee

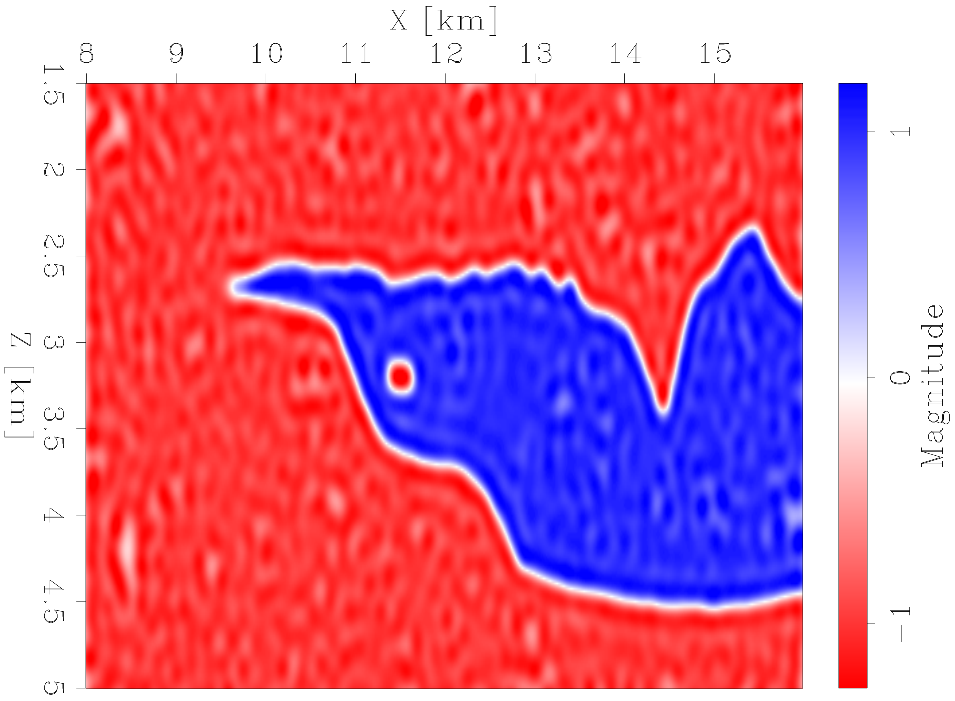


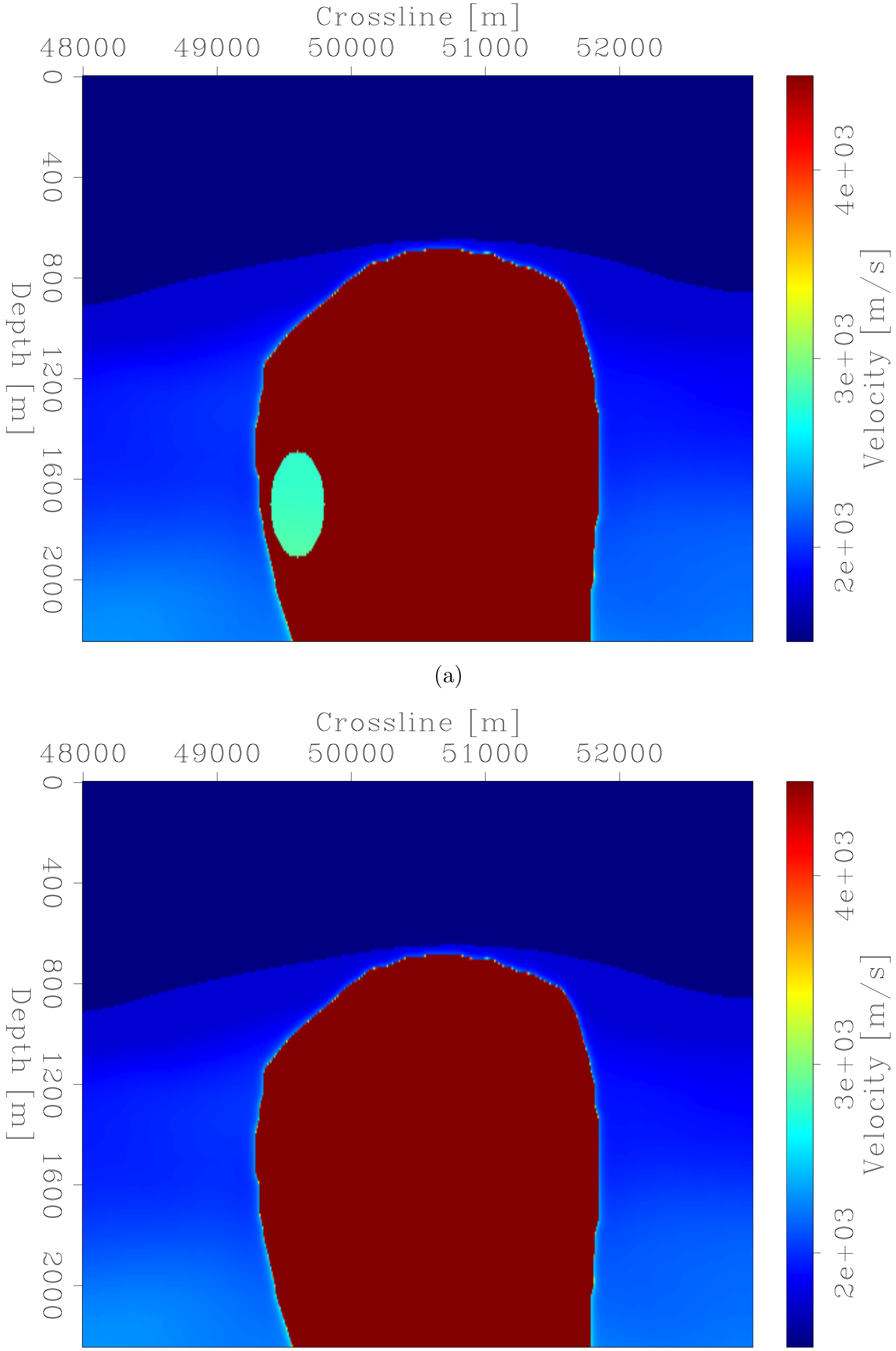
Figure 4.4: Initial implicit surface (unguided). Note the lack of the three light colored regions that are present in Figure 4.3. [**ER**] chapter4/. initialphi-unguided-sigsbee

the masking to allow for updating in the zone where I anticipate that an inclusion exists (Figure 4.6(a)). The unguided approach has no expanded gradient (Figure 4.6(b)). In the fully guided example, I also initialize the implicit surface using the same interpreter guidance so that it has a lower value where I anticipate an inclusion (Figure 4.7(a)), while the unguided and partially guided approaches do not use this guidance for initialization of the implicit surface (Figure 4.7(b)).

As expected, I find that the guided inversion approaches perform much better (Figures 4.8 and 4.9) in terms of reducing both the model residual and data residual norms. In the partially guided approach, the data guides the inversion to the correct inclusion shape regardless by means of the expanded gradient (see Figure 4.10(b). However, the implicit surface initialization used in the fully guided approach (Figure 4.10(c)) further improves the convergence rate. Meanwhile, the unmodified method begins to approach the true answer only after 65 iterations (Figure 4.11). In a case where the true inclusion position is further from the starting salt boundary, I would likely have an even worse result using the unmodified approach.

CONCLUSIONS

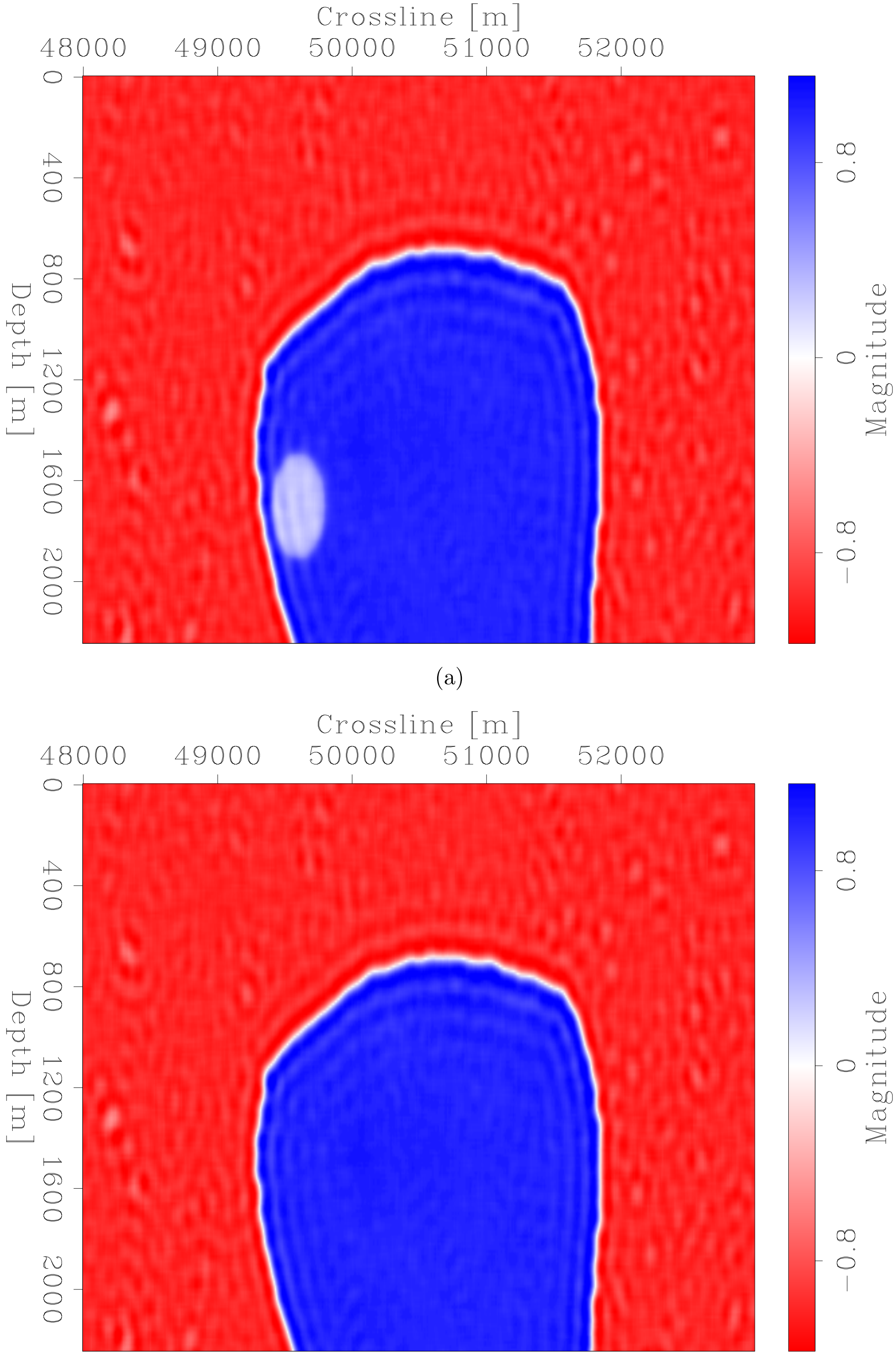
I discuss the limitations of the standard level set formulation and how it constrains our ability to create salt models with inclusions. I then introduce the idea of using a likelihood map generated by interpreters as guidance for the inversion. This guidance is used to expand the level set gradient footprint, as well as initialize the implicit surface to create areas of higher sensitivity to updating. Using 2D synthetic examples, I show how the expanded gradient and the implicit surface initialization both contribute to speeding up the convergence.



(a)

(b)

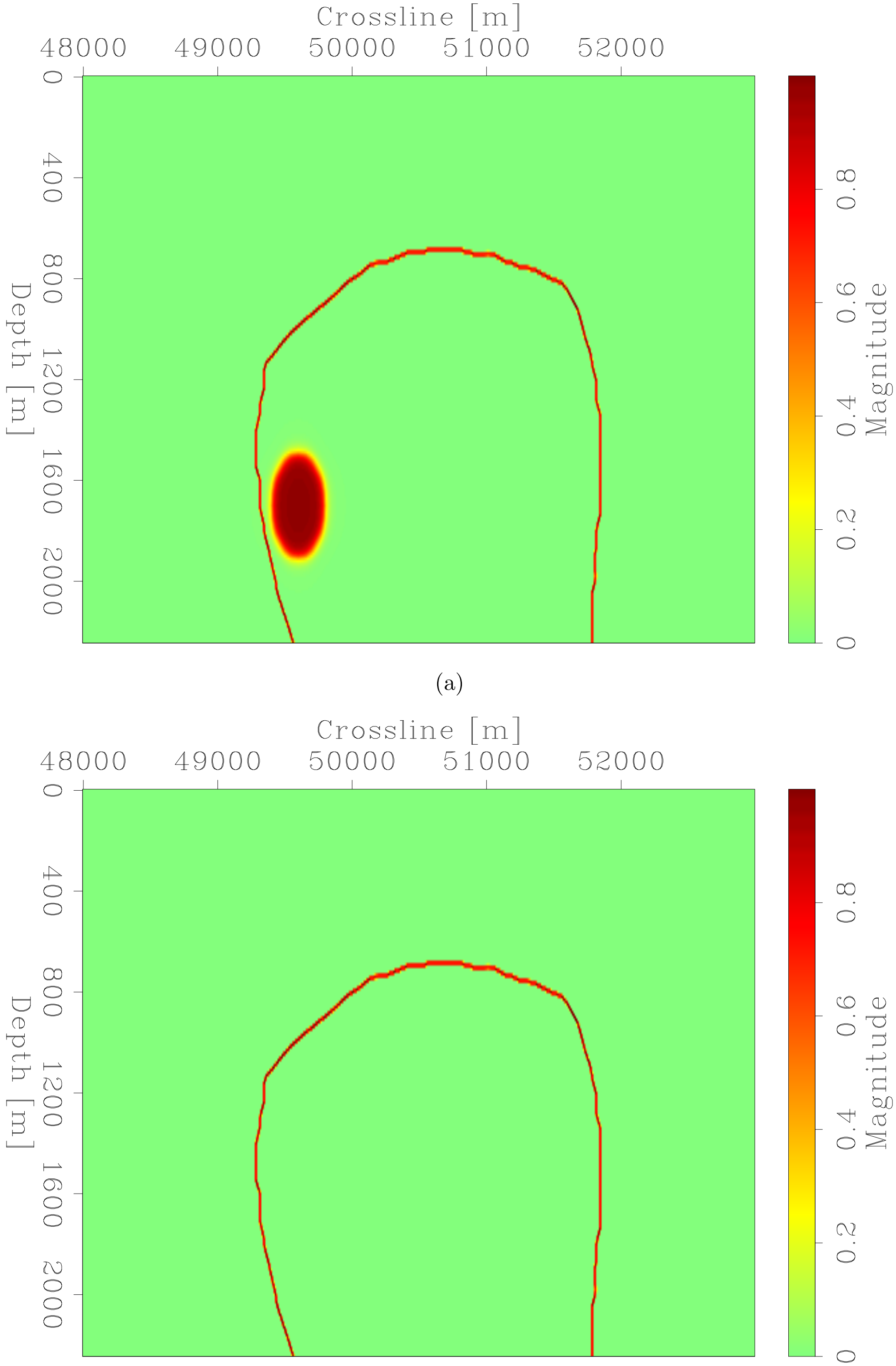
Figure 4.5: True model a) and starting model b) and used for all tests. **[ER]**
chapter4/. truemodel-gom,startingmodel-gom



(a)

(b)

Figure 4.6: Implicit surface with interpreter guidance initialization a) and without it b). [**ER**] chapter4/. initialphi-fullyGuided-gom,initialphi-unguided-gom



(a)

(b)

Figure 4.7: Masking term ($\hat{\delta}(\phi_0, G)$) for guided gradient methods (a), and masking term ($\delta(\phi_0)$) for the unguided approach (b). [**ER**]

chapter4/. guidedMask-gom,unguidedMask-gom

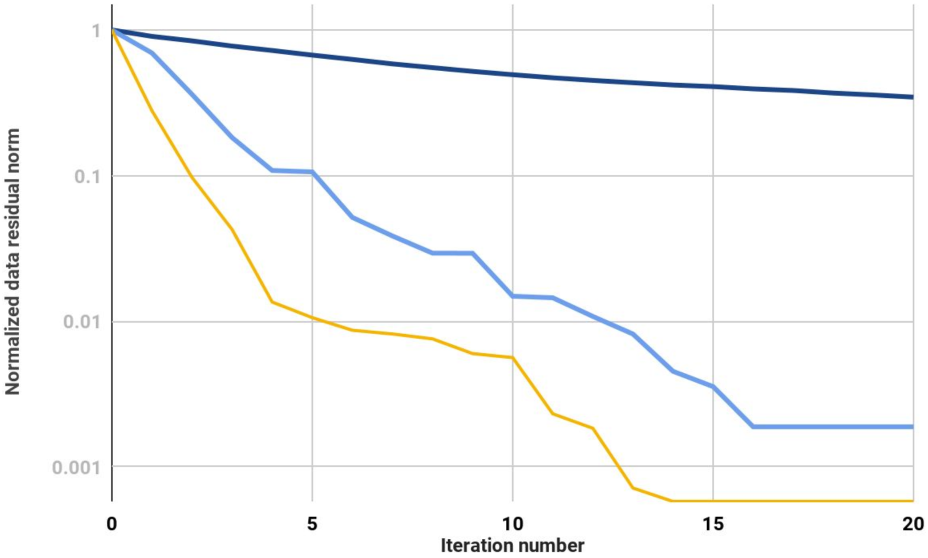


Figure 4.8: Data residual norm for level set inversion with unmodified approach (dark blue), partially guided approach (light blue), and fully guided approach (yellow). [**CR**] chapter4/. gom-guidanceVsnoguidance-dataNorm

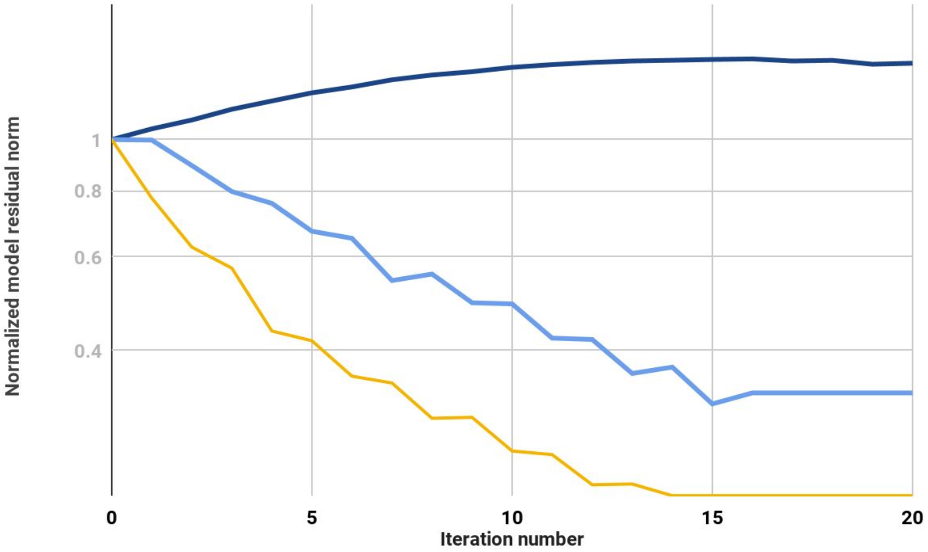
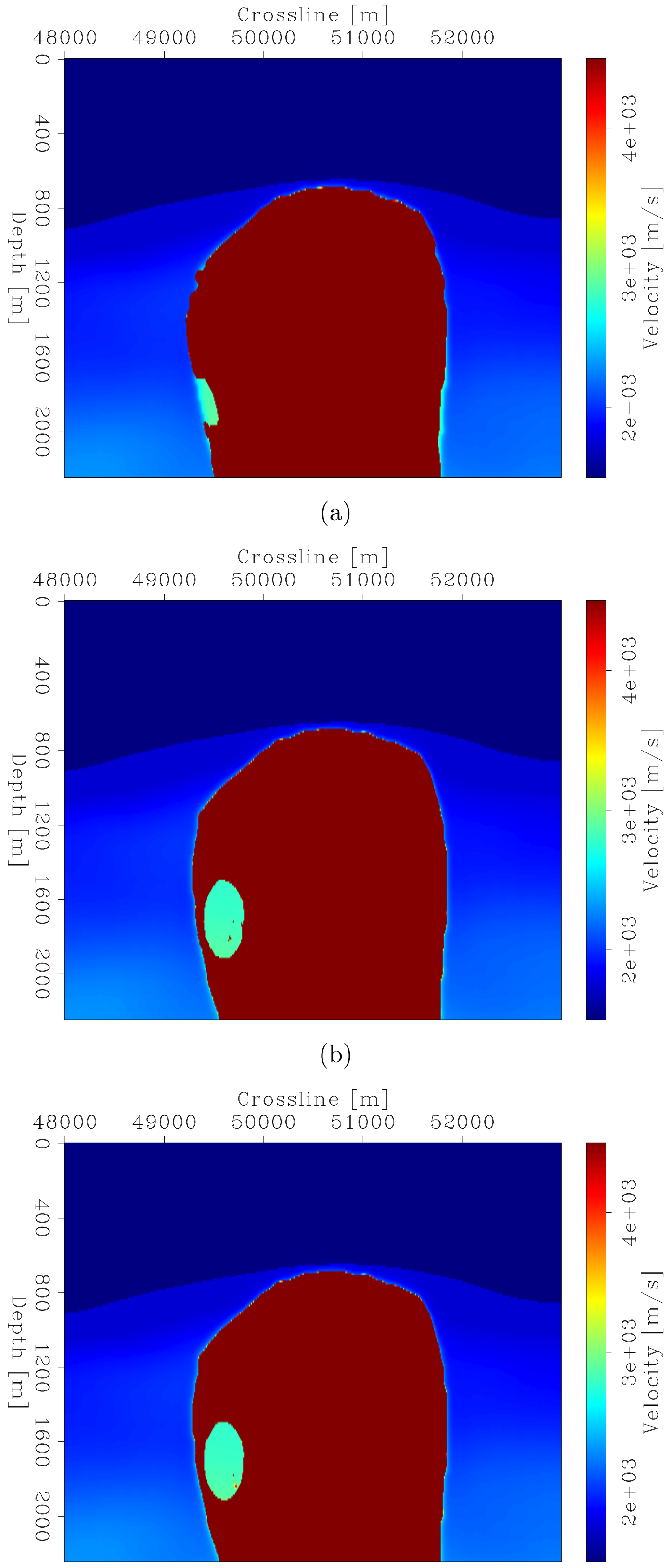


Figure 4.9: Model residual norm for level set inversion with unmodified approach (dark blue), partially guided approach (light blue), and fully guided approach (yellow). [**CR**] chapter4/. gom-guidanceVsnoguidance-modelNorm



(a)

(b)

(c)

Figure 4.10: Inverted model after 15 steepest descent iterations using unguided approach a), partially guided approach b), and fully guided approach c). **[CR]** chapter4/. model-15-unguided,model-15-partiallyGuided,model-15-fullyGuided

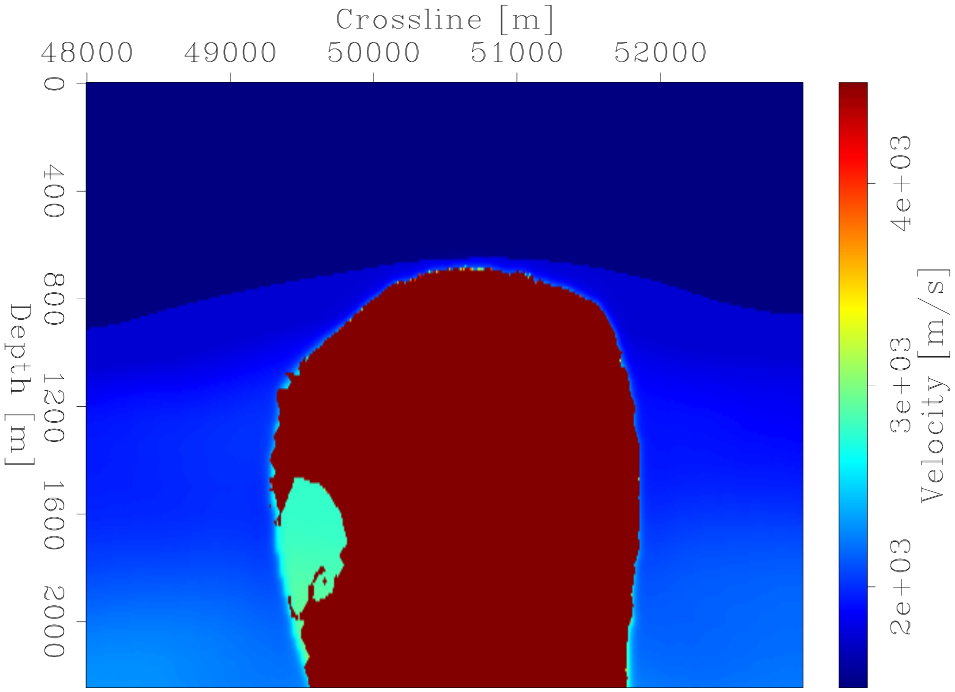


Figure 4.11: Inverted model after 65 steepest descent iterations using unmodified approach. [**CR**] chapter4/. model-65-unguided