Mismatches between physics and operators for Least Squares Migration: comparison between Kirchhoff and RTM

Daniel Trad

ABSTRACT

In this report, I use least squares Kirchhoff (LSK) and reverse time migrations (LSRTM) to analyze how least squares techniques behave in the presence of mismatches between the physics that produce the data and the operators we use to predict them. By implementing both approaches with similar algorithms I show that noise is added during the inversion because of contributions to the residuals from poor operator approximations. When applying LSRTM with exact operators the method works really well, and we don't observe any noise. However, when the operators differ, errors start to accumulate very quickly from inversion. Examples for Kirchhoff migration with different types of traveltime tables show that noise in LSK is produced by multipathing and shadow zones and inaccurate traveltime calculations. I interpret this noise as a consequence of calculating the gradient of the inversion as a mapping from a wrong residual space to the model space. I interpret this noise as a consequence of assuming that the prediction operator is linear and accurate when is not. Finally, I analyze two well-known mechanisms based on data and model weights to control this problem: regularization by adaptive residual (data space) and image (model space) filtering.

INTRODUCTION

Inversion is a very common technique in seismic processing and other branches of science, consisting of creating a model that can predict acquired data. This approach has three main components, the model $\mathbf{m}$, the data $\mathbf{d}$, and the operator $\mathbf{L}$ that connects them (Figure 1). Choosing the kind of model defines the operator. In seismic we use many types of models. For example, in Full Waveform inversion (FWI) we use some function of velocity, like slowness square. In Least squares migration (LSMIG) we use a model that resembles reflectivity (Kirchhoff modeling), or velocity perturbations (scattering or Born modeling). When the goal of our processing is to predict data, for example in noise attenuation or interpolation, we have flexibility in choosing the model and the operator as long as they predict the part of the data we want. In FWI and LSMIG, on the other hand, we need to make sure that the operator resembles the physics that generated the data because the goal is the model itself. In other words, a wrong operator may predict the data correctly and the model incorrectly.

What is common to all these techniques is the approach we use to calculate the model that best predict the data: we minimize the residual square or average difference between data and predictions. This minimization always proceeds in some form of optimization, which depends on whether the operator represents a linear transformation (the operator is independent of the model we seek), or non-linear (the operator changes as we improve our estimation of the model). This is, in fact, the main difference between FWI and LSMIG. By keeping the operator independent of the model in LSMIG, the inversion procedure is simplified, but at the same time, we make the assumption that all error present in the

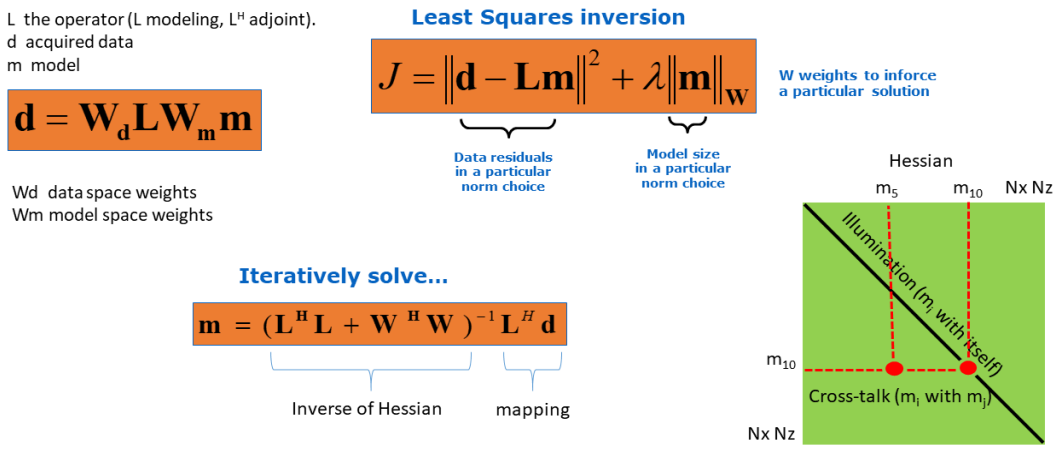

FIG. 1. Least squares migration and Hessian deconvolution

residuals can be mapped to and corrected by changes in the model.

The consequence of this assumption is far more reaching than it seems because residuals contain not only parts that can be improved by changing the model, but also parts that are not predicted by the model-operator combination, and parts that are predicted but with wrong amplitude and phase. Since the least squares mechanism is designed to reduce error in the prediction, it will try to use any component of the model space to predict something similar to the input data. For example, an aliased operator can use low frequencies to produce data with higher frequencies. A mismatch between physics and mathematics will not only prevent the algorithm to produce zero residuals but also will produce models with artifacts or noise. In this work, I will discuss this problem and show some ways of limiting its effects.

LEAST SQUARES MIGRATION/MODELING

There are two different ways to define least squares migration, iterative and non-iterative. In this work, I focus only on the iterative version. In LSMIG given the acquired data $\mathbf{d}$, we want to find a model $\mathbf{m}$ to predict them with minimum least squares error (Nemeth et al., 1999). This process involves the choice of a mathematical operator $\mathbf{L}$,

$$\mathbf{d} = \mathbf{Lm}. \tag{1}$$

This formulation is similar to what we use in many other problems in geophysics (Claerbout, 1992), but the meaning of the model $\mathbf{m}$ varies. The model for LSMIG is usually reflectivity or velocity perturbation, making $\mathbf{L}$ to be either Kirchhoff or Born modeling respectively. This operator is assumed to be independent of the model by using a linearized wave equation. This is different from FWI, where the model is velocity or slowness, and the operator changes with the model as well. We will see later that this choice imposes some difficulties for the application of this technology. However, before that, we can make our problem more general by extending the modeling operator with other operators that can be used to modify data and model spaces. A general least squares formulation is:

$$\begin{aligned} &\text{minimize } \|\mathbf{W}_m\mathbf{m}\|_p^p \\ &\text{subject to } \|\mathbf{W}_d(\mathbf{d} - \mathbf{Lm})\|_q^q = \phi_d \end{aligned} \tag{2}$$

where ϕ_d is some estimate of the noise level in the data plus a residual due to the failure of the proposed model to explain the data. p and q indicate that different norms can be applied to measure the norm of vectors. $\mathbf{W}_d$ could be a matrix or vector of data weights, often a diagonal matrix containing the inverse of the standard deviation of the data, but more generally it could be any kind of filter that leaves out bad data. $\mathbf{W}_m$ is an operator of model weights that can be customized to enhance our preferences regarding the model. Regarding this equation, there are a few points to notice:

- The term ϕ_d represents a global measure of the failure to explain the data. In some problems, it may be necessary to account for a local measure of the misfit instead of global.
- $\mathbf{W}_d$ is a tool to diminish the effect of bad data. The term "bad data" refers in general to wrong measurements. However, "bad data" may also be "good data" that can't be predicted by a "bad operator".
- $\mathbf{W}_m$ can account for certain types of undesired features of the model, as is usually considered in standard regularization, but also, it can be used to change the mapping between data and model space. In this sense, this operator is more a preconditioner operator than a regularization term.

By minimizing the cost function obtained from equations (2) we can obtain the model that better approximates the desired solution by solving the system of equations

$$(\lambda \mathbf{W}_m^T \mathbf{W}_m + \mathbf{L}^T \mathbf{W}_d^T \mathbf{W}_d \mathbf{L})\mathbf{m} = \mathbf{L}^T \mathbf{W}_d^T \mathbf{W}_d \mathbf{d}, \tag{3}$$

where λ is a trade-off parameter that gives different weights to the misfit and model constraints. To eliminate this parameter and simplify this system of equations, we incorporate these weights and filters into a general operator $\tilde{\mathbf{L}} = \mathbf{W_d L W_m^{-1}}$, which it is just a change of variables in the fundamental equations, with $\tilde{\mathbf{m}} = \mathbf{W}_m \mathbf{m}$ and $\tilde{\mathbf{d}} = \mathbf{W}_d \mathbf{d}$. This is equivalent to a right and left preconditioning, transforming the modeling equation (1) to

$$\mathbf{W}_d \mathbf{d} = \mathbf{W}_d \mathbf{L} \mathbf{W}_m^{-1} \mathbf{W}_m \mathbf{m} \tag{4}$$

and the system of equations (3) to

$$(\lambda \mathbf{I} + \tilde{\mathbf{L}}^T \tilde{\mathbf{L}})\tilde{\mathbf{m}} = \tilde{\mathbf{L}}^T \tilde{\mathbf{d}}\,. \tag{5}$$

The system (5) can then be solved by setting $\lambda = 0$ (no regularization) and limiting the number of iterations. This is a simplifies the problem because now the operator contains all our choices. Also, this allows us to a generic numerical solver, for example an iterative conjugate gradient method (CG).

LEAST SQUARES MIGRATION DEFINED IN TERMS OF MIGRATION

In Figure 1 we can see that LSMIG is the operation of the Hessian inverse acting on a standard migration (referenced as "mapping" in the figure). Therefore, to understand the differences of LSMIG and migration we have to look at the Hessian. Defining the

true solution as the model that can predict the data exactly, migration gives the true model distorted by the Hessian:

$$\mathbf{m_{migration}} = \mathbf{L^T d} = \mathbf{L^T L m}. \tag{6}$$

On the other hand, the least squares solution is

$$\mathbf{m_{ls}} = (\mathbf{L^T L} + \lambda \mathbf{I})^{-1} \mathbf{L^T L m} \tag{7}$$

In general, $\mathbf{m_{ls}}$ is closer to the true model than $\mathbf{m_{image}}$ because it removes the blurring effect of the Hessian. The Hessian is often diagonal dominant, so the real improvement achieved by LSMIG depends on how large the off-diagonal terms of the Hessian are. So, what are these off-diagonal terms? Schuster (2017) shows that the Hessian corresponds, for a linear or linearized problem, to a scaled version of the model covariance. Therefore the diagonal elements of the Hessian are proportional to the variance of each element of the model. Because the Hessian contains all the mapping information of the seismic experiment, the variance of the model contains information on how well we can see each model component (that is the illumination), and it is often used to correct for poor illumination and improve the image without going through the work of calculating the full Hessian. The off-diagonal components are cross-covariances between different model components. They tell us how similar will be the effect of changing one model component to the effect of changing the other. In other words, they tell us how well the data, and therefore the cost function, can distinguish effects for different model components. A well-designed experiment, where data are very sensitive to different components of the model, will have small off-diagonal elements and therefore LSMIG will not add much to our results. If the experiment is deficient, for example, because of poor sampling, the use of LSMIG, should be capable of removing interferences across model components and producing good results in spite of the complications. This is the big promise of LSMIG, one that it has been really difficult to take advantage in everyday imaging done by the industry (interpolation is used much more often than LSMIG).

IMPLEMENTATIONS FOR KIRCHHOFF AND REVERSE TIME MIGRATION

I use the Conjugate Gradients framework to design both LSK and LSRTM as a linearized inversion of forward-adjoint operator pairs. Since the inversion formulation is identical, all differences are in the modeling operators, which leads to different degrees of mismatch between physics and data. Next, let us discuss the form that the adjoint-pair operators take for both Kirchhoff and RTM.

LSK implementation

At the heart of every Kirchhoff modeling operator, the fundamental operation that creates data from a reflector is given by:

$$data(mid, h, time) += W(V, h, depth, time) \times model(mid, h, depth) \tag{8}$$

where mid is midpoint, h is offset, V is velocity and W is an expression that connects amplitudes and that is subject to different approximations (Dellinger et al., 1999). The connection between $time$ and $depth$ is obtained through ray tracing (depth migration) or parametric methods (time migration). The adjoint operator for Kirchhoff modeling is close,

but not equal, to Kirchhoff migration. To find out the exact form of the adjoint-pairs it helps to think of any linear operator as a matrix vector multiplication, keeping in mind that the adjoint of a matrix operator is its complex conjugate. Therefore we can write the adjoint operator by:

$$model(mid, h, depth) += conj(W(V, h, depth, time)) \times data(mid, offset, time) \quad (9)$$

Notice how the weights are conjugated (if complex) but not inversed. That is one of the indicators that it is different to start from a modeling operator than from a migration operator (Trad, 2016). Equations (8) and (9) are simplifications because other operators are involved in real implementations, like antialiasing filtering, omega filter and interpolation. Adding these filters, the adjoint pair looks more like:

Adjoint (migration)

- resample data to fine sampling (usually 1 ms for efficiency)
- apply phase filter + double integration to input
- calculate mapping and amplitude weights from traveltime tables ($t_{shot} + t_{receiver}$)
- calculate AAF length (which produces three different traveltimes)
- add the three data samples to model with corresponding weights $(-1, 2, -1)$

Modeling

- calculate mapping and amplitude weights from traveltime tables ($t_{shot} + t_{receiver}$)
- calculate AAF length (which produces three different traveltimes)
- spread value from model to three different data samples with corresponding weights $(-1, 2, -1)$
- apply conjugate phase filter + double integration to output
- resample data to sampling rate

LSRTM

The adjoint-pairs for LSRTM are more difficult to understand because it is not easy to think of RTM as a matrix. Ji (2009) provides a quite complete discussion but only for the post-stack RTM case. Xu and Sacchi (2016) and Chen and Sacchi (2017) provides detailed discussions for the elastic case. For this report, I developed a RTM forward-adjoint pair based on work from Penglyang Yang (Yang et al., 2014).

Adjoint

We choose as starting point the RTM algorithm.

- LOOP1: inject source and forward propagate source wavefield from $time = 0$ to $time = t_{max}$
- at each time step save boundary conditions for wavefield (first and last two rows, first and last two columns for wavefield). We use this to reconstruct the source wavefield in next step
- LOOP2: inject data (or residuals) and reverse time propagate from $time = t_{max}$ to $time = 0$, to calculate the receiver wavefield
- imaging (crosscorrelation between source and receiver wavefields), for every time step and sum
- reconstruct the source wavefield from the boundary conditions (forward time step)

Modelling

- Forward propagation of the source wavefield. Since the velocity model is smooth, this wavefield is not the same as the one obtained when creating synthetic data.
- cross-correlate source wavefield with reflectivity to generate receiver wavefield

The forward model operator is calculated in such a way that it passes the adjoint test in the CG algorithm. For this reason, we can not use the same approach as in FWI, but we are restricted to Born modelling (since migration si linear, modeling has to be linear as well). Notice how that introduces a mismatch between physics and operator. This can be observed in particular when the data contains multiples as we will see in a later example.

A COMPARISON BETWEEN LS KIRCHHOFF AND LSRTM

In this section, we will look at results from LSMIG for both RTM and Kirchhoff. In both cases, I use a similar implementation with adjoint-pairs that pass the adjoint test with similar precision. Figure 2 shows a RTM for the Marmousi model using 25 shots, and Figure 3 shows the LSRTM result after 9 iterations. I used a small number of shots to illustrate the benefits from the inversion. LSRTM shows a better focusing, in particular in the shallow. For example, it is possible to see the curvature of the events in the upper left of the model that are not visible in the RTM image. Even without noise control mechanism or filtering, there is no noise introduced by the inversion.

Much of the high-resolution definition of this last result comes from the almost exact match between the wavefield simulation and the prediction operator inside LRTM, so-called inverse crime (Schuster, 2017). In reality, there are some important differences: the data are created with finite difference but RTM prediction operator is Born modeling obtained by a crosscorrelation between source wavefield and reflectors. The RTM operator

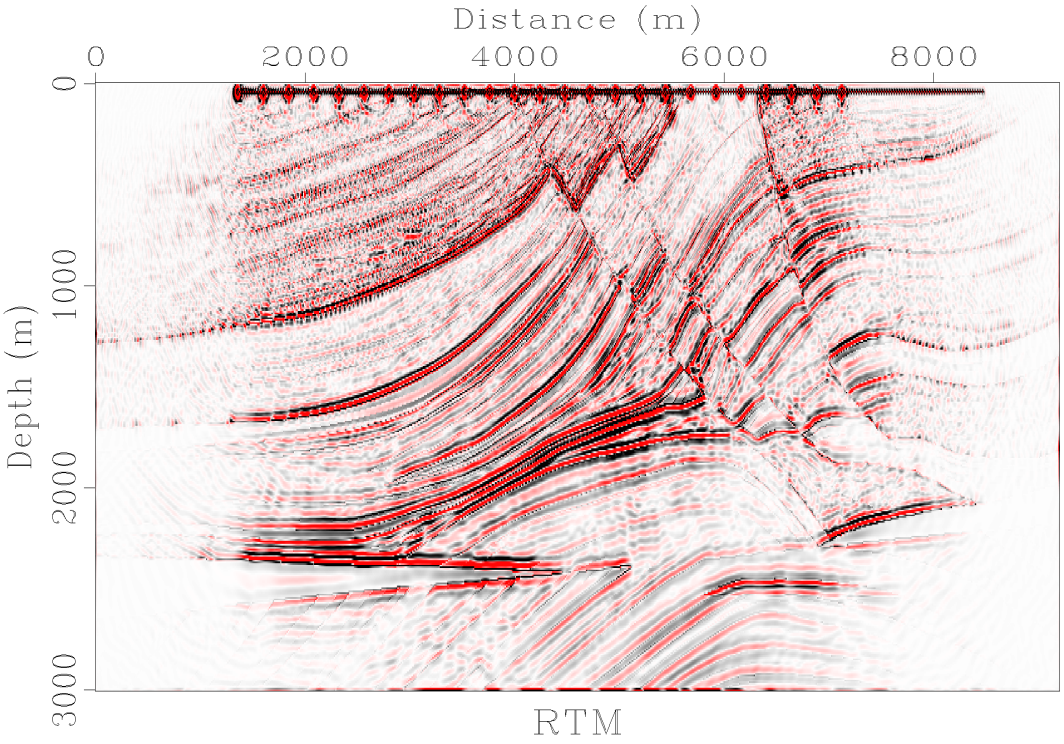

FIG. 2. RTM with accurate model and right physics.

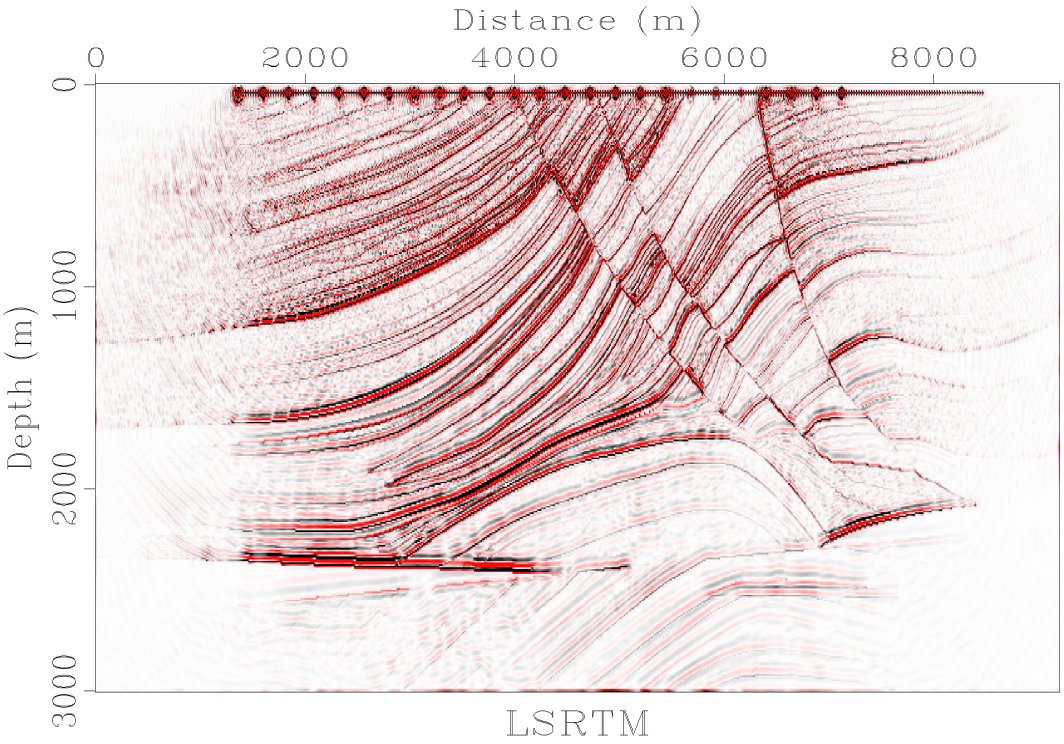

FIG. 3. LSRTM with accurate model and right physics. Best possible scenario for LSMIG.

uses a 4th order FD in space, while the data are created with an 8th order accuracy in space. In spite of these differences the match is very good because the source wavefield used in the Born modeling is quite similar to the data. This high-resolution image may even include a partial matching of internal multiples, probably with the addition of some low amplitude cross-talk produced by the cross-correlation.

If we introduce some differences between the source wavefield and the data, for example by smoothing the velocity model, we get the results in Figures 4 and 5. Although both images are clean, the iterations have not changed much the result. There is a mismatch between Born demigration and finite difference data, and the source wavefields are different as well. Internal multiples, for example, are not predicted by the RTM operator. Also, the velocity changes may have decreased the coherence of the predicted data across different angles, which would decrease the sharpness of the reflectors.

If we introduce a more serious mismatch, for example by allowing the RTM prediction to generate surface multiples, then we get noise in the RTM image (Figure 6). This noise is, however, much stronger in the LSRTM result (Figure 7), because inversion is using wrong residuals, caused by wrong physics, to update reflectivity.

For LSRTM with synthetic data, it is possible to take a look at the matching between physics and operator in detail by looking at the wavefields. Figure 8 shows a snapshot for a shot generated in a sharp velocity model. We see the wavefield reflected on interfaces. Figure 9 shows the same snapshot for the smooth velocity model. For LSRTM to fit the data, it has to adjust the reflector amplitudes to be able to generate the wavefield in Figure 8 from the wavefield in Figure 9 (actually LSMIG does not match all the wavefields but one sample per time step and receiver).

In Figure 10 we see the source wavefield just before hitting the reflectors and in Figure 11 the receiver wavefield just after. Comparing with the wavefield in Figure 8 we see reflectors need to keep changing to get a better match. In reality, these fields would look even more different when reflectors from all shots are included in the image. If the velocity is different, then these wavefields may never be able to match each other even if the perfect reflectivity were used. These figures give us a hint at why it is not easy to get LSMIG to work well in real data. For acquired data, we don't have the actual source wavefield but a mimic we create from each time sample.

We saw in Figure 3 a very high-resolution image when the velocity is accurate with sharp reflectors. To understand why in that case the method works so well, we can compare the source and receiver wavefields (Figures 12 and 13) for a sharp velocity model. Because the source wavefield already contains much of the details of the input data (almost perfect match between physics and operator), the receiver wavefield can be made to match the input data really well by changing reflectors. Still, some differences appear in the wavefields the multiplication with reflectors must produce some cross-talk between internal multiples. With some speculation, we could say that the image seems to be less affected by the extra detail caused by the crosstalk, that is affected by the lack of complexity in the source wavefield.

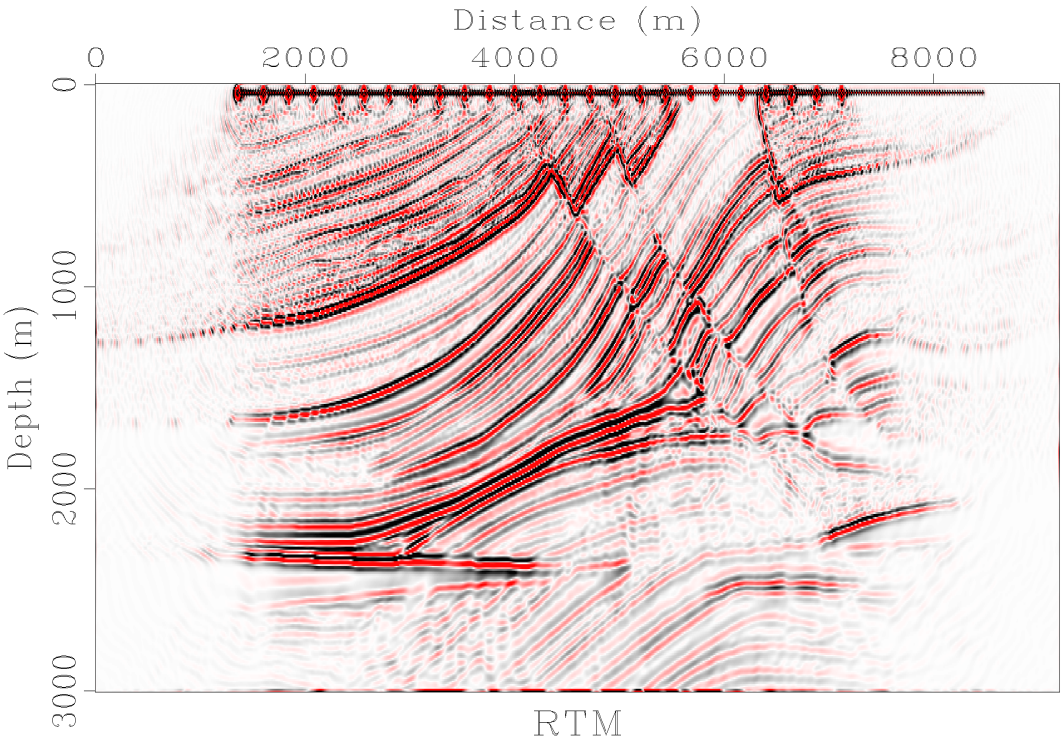

FIG. 4. RTM with smooth model.

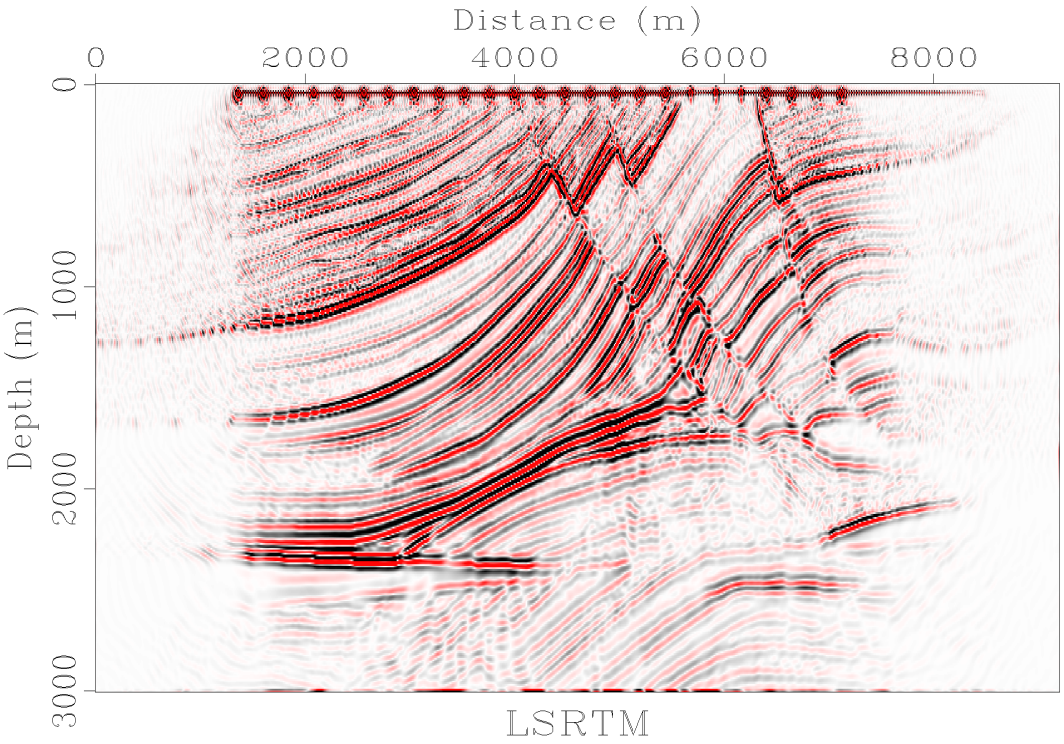

FIG. 5. LSRTM with smooth model. Iterations do not seem to improve much results even when fitting has improved.

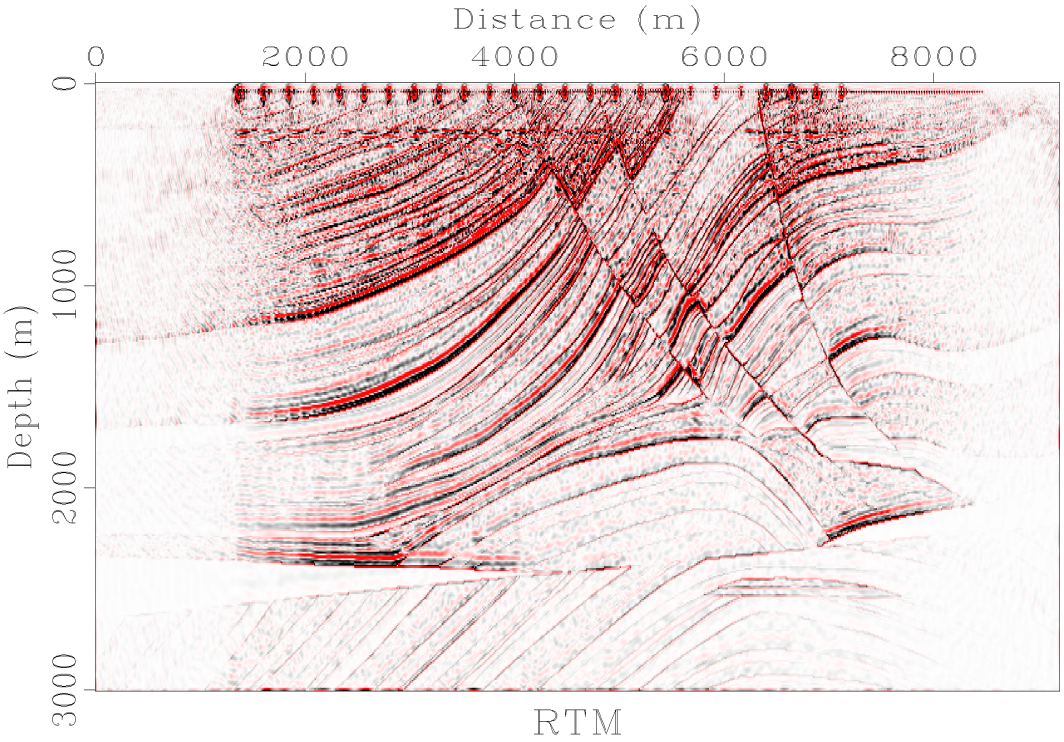

FIG. 6. RTM with wrong physics. Some noise appears as a consequence of the operator mismatch but not much.

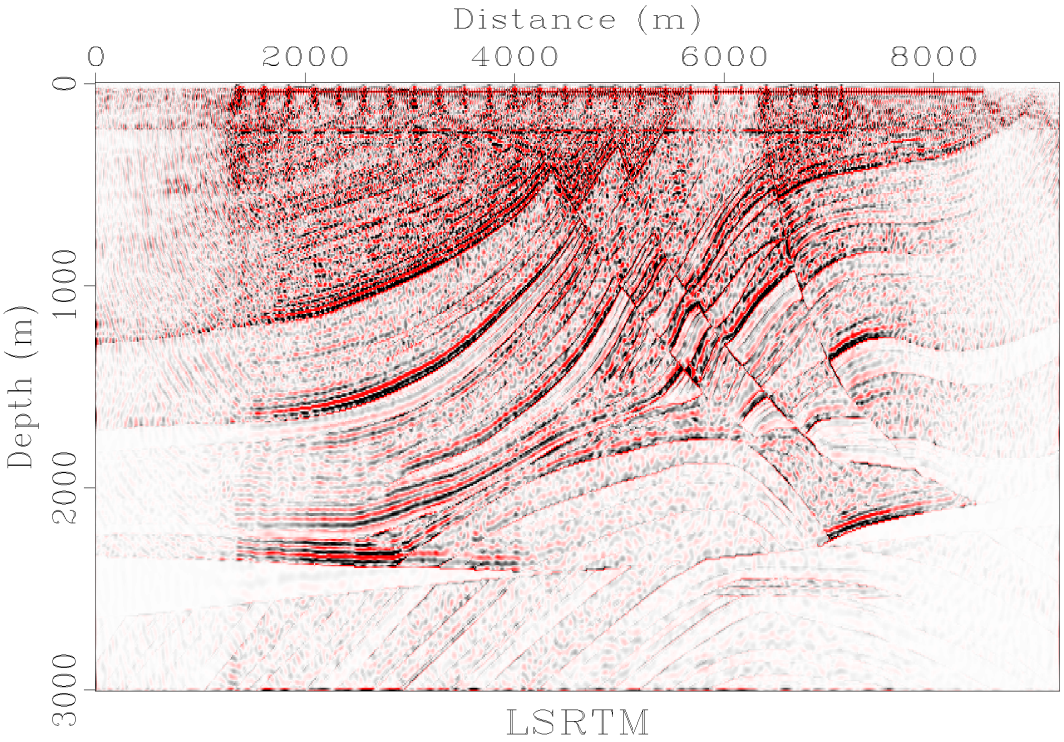

FIG. 7. LSRTM with wrong physics. Iterations fail to match the data and trying to diminish the residuals just introduces noise.

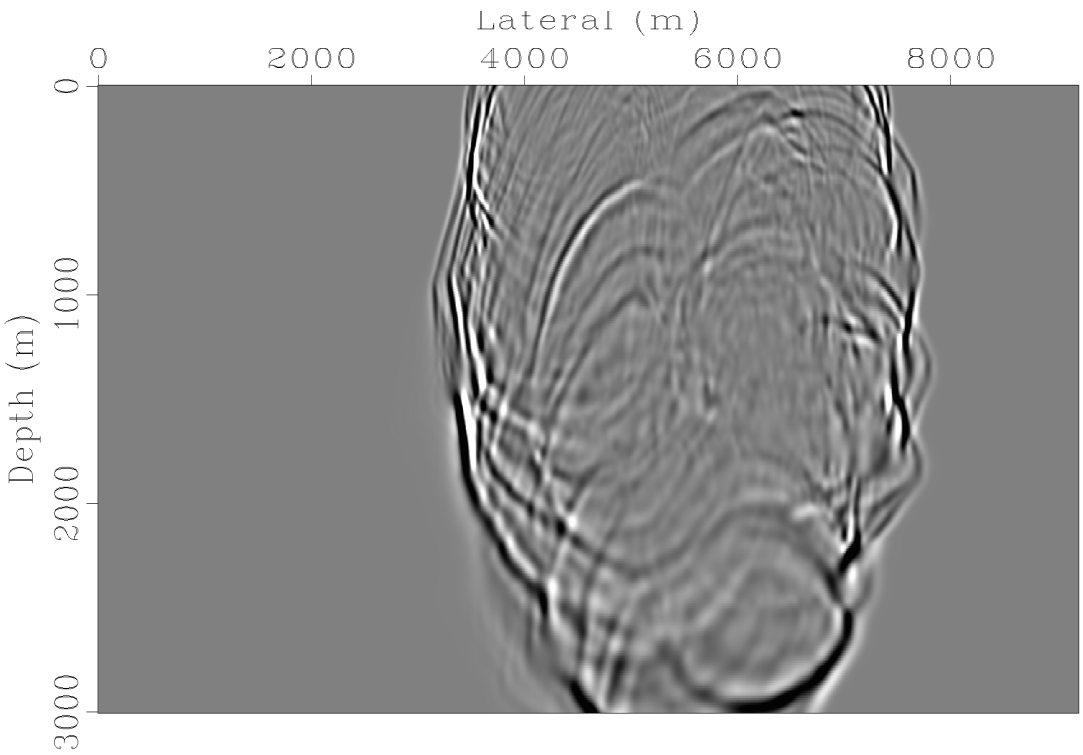

FIG. 8. Snapshot (wavefield at fixed time) for a synthetic shot (generated in a sharp model)

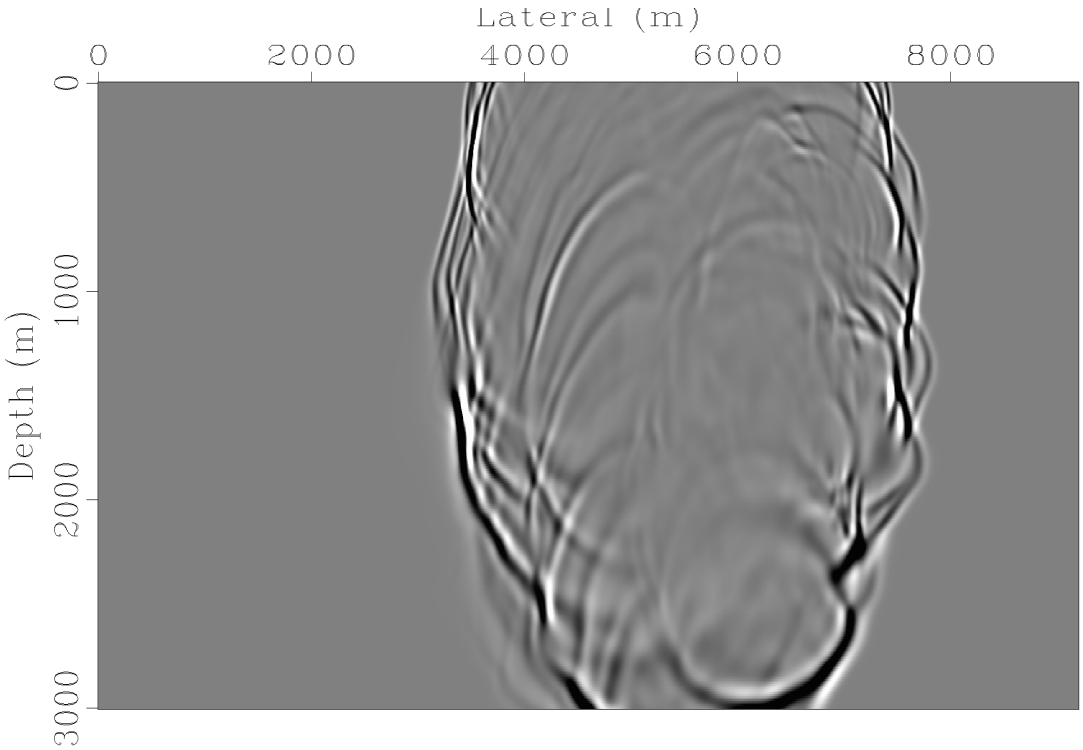

FIG. 9. Snapshot for a shot generated in smooth velocity. LSRTM will try to convert this wavefield to the one above by multiplying it with reflectors. The error of this conversion translates into corrections to reflector amplitudes.

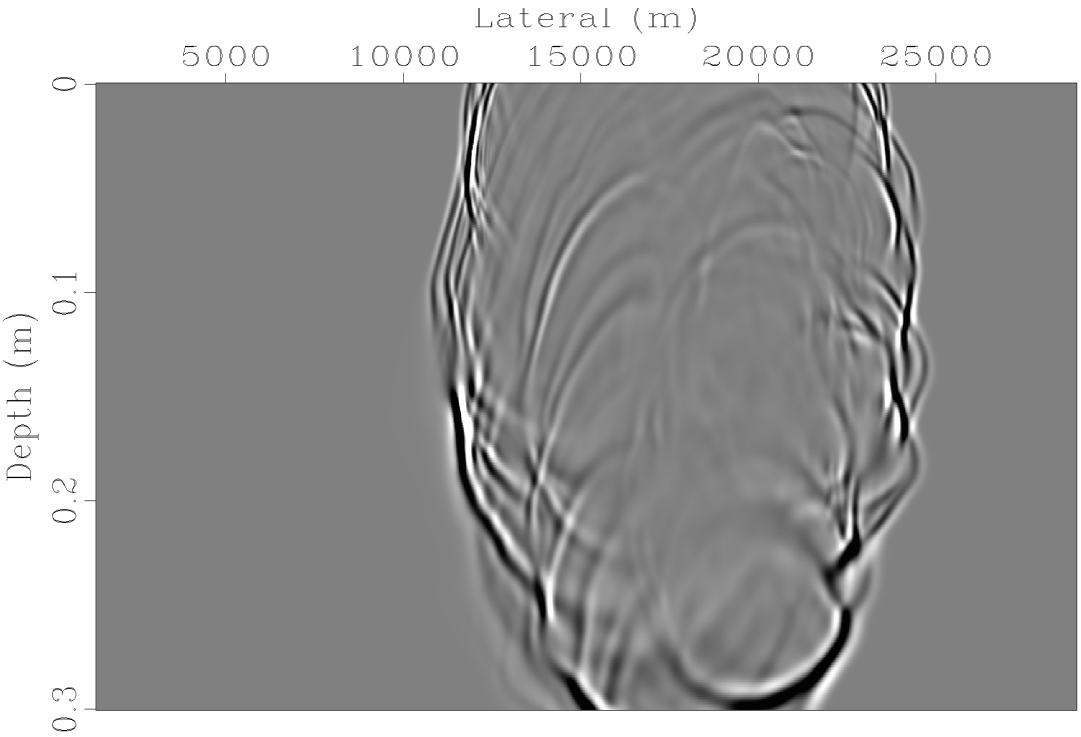

FIG. 10. Source wavefield inside forward modeling for LSRTM (smooth velocity)

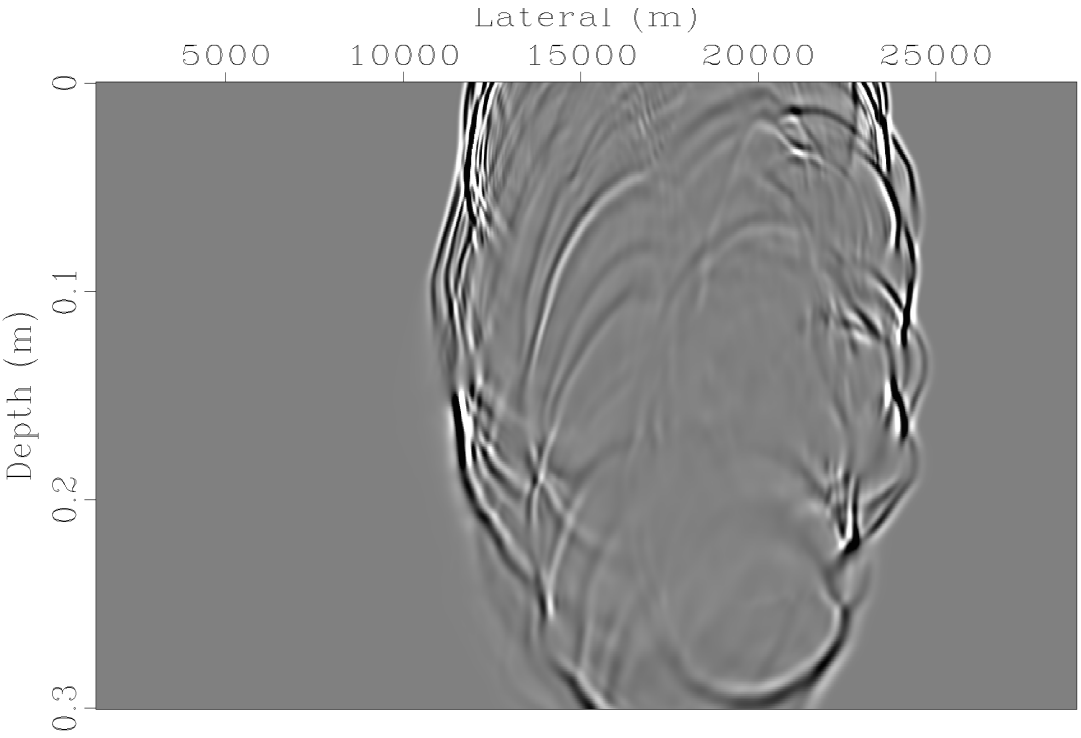

FIG. 11. Receiver wavefield generated in forward modeling operator after hitting reflectors.

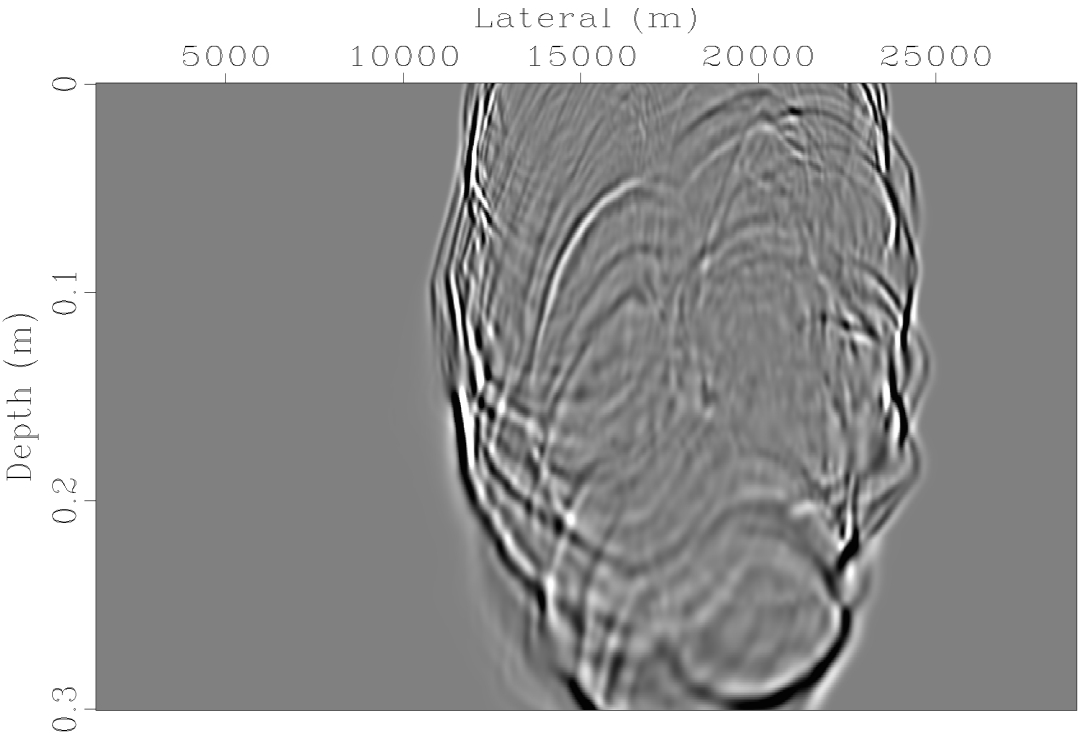

FIG. 12. Source wavefield inside forward modeling for LSRTM (exact velocity)

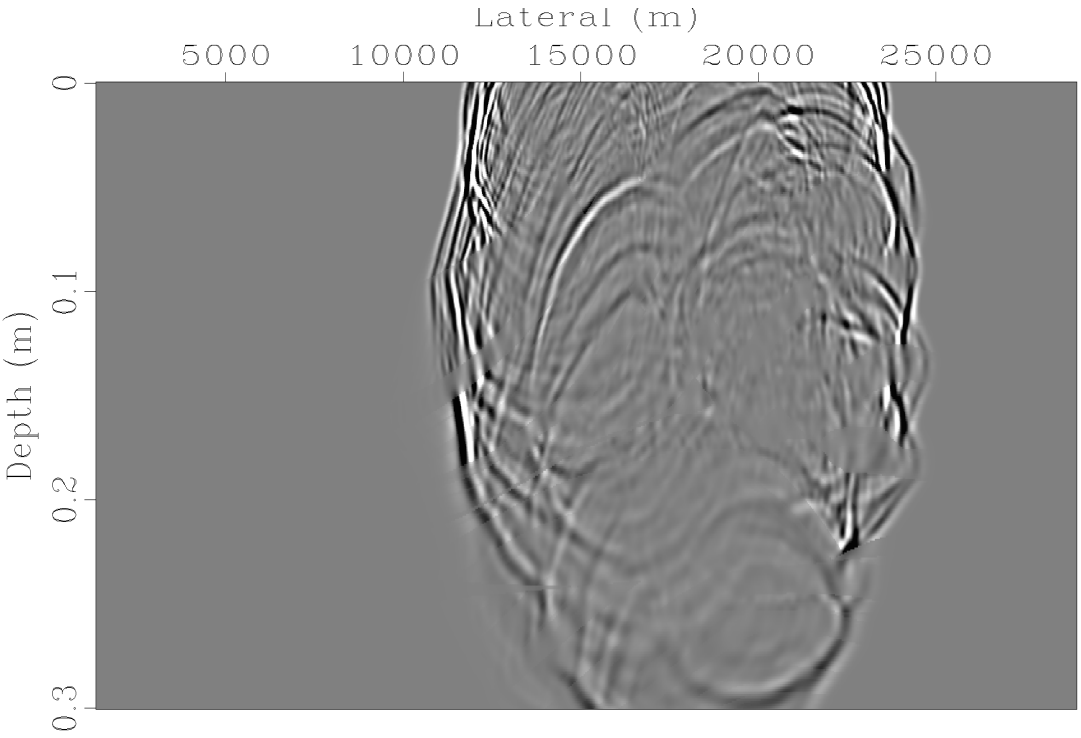

FIG. 13. Receiver wavefield generated in forward modeling operator after hitting reflectors.

Figures 14 and 15 show the equivalent of Figures 2 and 3 but this time using Kirchhoff operator. Although the shallow part has become a bit sharper after just a few iterations, we see significant noise introduced by the inversion in the deeper part. This is quite different from the encouraging result we saw in Figure 3.

We will see later a method to address this noise by residual filtering (Figure 17), but first, let us discuss why this noise is much stronger with Kirchhoff than with RTM operators. The algorithms are equivalent so we can infer that the differences in the operators are responsible for this noise. Since the Kirchhoff migration itself (Figure 14) is clean and reasonably accurate, it is somewhat surprising that the combination of a correct migration operator and a correct inversion algorithm gives incorrect results.

Different traveltime tables and LSMIG

Kirchhoff operator contains two somewhat independent components: amplitude and phases. The amplitudes are mainly due to the weights used during the operator summation. These weights can be calculated from ray tracing algorithms, but more often are simply approximated from shot and receiver times and a constant velocity approximation (Dellinger et al., 1999). If the amplitudes used in the operator are wrong, the reflectors in the image will have wrong amplitudes as well, but the effect in the inversion is not that critical (Trad, 2016). Phases, on the other hand, are obtained by tracing rays across a smooth velocity model and stored in traveltime tables after smoothing. These tables have a very strong effect on the quality of the image (coherence) and even more on the inversion.

Figure 18 shows traveltime tables (top) and rays (bottom) for the ray fan reconstruction method (Červený and Hron, 1980). The tables show many discontinuities caused by shadow zones and multipathing. Of course, in reality, smooth velocity models are used for ray tracing. This reduces the discontinuities but not completely. In fact, although the ray fan method can be very precise, other methods can produce smoother tables. Figure 19 shows the comparison between tables from the ray fan method (above) and from the wavefront reconstruction method (Vinje et al., 1993). When using the smooth tables, LSMIG does not accumulate noise as much (Figures 20 and 21). However, the focusing suffers as a consequence of errors in phase introduced when smoothing the wavefronts. It seems that accurate tables with abrupt discontinuities produce accurate images but noisy inversions, whereas inaccurate smooth tables reduce the capabilities of the inversion to improve focusing. These results are very similar to what we observed for LSRTM, except that in the case of Kirchhoff the effects are far more serious. Furthermore, none of the results for LSKirchhoff where able to match the improvements we saw in Figure 3. This is expected since the Marmousi model is not very friendly to ray tracing algorithms. LSKirchhoff represents a difficult challenge in complex structures because in these scenarios it is very difficult to achieve a good match between wave propagation and ray tracing.

A CLOSE LOOK AT RESIDUALS AND THEIR COMPONENTS

In previous examples with LSRTM and LSKirchhoff migration, we have seen that mismatches between physics and operators result in accumulation of noise during LSMIG. To understand why, let us look at the expression of the cost function that least squares

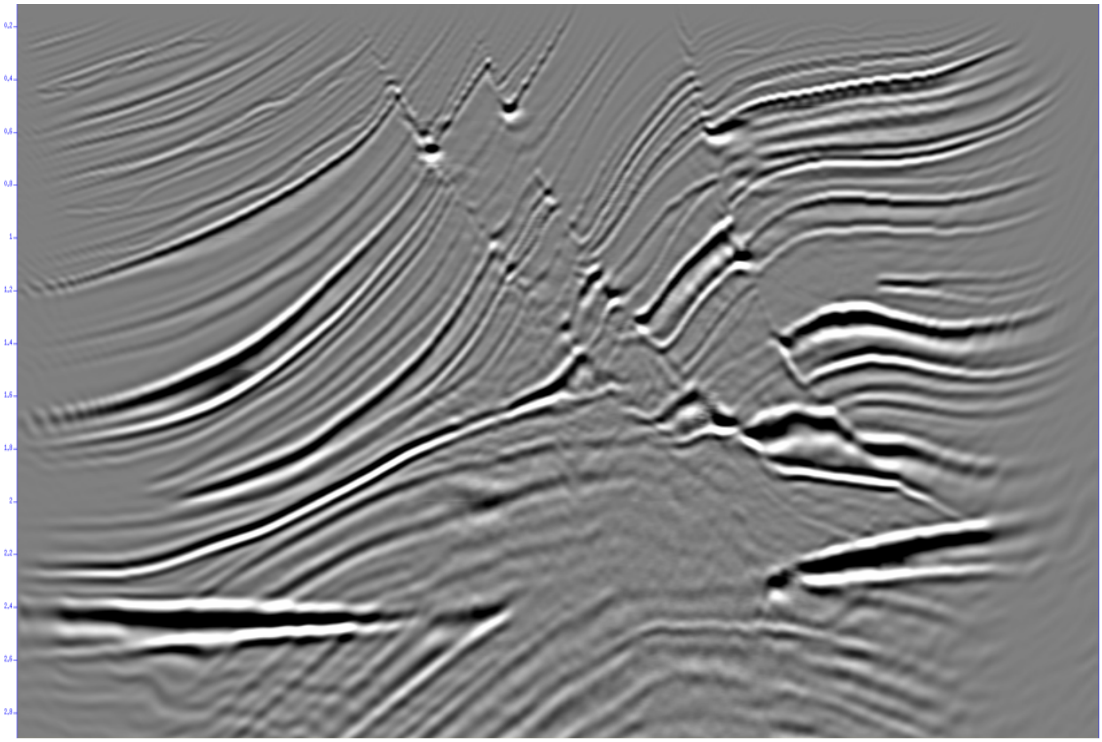

FIG. 14. Kirchhoff migration from Marmousi model

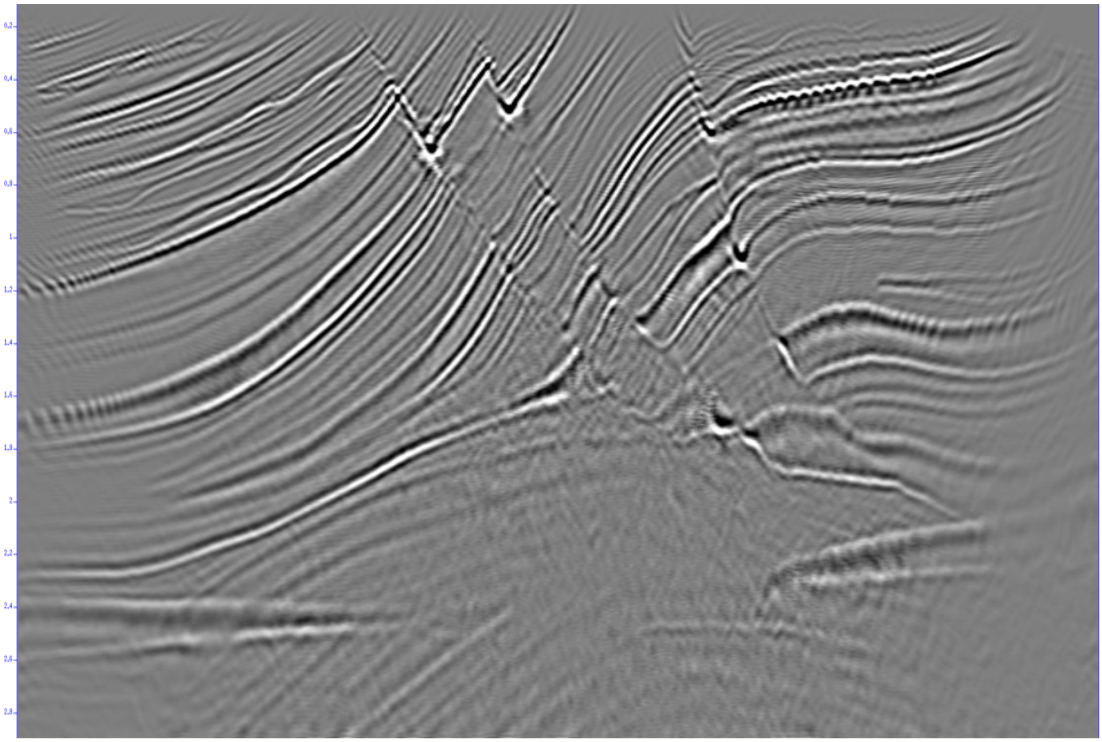

FIG. 15. LSK migration for Marmousi model after 9 iterations

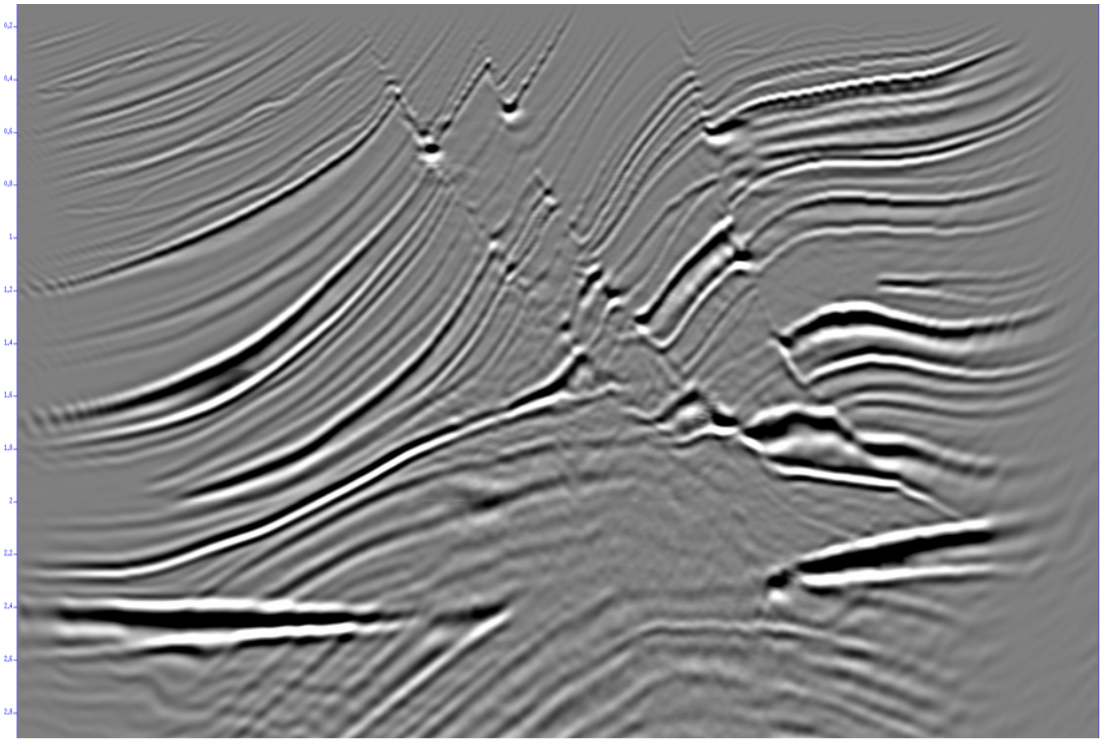

FIG. 16. Migration from Marmousi model (repeat)

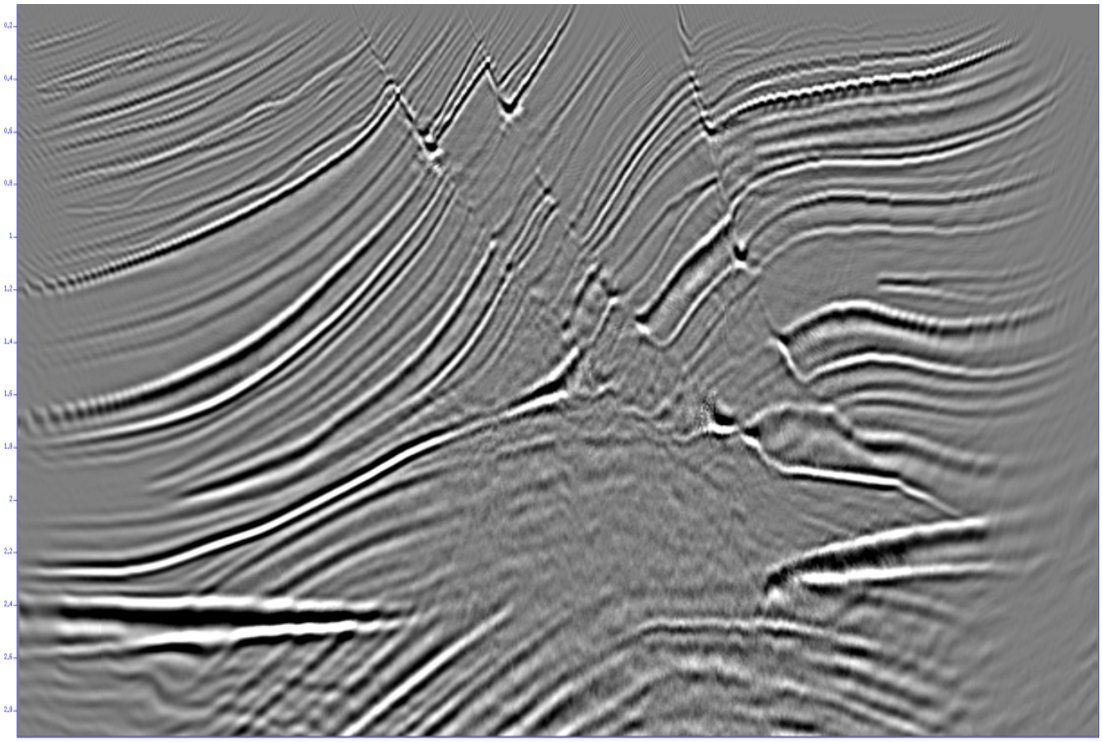

FIG. 17. LSK migration for Marmousi model after 9 iterations, with noise control by residual filtering

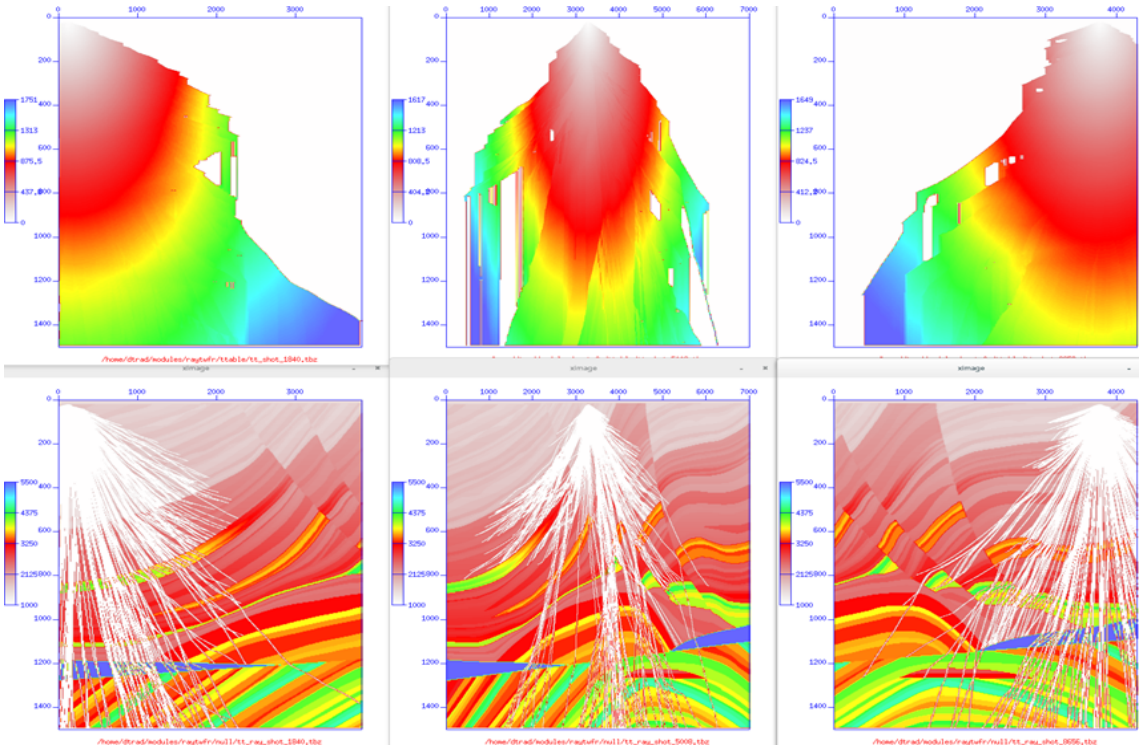

FIG. 18. Traveltime tables and corresponding rays for ray fan method for sharp velocity model

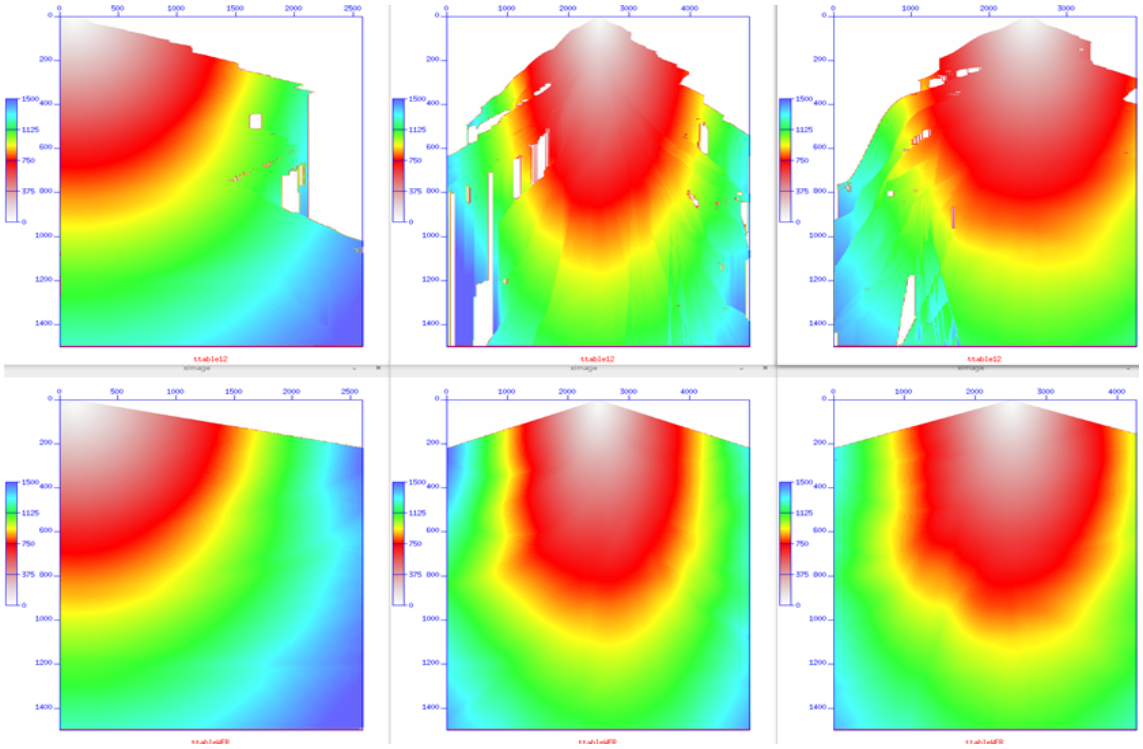

FIG. 19. Traveltime tables for ray fan method (top) and wavefront reconstruction method (bottom) for smooth velocity model

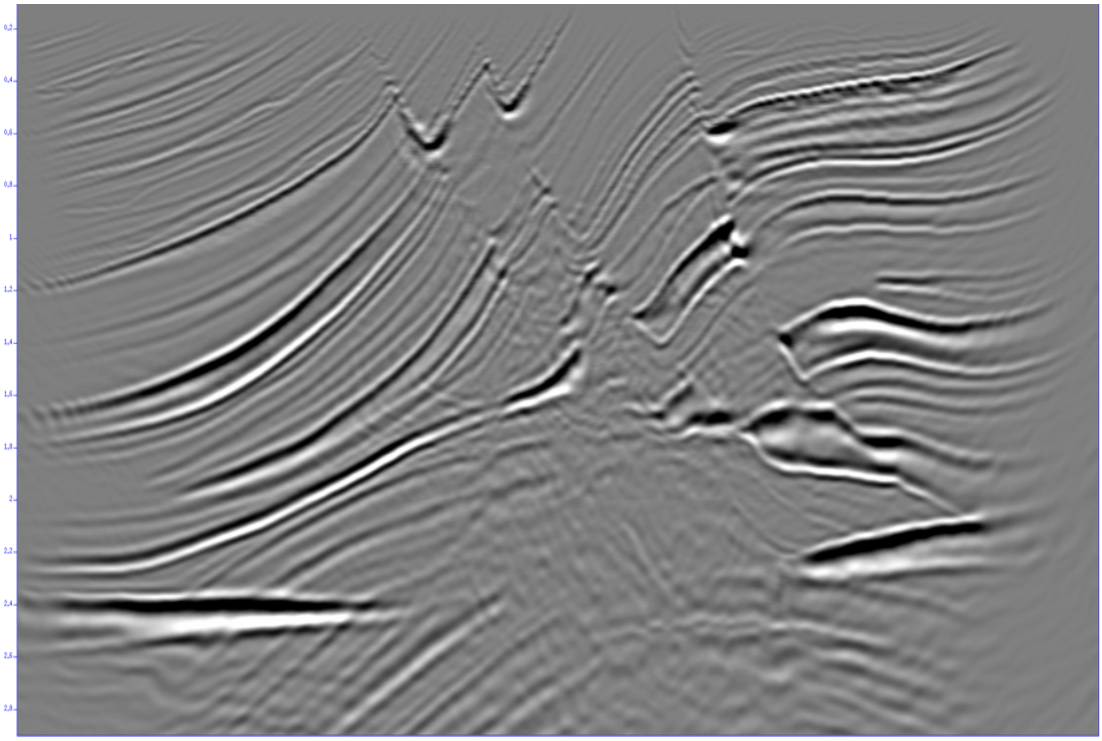

FIG. 20. Migration from Marmousi model with wavefront reconstruction

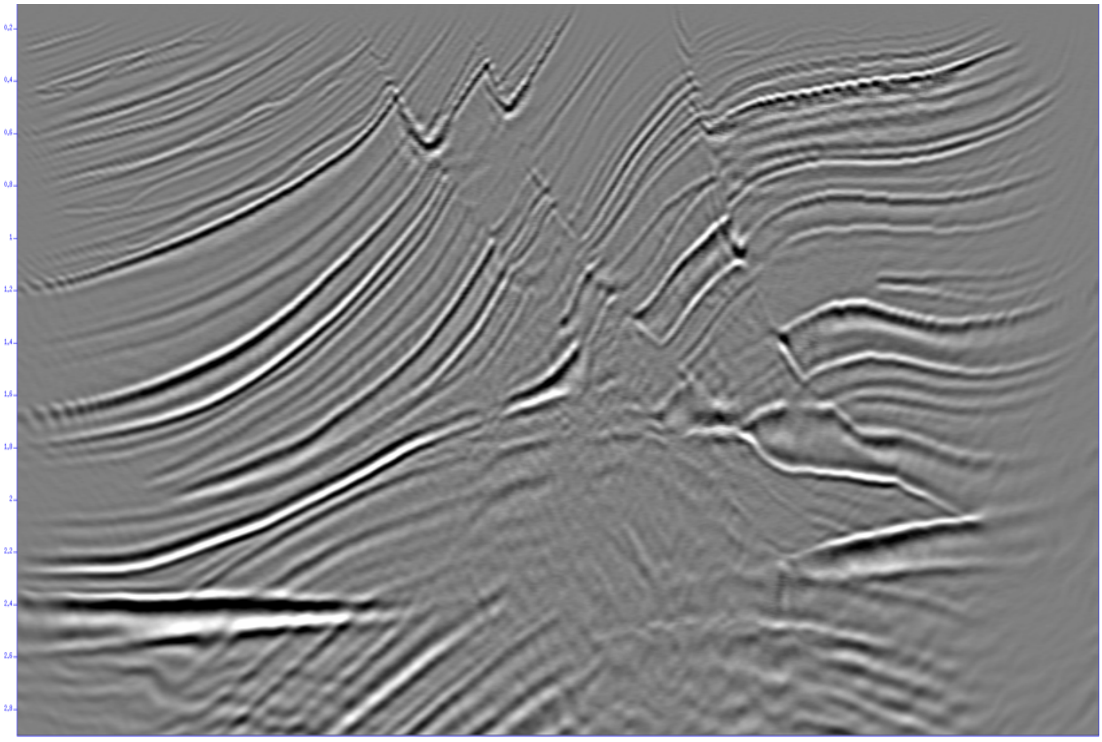

FIG. 21. LSK migration with wavefront tables

migration is minimizing:

$$J = \|\mathbf{d} - \mathbf{Lm}\|^2 = \|\mathbf{R}\|^2 \tag{10}$$

where $\mathbf{R} = \mathbf{d} - \mathbf{Lm}$ are the residuals or error in predicted data from our current model $\mathbf{m}$. For LSMIG the model is the reflectivity, so the predictions are the result of demigrating reflectors (Kirchhoff modeling) or scattering from velocity perturbations (Born modeling). Similarly to FWI, we obtain model corrections by mapping (migrating) residuals from data space to model space with the operator $\mathbf{L^T}$, which is the adjoint to the modeling operator $\mathbf{L}$:

$$\mathbf{\Delta m} = \mathbf{L^T R}. \tag{11}$$

In this procedure there is an implicit assumption that all the prediction errors can be corrected by a suitable correction in the model. However, in general, this is a wrong assumption, because residuals contain several components: one that is the result of errors in the model

$$\mathbf{R_1} = \mathbf{d} - \mathbf{L\Delta m}, \tag{12}$$

another that is the missing physics, or events in the data that the operator cannot predict

$$\mathbf{R_2} = \tilde{\mathbf{d}}, \tag{13}$$

and finally another component that results from errors in the operator

$$\mathbf{R_3} = -\mathbf{\Delta L m}. \tag{14}$$

The first component $\mathbf{R_1}$ is the only one that is useful to improve the model by diminishing the cost function. The second component often falls outside of the mapping from data to model as well (see below), in which case it will not have consequences (for example, frequencies that we are not predicting). But also it may contain events that map to the wrong model components (for example aliasing, direct wave, multiples), and will appear as noise. The third component is always a problem, because it maps back to the model space in the form of wrong model components.

In general, some of the components $\mathbf{R_2}$ and all $\mathbf{R_3}$ will produce noise in the model space. To address this issue is common practice to remove from the data before inversion anything that can lead to $\mathbf{R_2}$ (attenuate multiples, direct waves and salt reflections). However, although these are examples of events we know the operator will not predict correctly, we don't know a priori everything that the operator cannot predict. In this work, I use an adaptive filtering of the residuals to remove those events during inversion because that is when they can be detected. For the events in $\mathbf{R_3}$ the situation is more difficult because they always produce noise, and are difficult to eliminate because they are essentially remaining errors in our procedures and a priori estimations (for example velocities). Notice that in non-linear inversion techniques like FWI, we have a mechanism to account for at least part of the operator errors (we change velocity and therefore the operator with iterations). In linearized techniques like LSMIG, we assume there is no error in the operator. Sometimes an extended operator can be defined to capture the wrong predictions into a separate model space to prevent them from affecting the primary model space.

As an aside note, the $\mathbf{R_2}$ component is not eliminated by applying numerical regularization to the inverse problem. Numerical regularization is a solution to the nullspace problem, which is a different issue. The nullspace is defined as a component of the model space that has no mapping to the data space:

$$\mathbf{0 = L m_n} \tag{15}$$

The nullspace can grow to large numerical values because it has no mapping to the data space and therefore minimizing the cost function has no control over its size. This is the familiar situation of lost information during forward modeling, for example, the D.C component of the data after convolution with a band limited wavelet. Numerical regularization can control the nullspace because it demands to minimize the model space as well. This effectively eliminates anything that is not required by the data. When using CG, this problem tends to disappear by itself because the adjoint operator cannot produce model components in the null space Nichols (1997).

Following this argument we could think of the component $\mathbf{R_2}$ as the data that fulfills:

$$\mathbf{0 = L^T \tilde{d}} \tag{16}$$

Because the model in CG is built by sequences of adjoint pair operators, it is reasonable to believe that this component has not effect on the inversion when performing conjugate gradient. Experience has proven however that data components are not predicted properly by the operator will still influence the inversion, but it is unclear to me why. A possible explanation is that it can bias the step size calculation. Figure 22 shows an illustration of how data and model spaces can map between each other.

Data and model space mappings in terms of SVD

It is useful to think of these mappings in a mathematical manner by looking at the Singular Value Decomposition (SVD) of the operator. Given the operator $\mathbf{L}$, its Singular Value Decomposition (SVD) is

$$\mathbf{L = USV}^T, \tag{17}$$

where $\mathbf{U}$ contains the singular vectors for the data space, $\mathbf{V}$ contains the singular vectors for the model space and $\mathbf{S}$ is a diagonal matrix containing the singular values of the matrix. A regularized solution for $\mathbf{m}$ that minimizes the cost

$$J = \lambda \|\mathbf{m}\|_2^2 + \|\mathbf{d - Lm}\|_2^2 \tag{18}$$

can be obtained by the singular vector expansion of $\mathbf{m}$ (Hanke and Hansen, 1993) as

$$\mathbf{m} = \sum_{\mathbf{i=1}}^{\mathbf{p}} \varphi(\sigma_\mathbf{i}^\mathbf{2}) \frac{\mathbf{u_i^T d}}{\sigma_\mathbf{i}} \mathbf{V_i} + \sum_{\mathbf{i=p+1}}^{\mathbf{n}} \mathbf{u_i^T d V_i}, \tag{19}$$

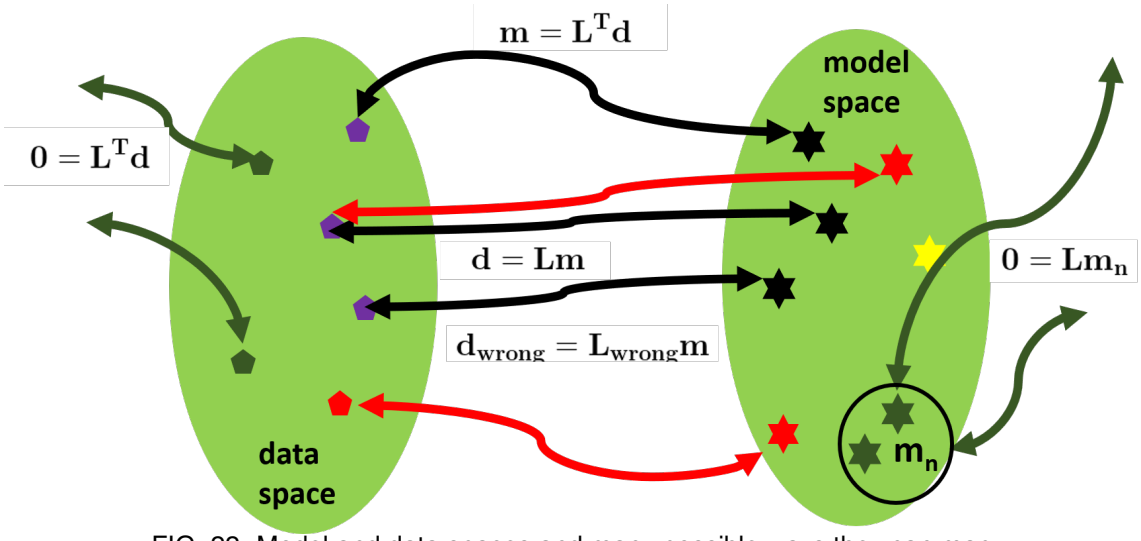

FIG. 22. Model and data spaces and many possible ways they can map

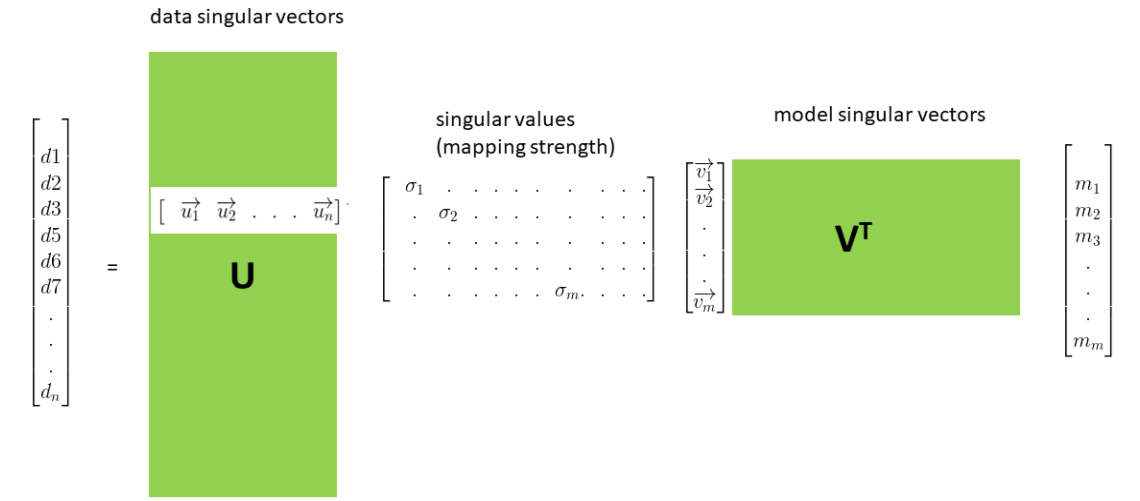

FIG. 23. Model and data spaces mapping through singular values and singular vectors

where n is the size of $\mathbf{d}$, p is the rank of $\mathbf{A}$, σ_i are the singular values of $\mathbf{A}$, the $\varphi(\sigma_i)$ are scalars between 0 and 1 that limit the contribution of the singular vectors in the rank of $\mathbf{A}$ to the solution. The second term in the summation gives $\mathbf{m}^*$, the nullspace of the kernel $\mathbf{A}$, whose dimension is $n - p$. The filter functions $\varphi(\sigma_i)$ can be computed as

$$\varphi(\sigma_i) = \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \ . \tag{20}$$

A simpler approach is the truncated SVD that uses only those singular vectors associated with the singular values above some numerical threshold. In this case $\varphi(\sigma_i) = 0$ or 1, and the expansion is limited to

$$\mathbf{m} = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{d}}{\sigma_i} \mathbf{v}_i, \tag{21}$$

where for example k can be chosen as *rank*$(\mathbf{A})$. Even though these two solutions are different, both minimize the ℓ_2 model norm for the model space subject to the data constraints. The difference is that in one case we minimize the model norm in the model space spanned by all singular vectors, and in the other case, the minimum model norm solution is in the model subspace span for only the singular vectors we use.

Noise control in data space and model space

A common way to control the noise in inverse problems is applying numerical regularization in the form of a penalization of model components that have no mapping to the data space (the model nullspace). This is normally achieved by a model weight function, $\mathbf{W_m}$, which can have many different expressions (Trad, 2016). For example, it can enforce a minimum size model in terms of a chosen norm, typically either ℓ_1 or ℓ_2 to enforce sparseness or smoothness respectively. In LSMIG is more common to have filters instead of weights, to eliminate abrupt variations in some direction of the model space that is, by design, supposed to be smooth (for example the offset or angle direction in common image gathers). As mentioned in the previous section, this type of regularization does not completely eliminate the problem of wrong physics. It helps because it enforces some type of behavior in the model space that is most likely not to be fulfilled by the wrong residuals obtained by wrong operators. However, these components on the residuals will still force the updates in each model in the wrong direction. Another issue with this type of filters is that they tell the problem what kind of solution we want. This is acceptable sometimes, but it may become a pitfall since it will remove from the solutions unexpected but important results.

It is also common in inversion to apply data weights or data space filters. This is known as robust inversion. Data weight functions proportional to the inverse of residuals, for example, transform the ℓ_2 norm in the cost function into the ℓ_1 norm, decreasing the effect of large residual components, also known as outliers, that do not fit the model. In the case of wrong physics or wrong operators, we can not assume that the residual components we want to isolate or down-weight are large. For example, a data component that is not properly predicted by the operator will appear as any other residual component. The problem is how to identify these components in the data space.

Although the total residual norm (that is a real number), decreases monotonically as the model fits the data better, individuals parts of residuals will grow or keep constant if they are not mapped correctly to the model space. In fact, the step size in inversion algorithms is one number that represents the gradient scaling the optimally decreases the global residual norm (as opposed to local residuals). Also, these poorly mapped parts of the data space may actually decrease if certain combinations of the model space can predict the data. This is exactly what happens in signal processing with aliasing, where low-frequency model components can explain high-frequency data components. In the case of LSMIG, this "impersonation" can take more complex shapes. Unphysical combinations or arrangements of the model space can work together to decrease components of the data space that can not naturally be predicted by an incomplete operator. Suitable model weight functions can be used to downplay or remove these unphysical arrangements, but with the drawback mentioned above: these components keep appearing in the residuals and therefore model updates, and the model is enforced by our prior knowledge.

If on the other hand, we were able to detect these residual components and filter them out during iterations, the inversion process will converge faster. In fact, it is common to remove events we cannot predict properly before inversion. The problem is that there are many parts of the data we do not know we can not predict, but we can try to use the individual residual evolution to detect these problem events. As a proof of concept, I use a very simple approach of turning off components of the residual space that consistently increase with iterations or do not decrease at all. In Figure 24 we can see an input shot, a predicted shot, the difference (residuals) and the image after 9 iterations of LSMIG with Kirchhoff operator for Marmousi data. The residuals show many events that are poorly predicted. Migration of these (broken) events, in each CG iteration, will produce broken reflectors that will be used to update the current image. These broken reflectors appear as noise in the image. As the iterations proceed the energy accumulates and the final image shows degrading. Figure 25 shows the same experiment but this time with a filter in the data space that turns off residuals when they increase with iterations. The residual plot now is much sparser because their components that have no proper mapping to the model space were zeroed during previous iterations. Continuing iterations will still decrease components we see on this filtered residual, but as we can infer from the image they are not very significant for the solution. As a consequence, we see the image is somewhat cleaner than the corresponding image in Figure 24.

Although this is not necessarily a solution to the problem we are considering, we can obtain useful information from this test. We can find which are the parts of the image that can be still improved by iterating longer. We can infer what parts of the operator have wrong physics or poor approximations. Finally, we could improve noise attenuation in the data space. In reality, neither the data or model weight functions should work alone. Both components have a role to play. In this test, for example, I simultaneously use the model weight component to emphasize updates in the deeper part of the model.

CONCLUSIONS

Although the mismatch between physics and operator is much stronger for Kirchhoff migration than for RTM, in particular working with synthetic data generated by finite dif-

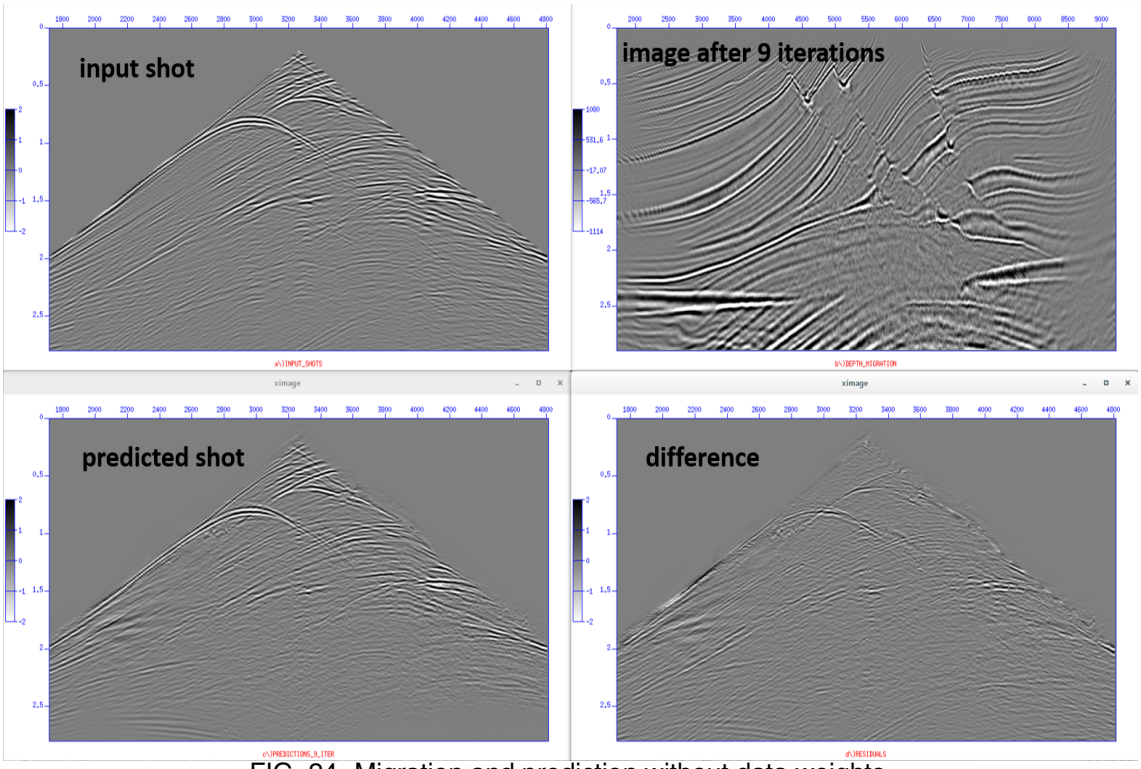

FIG. 24. Migration and prediction without data weights.

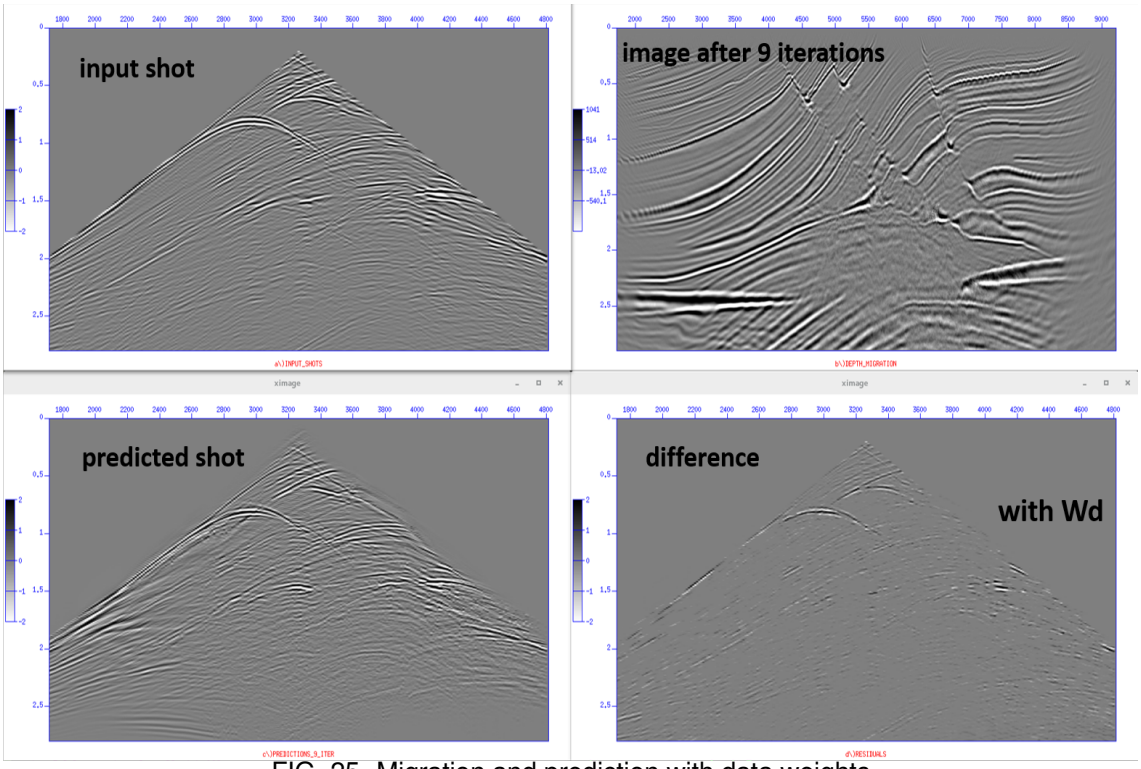

FIG. 25. Migration and prediction with data weights

ferences, we can also see some similar effects for LSRTM. I discussed above in the case of real data there are numerous aspects of the acquired data that will not be explained properly by finite difference modeling, like elasticity, attenuation, and anisotropy. We can, however, illustrate these issues with simple finite difference modeling for the acoustic-isotropic case. I will illustrate how this mismatch occurs for prediction of surface and internal multiples. Surface multiples can be turned on and off by simple removing the absorbing boundary condition (ABC) at the top of the model. The internal multiples can be turned on and off by changing the degree of smoothness in the input velocity model to migration.

ACKNOWLEDGMENTS

My gratitude to Sam Gray for many insightful discussions during his visits to CREWES. I also thank CREWES sponsors for contributing to this seismic research. I also gratefully acknowledge support from NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 461179-13.

REFERENCES

Červený, V., and Hron, F., 1980, The ray series method and dynamic ray tracing system for three-dimensional inhomogeneous media: Bulletin of the Seismological Society of America, **70**, No. 1, 47–77.

Chen, K., and Sacchi, M. D., 2017, Elastic least-squares reverse time migration via linearized elastic full-waveform inversion with pseudo-hessian preconditioning: GEOPHYSICS, **82**, No. 5, S341–S358.

Claerbout, J., 1992, Earth sounding analysis, Processing versus inversion: Blackwell Scientific Publications, Inc.

Dellinger, J., Murphy, G., Etgen, J., Fei, T., and Gray, S., 1999, Efficient 2.5 d true amplitude migration: The Leading Edge, **18**, No. 8, 946–949.

Hanke, M., and Hansen, P. C., 1993, Regularization methods for large-scale problems: Surv. Math. Ind, **3**, No. 4, 253–315.

Ji, J., 2009, An exact adjoint operation pair in time extrapolation and its application in least-squares reverse-time migration: GEOPHYSICS, **74**, No. 5, H27–H33.

Nemeth, T., Wu, C., and Schuster, G. T., 1999, Least squares migration of incomplete reflection data: GEOPHYSICS, **64**, No. 1, 208–221.

Nichols, D., 1997, A simple example of a null space and how to modify it: Stanford Exploration Project Report, **82**, 185–192.

Schuster, G., 2017, Seismic Inversion: Society of Exploration Geophysicists.

Trad, D., 2016, Nuts and bolts in least squares kirchhoff migration: CREWES Research Report, **28**.

Vinje, V., Iversen, E., and Gjoystdal, H., 1993, Traveltime and amplitude estimation using wavefront construction: GEOPHYSICS, **58**, No. 8, 1157–1166.

Xu, L., and Sacchi, M., 2016, Least squares reverse time migration with model space preconditioning and exact forward/adjoint pairs, *in* EAGE Annual Conference Proceedings, (EAGE, Vienna).

Yang, P., Gao, J., and Wang, B., 2014, Rtm using effective boundary saving: A staggered grid gpu implementation: Computers and Geosciences, **68**, No. Supplement C, 64 – 72.