Representing salt bodies with radial basis functions

Taylor Dahlke

ABSTRACT

In this work we show how radial basis functions can be used to sparsely represent the implicit surface used to represent salt bodies. We show that this methodology is effective even when the model parameter reduction is roughly 2% of the original model size. This is important to making shape optimization effective for 3D velocity models. When the Hessian of a modified FWI objective function is used for shape optimization, we must invert the Newton system for the search direction. When using iterative methods like conjugate gradient for this, the reduction in parameters improves the speed and stability of this inversion.

INTRODUCTION

Previously, Kadu et al. (2016) demonstrated how radial basis functions (RBFs) can be used to help define salt features using level sets. In their work, they chose to use a regular gridding for the RBF centers. This has the advantage of allowing the same resolution of updates to all areas of the salt model. Assuming that a reasonably good initial salt pick is chosen, we can further assume that any changes to this salt boundary will occur within its vicinity. Following this, we prefer to concentrate our resolution (and subsequently the RBF centers) around the areas where we actually expect updating to occur, i.e, the edges of the initial salt body picks. For this reason, we apply an approach that builds a probability distribution around the edges of the initial salt picks, and then randomly assigns RBF centers based on this probability distribution. This builds a 'cloud' of RBF centers that is densest where we expect the updates to occur and gives us enhanced resolution in these areas using far fewer parameters than a regular grid approach would require to achieve the same resolution. In this paper we first show how we create this probability distribution. Then we describe the linearization that we used to perform the non-linear inversion, and last we show results that demonstrate the efficacy of the method on the Cardamom salt model.

INVERSION

To use a sparse representation of our implicit surface as the model in an FWI type workflow, we first have to begin with some initial model. We find this by starting with some initial binary guess of salt/no-salt in the spatial field such as the salt overlay shown Figure 1. Next we use a non-linear inversion workflow to find the RBF parameters that best fit that 2D spatial salt mapping. We define the function that maps from the RBF parameter (λ) space to the implicit surface (ϕ) space as:

$$\phi(\lambda; \epsilon, r) = \sum_{i}^{\mathrm{N}\lambda} \lambda_i \exp^{-(\epsilon r)^2}, \tag{1}$$

where r is the radial distance from the RBF center and ϵ is a constant that influences the taper of the Gaussian kernel. With this formulation of the implicit surface, we can then map from the implicit surface (ϕ) to a velocity field (m) using the non-linear Heaviside function (H):

$$m(\lambda) = H(\phi(\lambda)). \tag{2}$$

We can approximate the Heaviside as a differentiable function $\hat{H}$, which allows us to take its derivative as the smooth Dirac-delta like term $\hat{\delta}$, and thus find a gradient for equation 2:

$$\triangle m = \hat{\delta}(\phi(\lambda)) \sum_{i}^{\mathrm{N}\lambda} \exp^{-(\epsilon r)^2} \triangle\lambda. \tag{3}$$

With this, we can code up linear forward and adjoint operators that allow us to transform between perturbations in the velocity and RBF parameters. Using this with the non-linear forward (equation 2), we can use a non-linear solver to invert for the correct RBF parameters (λ). I used a python-based, out-of-core solver developed by Ettore Biondi and Guillaume Barnier to perform the inversion itself (Biondi and Barnier (2017)).

APPLICATION

In order to preserve efficiency of the algorithm, we keep the RBF parameter ϵ constant, and thus use the same Gaussian function for each RBF kernel. This means we only solve for the weights (λ) applied to each of these kernels. Because this RBF kernel is static, we can calculate it beforehand. Since the value of the RBF diminishes with radial distance, we only use a small section of the full RBF that is in a region of relatively close radius to the center. Far-radius regions of the RBF are negligibly small, and so the computational cost of storing and using a larger kernel is a fruitless exercise.

Parameter choices

Because we choose to define our RBF kernel before inversion, we need to find parameter values for ϵ, for the size of the kernel footprint, and for the density of the RBF centers themselves (see Figure 3 showing probability mapping and Figure 4 showing the resulting RBF centers). These must be determined manually beforehand, and the interplay between the parameters must be analyzed. For example, if we choose a ϵ that is too large, the RBF will taper off rapidly. If a regular gridding of RBF centers is used, it is relatively easy to calculate a ϵ value that will allow for full spatial coverage via overlapping RBF functions. However, since we choose a randomized centering for each RBF, we need to check to see that our ϵ value is small enough that the summation of all RBF functions has complete coverage over the spatial domain we are working with. However, if it's too small, then we smooth out the radial basis functions, and thus decrease the resolution of the implicit surface we create. Alternatively, we could keep our ϵ value and RBF centers the same, and increase the footprint of the kernel itself. We show the impact of varying ϵ with a fixed kernel footprint and RBF centers in Figures 2a and 2b. In Figure 2a where we use $\epsilon = 2.25$, we get RBF kernels that taper off quickly, leaving little to no overlap in areas where the RBF are more sparsely centered. However, if we decrease ϵ to 0.25, we reduce this tapering and get full coverage in the area of interest (Figure 2b). This does reduce the precision slightly around the boundaries themselves since the smoother RBF mean we have less resolution in our aggregated implicit surface. Overall, the parameters that tune the minimum probability of a center occurring, ϵ, and kernel footprint size must be balanced to achieve both full coverage, low number of parameters (RBF centers), and resolution of the resulting aggregate surface. This tuning can be done relatively quickly by doing simple inversions on small test sections of the full model.

Cardamom salt model

The Cardamom field is in the Gulf of Mexico, and lies about 360 km south-west of New Orleans, Louisiana in approximately 830 m of water. The reservoir itself sits beneath thick layers of salt in rock more than 6 km below the sea floor. We choose a section of the velocity model provided to us by Shell that has a notable salt protrusion in it as an example.

Beginning with a salt model from the Cardamom field dataset (Figure 1), we are able to build a probability density map that favors putting RBF centers near the original picked boundary (Figure 3). From this, we are able to generate random RBF positions (Figure 4). Using these RBF centers, we then perform a conjugate gradient inversion to find the proper weighting of the RBF kernels in order to best fit our starting model. The inversion converges relatively quickly (Figure 5), and produces a result that is relatively close to the desired matching model. Figure 2b shows that the matching model and the resulting inverted model generated from sparse RBF parameters are quite similar.

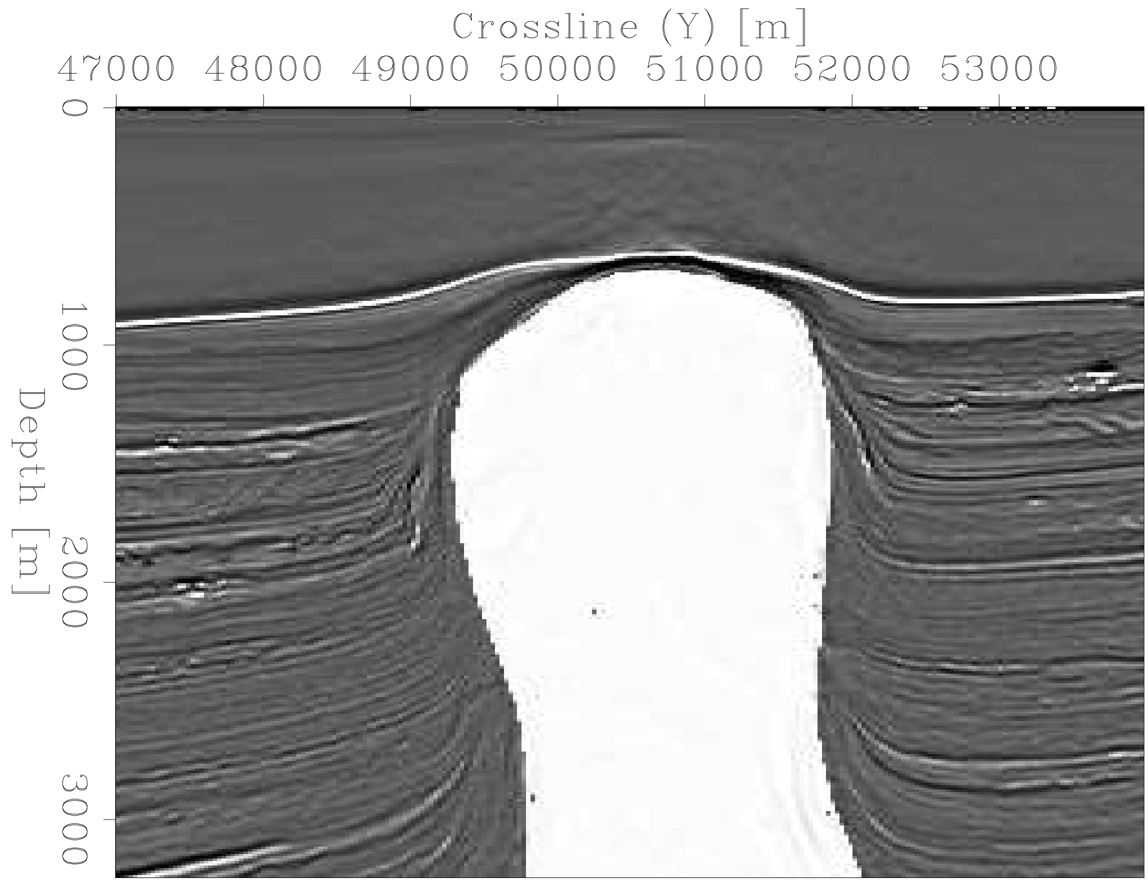

Figure 1: Overlay of salt model used by Shell and the corresponding RTM image. [**ER**]

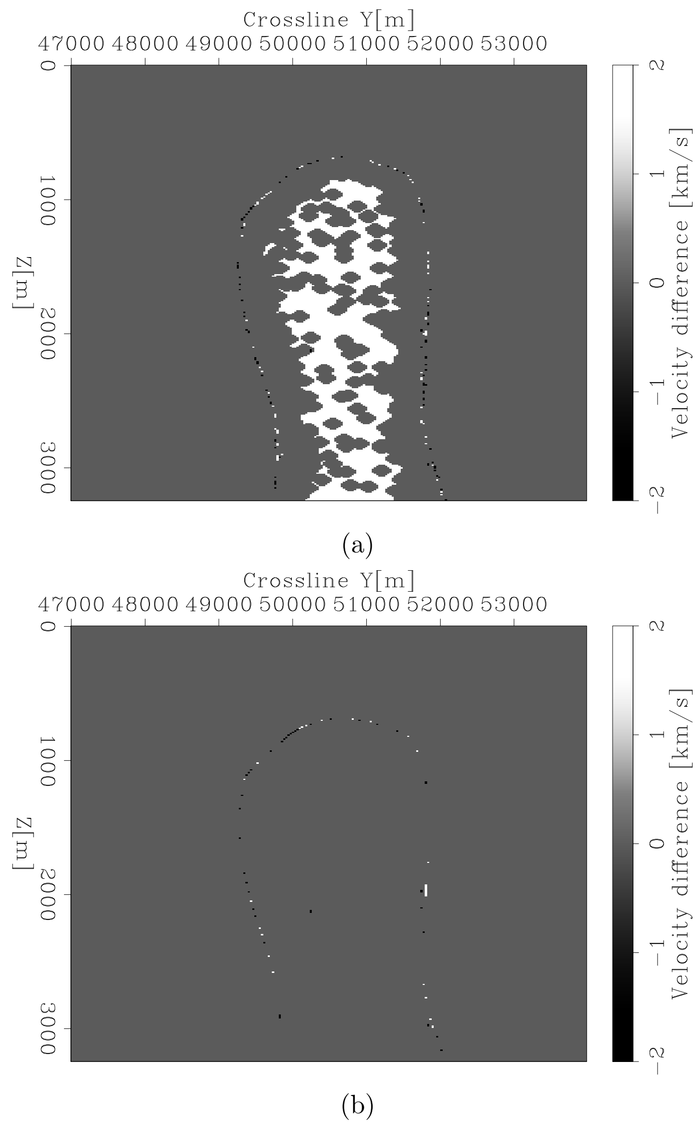

Figure 2: Differences between original salt model and the resulting model produced by the radial basis function representation. (a) shows fitted salt model using $\epsilon = 2.25$ value, while (b) shows fitted salt model using $\epsilon = 0.25$ value. Both cases used 98% fewer model parameters than the original full-grid scheme. Background velocity is 2.5 km/s and salt velocity is 4.5 km/s. [**ER**]

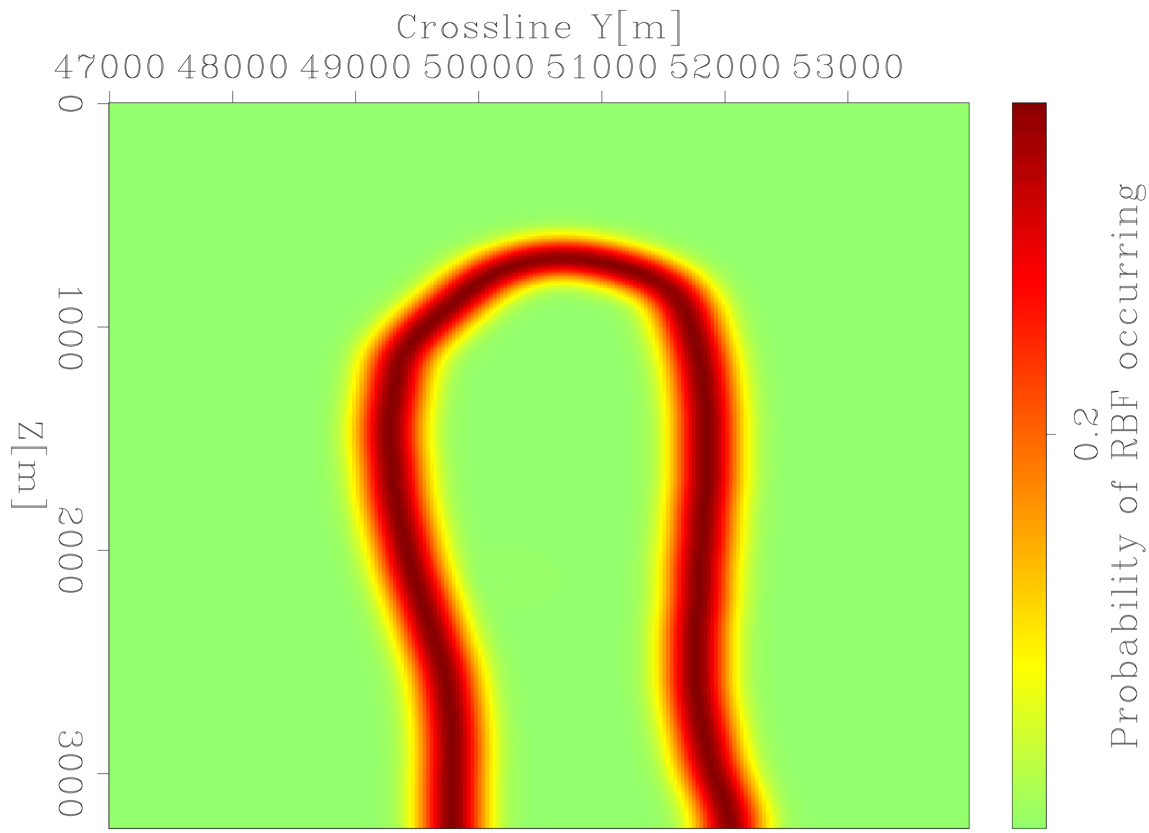

Figure 3: The probability distribution that was used to randomly position the radial basis function centers. [**ER**]

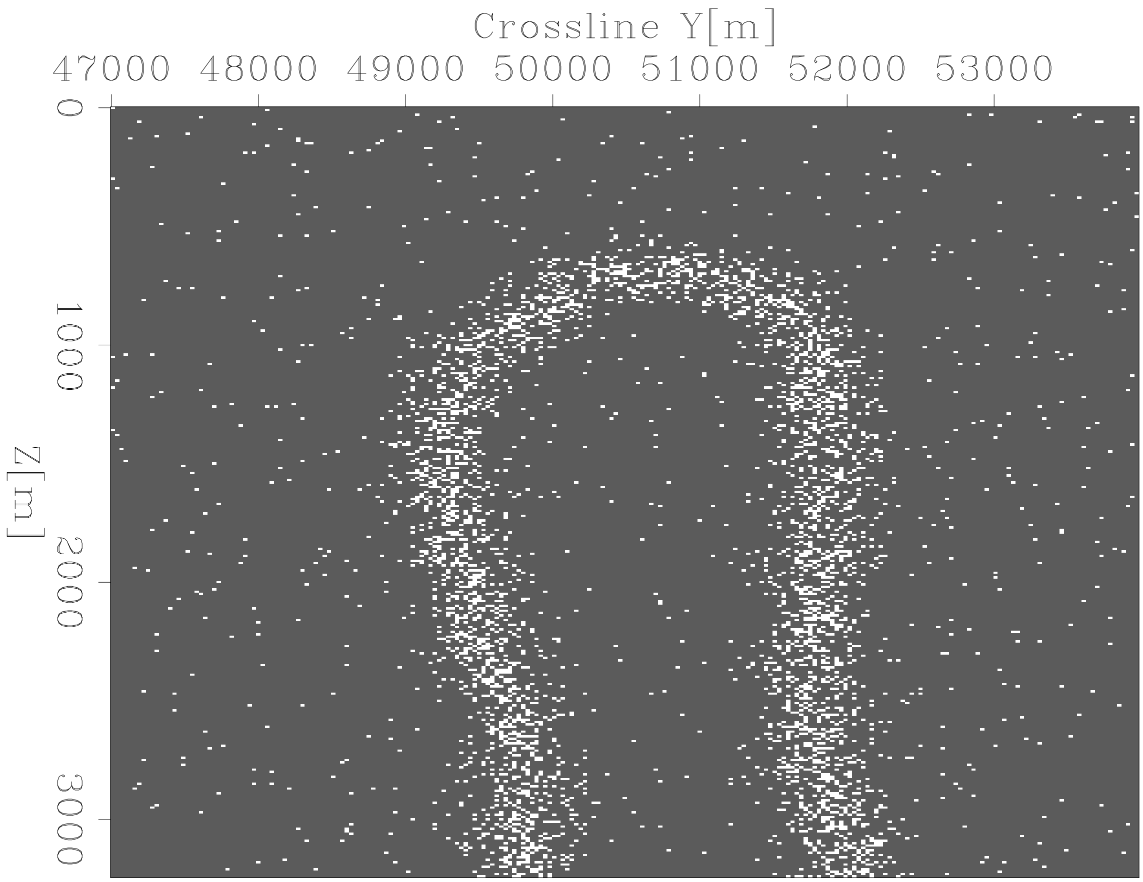

Figure 4: Center points for radial basis functions used to construct the implicit surface. [**ER**]

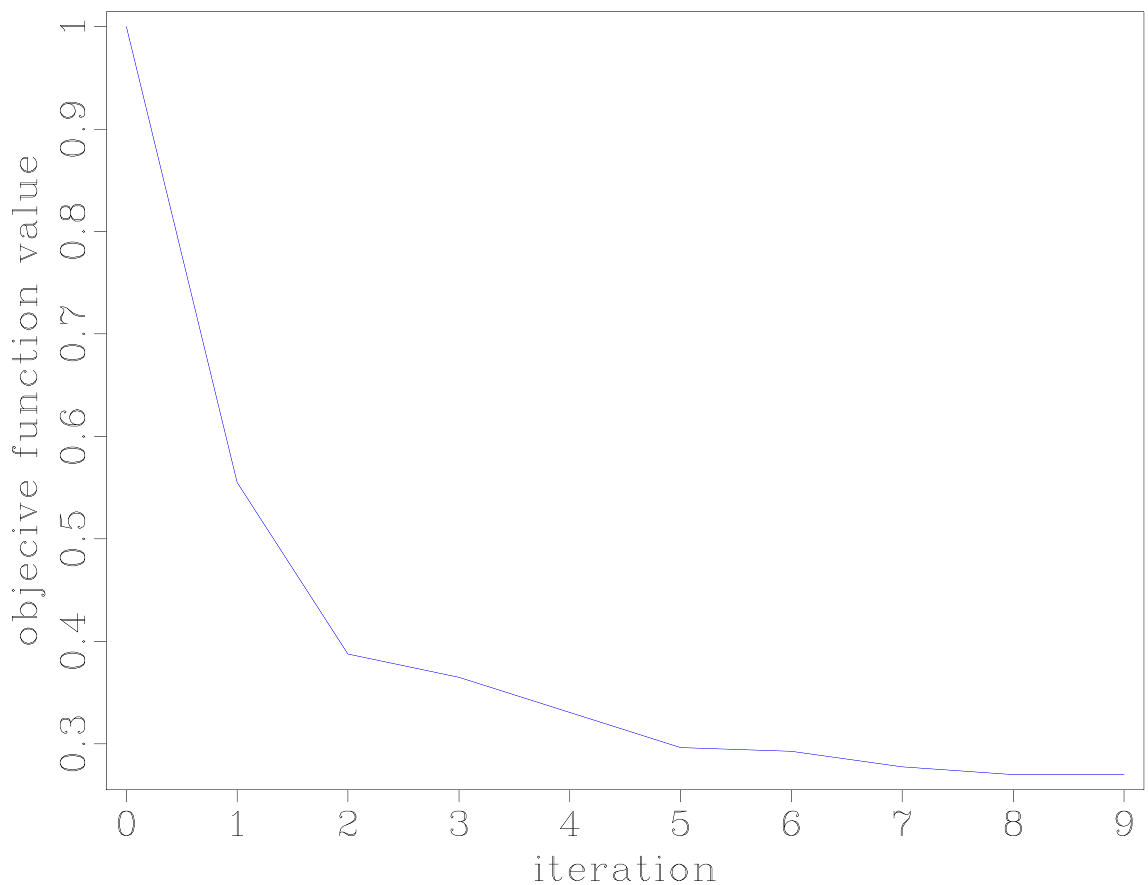

Figure 5: Objective function from the non-linear inversion used to find the RBF parameters. [**ER**]

CONCLUSIONS

In this paper we showed that radial basis functions can be used to significantly reduce the number of parameters necessary to represent a 2D salt model. For the Cardamom example we achieve 98% reduction in model parameters while still closely matching the original boundary described in the fully-gridded space. The benefit of using random locations for RBF centers is that we can achieve a higher resolution implicit surface than if we were to use a regular gridded model for the same number of model parameters. However, we need to take the time to choose our parameters like ϵ carefully in order to maintain full coverage, high model reduction, and final surface resolution. In future work, I hope to show how shape optimization can incorporate this approach on a larger 3D cube of the same Cardamom velocity model for the purposes of model refinement according to the FWI objective function.

ACKNOWLEDGEMENTS

I would like to acknowledge Shell Oil for providing the RTM data and salt model used in this work. I would like to thank Bonnie Jones, Jan Douma, and their affiliates at Shell for their help in accessing and working with this dataset.

REFERENCES

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

Kadu, A., T. van Leeuwen, and W. A. Mulder, 2016, Salt reconstruction in full waveform inversion with a parametric level-set method: CoRR, **abs/1610.00251**.