

Preconditioning a non-linear problem and its application to bidirectional deconvolution

Yi Shen, Qiang Fu and Jon Claerbout

ABSTRACT

Non-linear optimization problems suffer from local minima. When we use gradient-based iterative solvers on these problems, we often find the final solution to be highly dependent on the initial guess. Here we introduce preconditioning and show how it helps resolve these issues in our current problem—bidirectional deconvolution. The results in three data examples show that results with preconditioning are more spiky than results without preconditioning. Additionally, field data results with preconditioning have fewer precursors, cleaner salt bodies, more symmetric wavelets, and faster convergence than those without preconditioning. In addition to the field data, we illustrate the theory and application of two methods of preconditioning: prediction-error filter (PEF) preconditioning and gapped anti-causal leaky integration followed by PEF (GALI-PEF) preconditioning. Unlike PEF preconditioning, GALI-PEF preconditioning helps produce the spike in the central wavelet or other position of the wavelet if we change the length of gap.

INTRODUCTION

Least-squares data fitting leads to multivariate linear equations and consequently more theories and techniques than any one person can master in a lifetime. In that field, we are always on well traveled paths. Problems with non-linear physics are another story: “My program worked great until I increased the model size a little bit.”

Nonlinear optimization problems have many unexpected traps—local minima, as shown in Figure 1. Problems with nonlinear physics require a deeper understanding of the setting than do linear ones. Luckily there exist helpful techniques that are universally applicable. The first one is to realize that linear equations can be solved with any starting guess, whereas with nonlinear relationships, a sensible starting solution is essential.

Preconditioning is a well established technique using linear regressions with prior information to hasten convergence. Preconditioning usually begins with regularization and then steers the iterative descent along the path set out by a prior model. However, it does not determine the final result.

The word “gradient” sounds like something fixed in the geometry of the application. Nothing could be further from the truth. Every application offers us a choice of coordinate systems and ways to parameterize the model, and changing the model representation changes the gradient. For example, we could be seeking the earth density as a function of location. We could establish the problem as just that, density as a function of location. On the other hand, we could establish the problem as finding the spatial derivative of the density. The two formulations really seek the same thing, but operators, unknowns, and gradients differ.

Each component of a gradient is independent of the other components and may be scaled arbitrarily as long as its polarity is unchanged. That means that any gradient can be multiplied by any diagonal matrix containing all positive numbers. Additionally, it is shown in the theory section below that a gradient may be multiplied by any positive definite matrix. That matrix happens to be the model covariance $\mathbf{B}^T \mathbf{B}$, which in local terminology is the inverse of the model styling goal times its adjoint. We may choose any positive definite matrix to modify the gradient. We may even change that matrix from one iteration to the next. What is important is that the matrix is positive definite. At early stages of descent, it is helpful to make the gradient large where confidence is high, and small where it is not. With linear regressions this has no effect on the solution. With nonlinear physics, it steers the solution away from unwelcome local minima.

In image estimation there generally are locations in physical space and in Fourier space where we have little interest, places where we have little expectation that our data contains useful information or that the model will be findable. We need (in nonlinear cases) to be certain such regions do not disturb our descent, especially in early iterations. Therefore, we should view our gradient both in the model space and in data space, then choose an appropriate diagonal weighting and filter. Given a filter \mathbf{F} and weight \mathbf{W} , we apply either \mathbf{FW} or \mathbf{WF} to the gradient. We then apply the matrix transpose, yielding either $(\mathbf{FW})^T(\mathbf{FW})$ or $(\mathbf{WF})^T(\mathbf{WF})$. This procedure destroys no information in the data, but merely selects what aspects of the data are used first. As the final solution is approached, the gradient diminishes; and the down-weighted regions eventually emerge in the gradient, because they are the only things left. Closer to the ultimate solution, it is far less dangerous having down-weighted regions affecting the solution.

THEORY

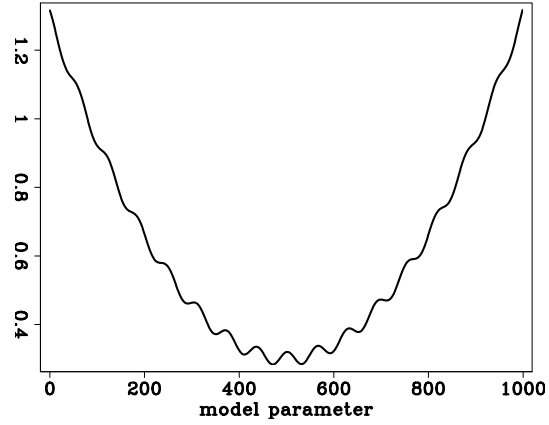
Preconditioning Offers Smart Directions

We start from fitting goals

$$\begin{aligned} \mathbf{0} &\approx \mathbf{Fm} - \mathbf{d} \\ \mathbf{0} &\approx \mathbf{Am} \end{aligned} \tag{1}$$

and change variables from \mathbf{m} to \mathbf{p} using $\mathbf{m} = \mathbf{A}^{-1}\mathbf{p}$:

Figure 1: Penalty function. [ER]



$$\begin{aligned} \mathbf{0} &\approx \mathbf{r}_d = \mathbf{F}\mathbf{m} - \mathbf{d} = \mathbf{F}\mathbf{A}^{-1}\mathbf{p} - \mathbf{d} \\ \mathbf{0} &\approx \mathbf{r}_m = \mathbf{A}\mathbf{m} = \mathbf{I}\mathbf{p} \end{aligned} \quad (2)$$

Without preconditioning, we have the search direction

$$\Delta\mathbf{m}_{\text{bad}} = \begin{bmatrix} \mathbf{F}^T & \mathbf{A}^T \end{bmatrix} \begin{bmatrix} \mathbf{r}_d \\ \mathbf{r}_m \end{bmatrix}, \quad (3)$$

and with preconditioning, we have the search direction

$$\Delta\mathbf{p}_{\text{good}} = \begin{bmatrix} (\mathbf{F}\mathbf{A}^{-1})^T & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{r}_d \\ \mathbf{r}_m \end{bmatrix}. \quad (4)$$

The essential feature of preconditioning is not that we perform the iterative optimization in terms of the variable \mathbf{p} , but that we use a search direction that is a gradient with respect to \mathbf{p}^T not \mathbf{m}^T . Using $\mathbf{A}\mathbf{m} = \mathbf{p}$ we have $\mathbf{A}\Delta\mathbf{m} = \Delta\mathbf{p}$. This enables us to define a good search direction in model space:

$$\Delta\mathbf{m}_{\text{good}} = \mathbf{A}^{-1}\Delta\mathbf{p}_{\text{good}} = \mathbf{A}^{-1}(\mathbf{A}^{-1})^T\mathbf{F}^T\mathbf{r}_d + \mathbf{A}^{-1}\mathbf{r}_m. \quad (5)$$

We define the gradient by $\mathbf{g} = \mathbf{F}^T\mathbf{r}_d$ and notice that $\mathbf{r}_m = \mathbf{p}$.

$$\Delta\mathbf{m}_{\text{good}} = \mathbf{A}^{-1}(\mathbf{A}^{-1})^T\mathbf{g} + \mathbf{m}. \quad (6)$$

The search direction (6) shows a positive-definite operator scaling the gradient. All components of any gradient vector are independent of each other and independently point to a direction for descent. Obviously, each can be scaled by any positive number. Now we have shown that we can also scale a gradient vector by a positive definite matrix and still expect the conjugate-direction algorithm to descend, as always, to the “exact” answer in a finite number of steps. This is because modifying the search direction with $\mathbf{A}^{-1}(\mathbf{A}^{-1})^T$ is equivalent to solving a conjugate-gradient problem in \mathbf{p} .

Application to Bidirectional Deconvolution

Bidirectional deconvolution (Zhang and Claerbout (2010), Shen et al. (2011) and Claerbout et al. (2011)) is a non-linear problem, which has a low convergence rate and unstable result when the starting solution is not close to the true answer. In this section, we apply preconditioning to this problem to obtain a fast and stable result by utilizing prior knowledge. The deconvolution problem is defined as follows:

$$d * a * b^r = \tilde{r}, \quad (7)$$

where d is the data, a and b are the unknown causal filters, and superscript r denotes the time reverse of filter b . The hybrid norm is applied to \tilde{r} , and the reflectivity model is simply \tilde{r} plus a time shift.

We notice that there is only model regularization in this deconvolution problem. Now we change our model from a and b to \tilde{a} and \tilde{b} using $a = p_a * \tilde{a}$ and $b = p_b * \tilde{b}$:

$$d * p_a * p_b^r * \tilde{a} * \tilde{b}^r \approx 0 \quad (8)$$

Thus, we focus on estimating \tilde{a} and \tilde{b} instead of a and b . By applying the prior knowledge on the preconditioners p_a and p_b , we can avoid unwelcome local minima.

GALI-PEF versus PEF preconditioning

In the previous subsections, we showed theoretically that preconditioning leads bidirectional deconvolution to the right global minimum in the nonlinear problem with the help of the prior knowledge from preconditioners p_a and p_b . We have various choices of preconditioners to indicate different prior knowledge. Here we present two kinds of preconditioning, prediction-error filter (PEF) preconditioning and gapped anti-causal leaky integration followed by PEF (GALI-PEF) preconditioning.

On one hand, PEF, whose output is white, is widely used for deconvolution in standard industry practice. The expectation of the whiteness in deconvolution encourages us to use PEF as our preconditioner. Thus we choose PEF as the preconditioner p_a and a spike as the preconditioner p_b in PEF preconditioning. On the other hand, a PEF is a causal filter with a causal inverse. Theoretically, this property adds confidence that deconvolution with a PEF might get the correct phase spectrum as well as the correct amplitude spectrum. However, the wavelet we aim to estimate is not always causal, but can be mixed-phase. In most field data—such as band-limited marine seismic data or land response of an accelerometer—the wavelet is similar to a Ricker wavelet. It is dangerous to deal only with the causal part of the data by using PEF, because it may mislead the bidirectional deconvolution to an incorrect phase spectrum and into an unwelcome local minimum.

Therefore, utilizing the prior knowledge of the anti-causal part of the data becomes necessary. A finite representation of the Ricker wavelet is the negative of the second

finite difference of some binomial coefficients. In Z-transform representation, this is

$$[(1 - 1/z)(1 - z)][(1 + 1/z)^N(1 + z)^N], \quad (9)$$

where N is the order of the binomial coefficient. In real cases, such as the marine data example, there is a time gap between the first ghost and first arrival; thus the numerical representation of the wavelet becomes

$$[(1 - (\rho/z)^g)(1 - (\rho z)^g)][(1 + 1/z)^N(1 + z)^N], \quad (10)$$

where g is an integer which indicates the length of the gap, and ρ is a real number which reduces the energy in a trace and deals with the situation where the gap is not an integer. With this numerical representation of the wavelet, we can divide the data by $[(1 - (\rho/z)^g)]$ to estimate the anti-causal part of the wavelet. The inverse of $[(1 - (\rho/z)^g)]$ is gapped anti-causal leaky integration, which is used as preconditioner p_b . After convolving the data with p_b , we apply PEF to the convolution result and use this estimated PEF as preconditioner p_a . We hope this GALI-PEF preconditioning leads the bi-directional deconvolution to the correct phase spectrum and makes the result fall into the global minimum.

NUMERICAL EXAMPLE

With and without preconditioning

We considered three bidirectional deconvolution methods (Zhang and Claerbout (2010), Shen et al. (2011) and Claerbout et al. (2011)). Of these three methods, the method proposed by Shen et al. (2011) most needs preconditioning. We therefore test our preconditioning on that model to illustrate the effectiveness and limitation of preconditioning.

To illustrate the capabilities of preconditioning, we analyze the results obtained by inverting the zero-phase wavelet. This wavelet is created by convolving the minimum-phase with its own time-reversed wavelet. Figures 2, 3 and 4 show the zero-phase wavelet and its bidirectional deconvolution proposed by Shen et al. (2011), without and with PEF preconditioning. The results show that the wavelet is not completely compressed into a spike without preconditioning, but preconditioning does yield a spike. These results indicate that preconditioning steers the non-linear problem away from unwelcome local minima. However, we can still see slight ringing around the spike in the preconditioned result, indicating that PEF preconditioning does not fully guide the result to the global minimum. This suggests we should introduce more prior knowledge into the preconditioning.

After deconvolving the simple 1D case, we test preconditioning on more complicated 2D synthetic data. Figure 5(a) shows the starting reflectivity model. Figure 5(b) shows the data generated by convolving the reflectivity model with the zero-phase

Figure 2: Zero-phase wavelet. [ER]

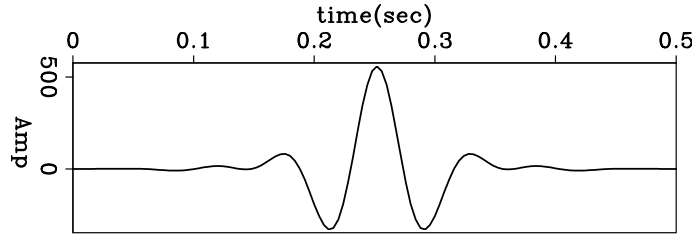
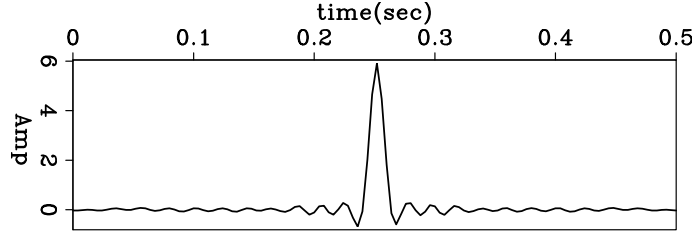


Figure 3: Deconvolution result without preconditioning. [ER]

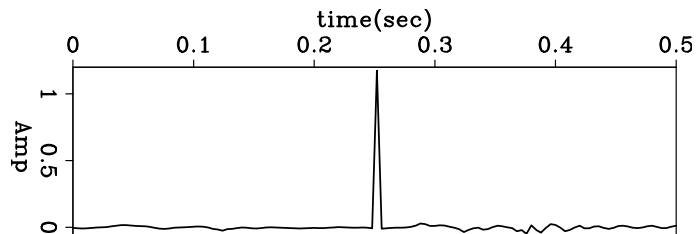


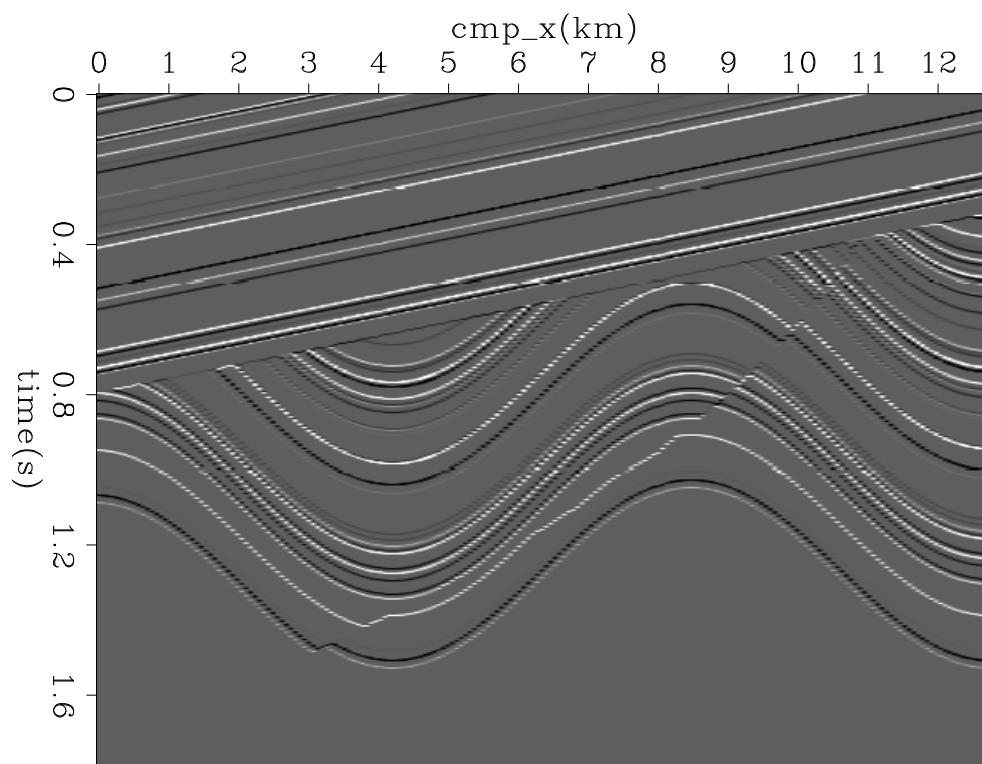
wavelet in the previous section. All traces use the same wavelet when we generate the data, and they share the same wavelet when we are doing the deconvolution.

Figures 6(a) and 6(b) show the bidirectional deconvolution proposed by Shen et al. (2011) without and with PEF preconditioning. The deconvolution model with PEF preconditioning is more spiky than the one without preconditioning, but it still retains some slight ringing around the events. This result is the same as the previous 1D synthetic example, because the same wavelet is used to generate the data in these two sections.

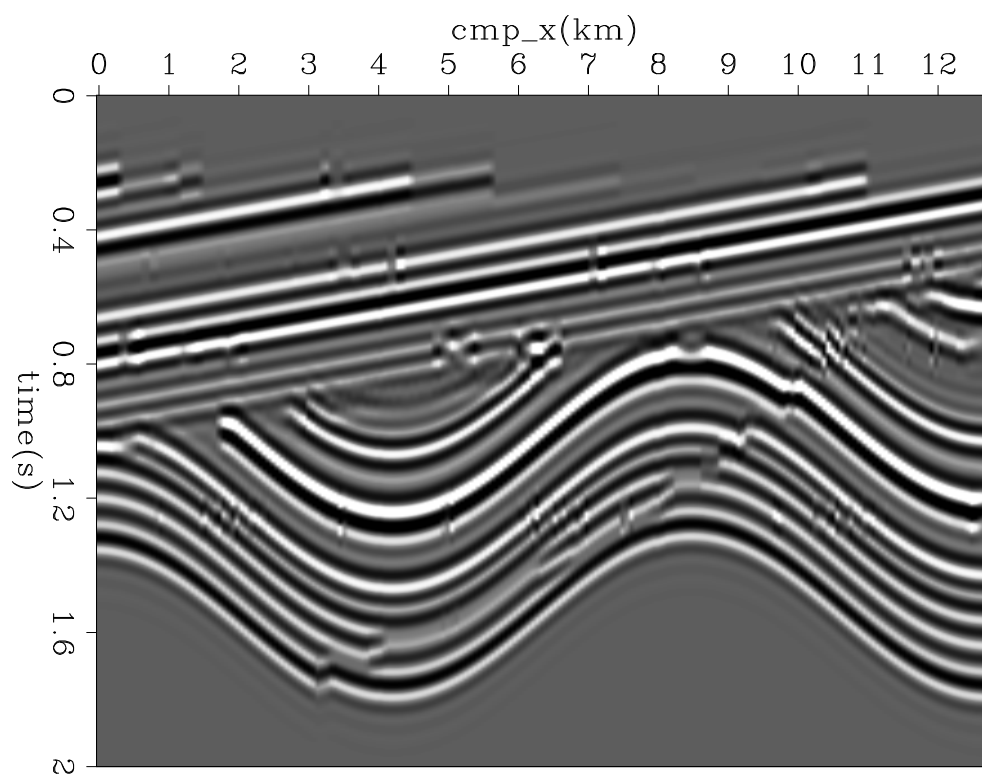
The last example is a common-offset section of marine field data. Figure 7 shows the input data. Figures 8(a) and 8(b) show the bidirectional deconvolution proposed by Shen et al. (2011) without and with PEF preconditioning. Both methods perform well to retrieve the sparse reflectivity in this field data. However, the preconditioned result has fewer precursors and cleaner events than the one without preconditioning. Another important difference is that around 2.4 seconds there is an unknown event appearing in figure 8(a), but it disappears in Figure 8(b). Thus we get a cleaner salt body when we apply preconditioning to this set of field data. The cause of the unknown event is still unidentified, but we have one possible explanation for this event. In this dataset, every trace looks identical, but with a time shift. There are two parallel events between 1.7 sec and 1.8 sec which have almost the same distance for all common midpoints. This phenomenon is unusual and may cause the unknown event, because the distance between the salt top and the unknown event is the same

Figure 4: Deconvolution result with PEF preconditioning. [ER]



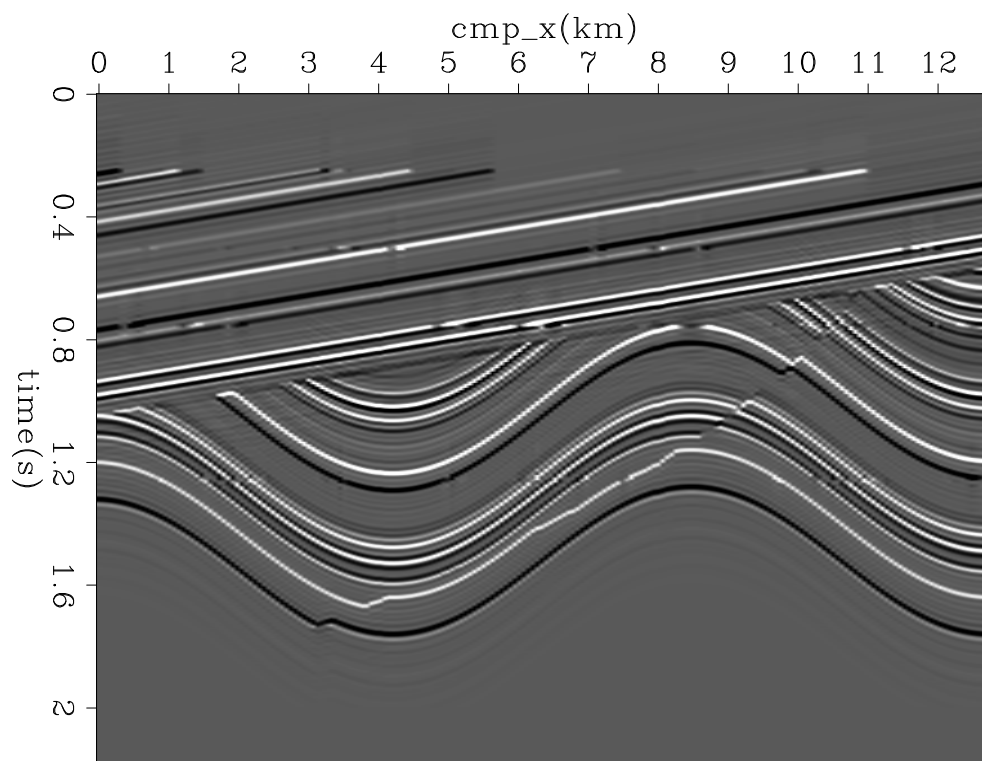


(a)

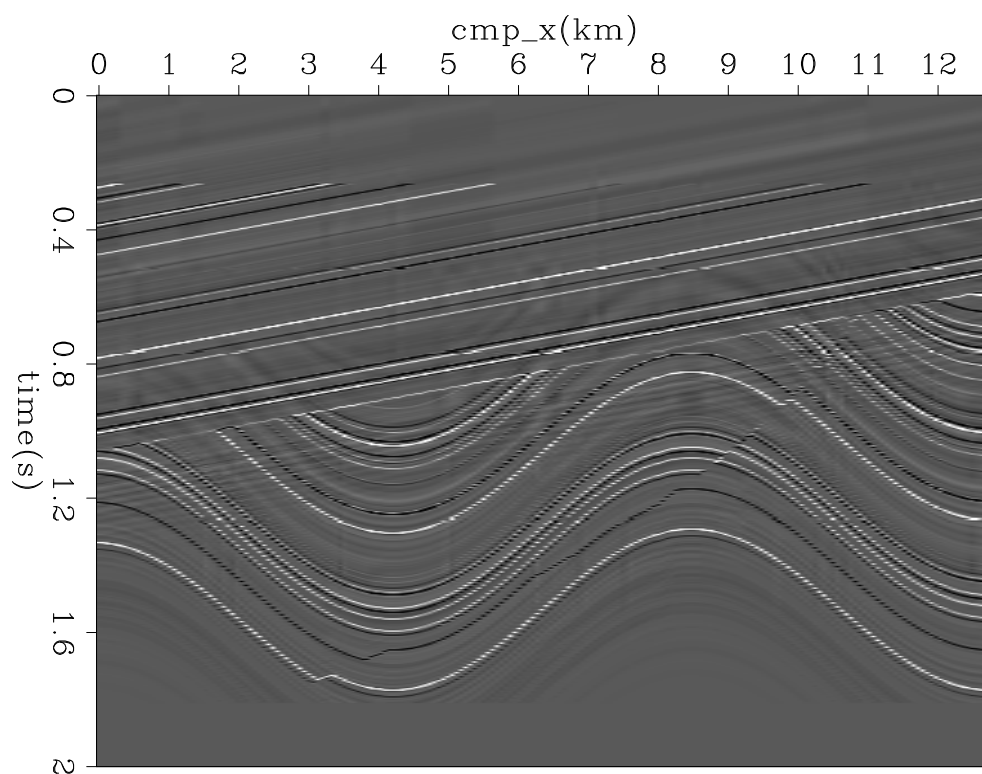


(b)

Figure 5: (a) The 2D synthetic reflectivity model; (b) the synthetic data generated using the zero-phase wavelet. **[ER]**



(a)



(b)

Figure 6: Given the 2D synthetic data in Figure 5(b), (a) reflectivity model retrieved without preconditioning; (b) reflectivity model retrieved with PEF preconditioning. **[ER]**

as that between the two parallel events. We hope the unknown event will disappear if we use another data set with more variable traces.

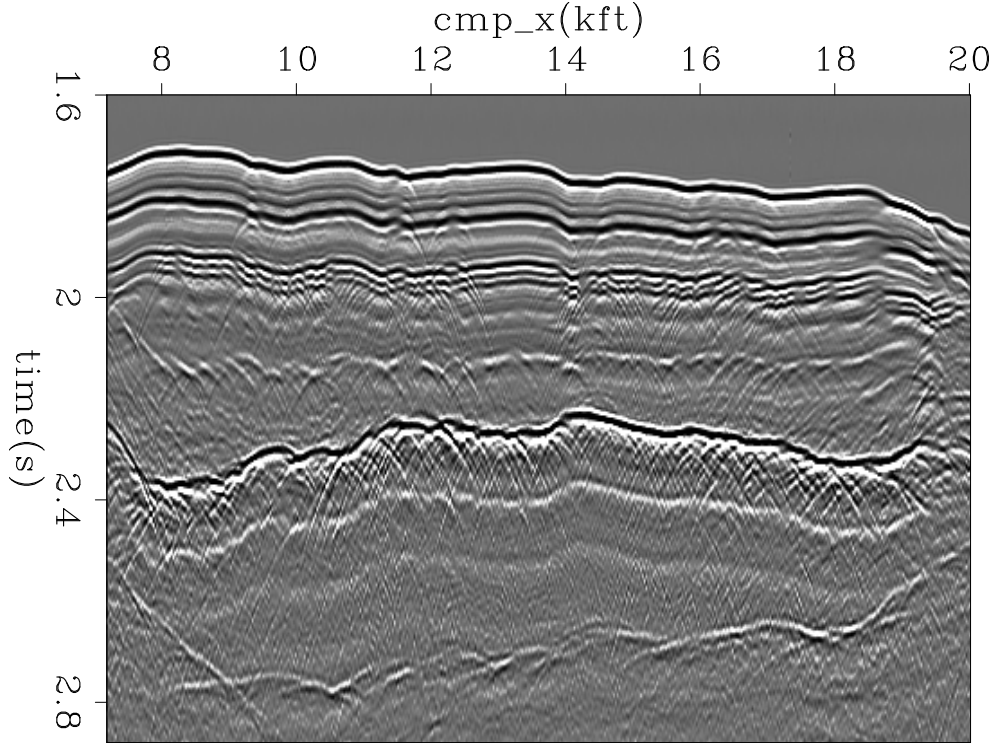


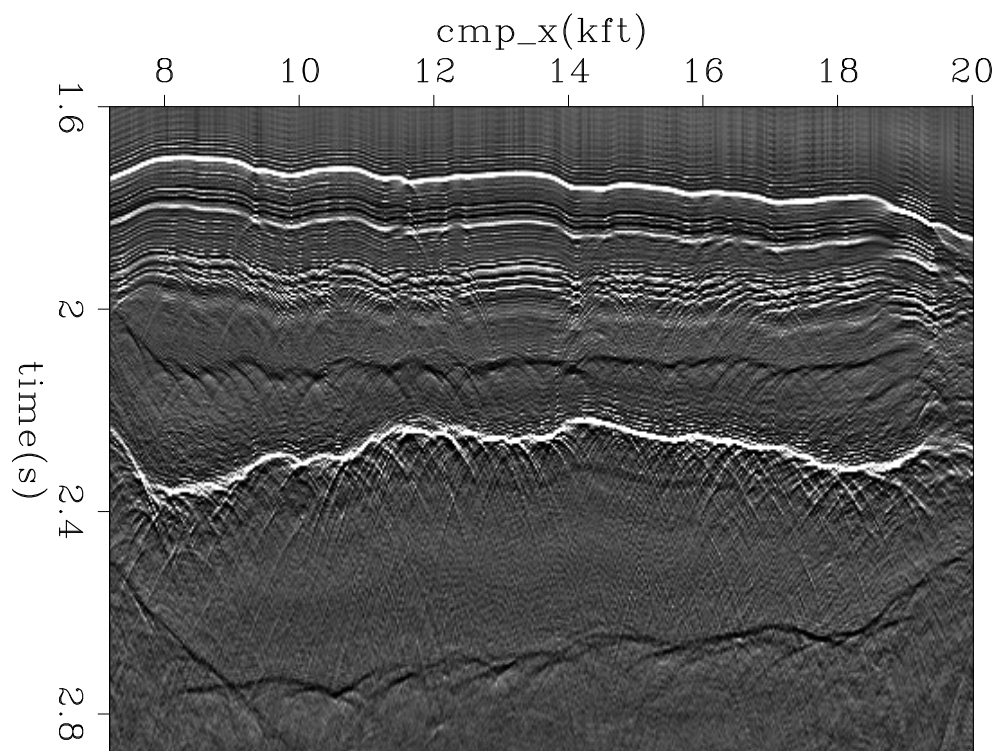
Figure 7: Input Common Offset data. [ER]

Figure 9(a) and 9(b) show the shot wavelet estimated without and with PEF preconditioning. We notice that both results estimate the bubbles and the double ghost, which can be seen in the data. However, the estimated wavelet with preconditioning is more symmetric than the one without preconditioning. This symmetric quality meets our expectation that the wavelet we invert should look like a Ricker wavelet.

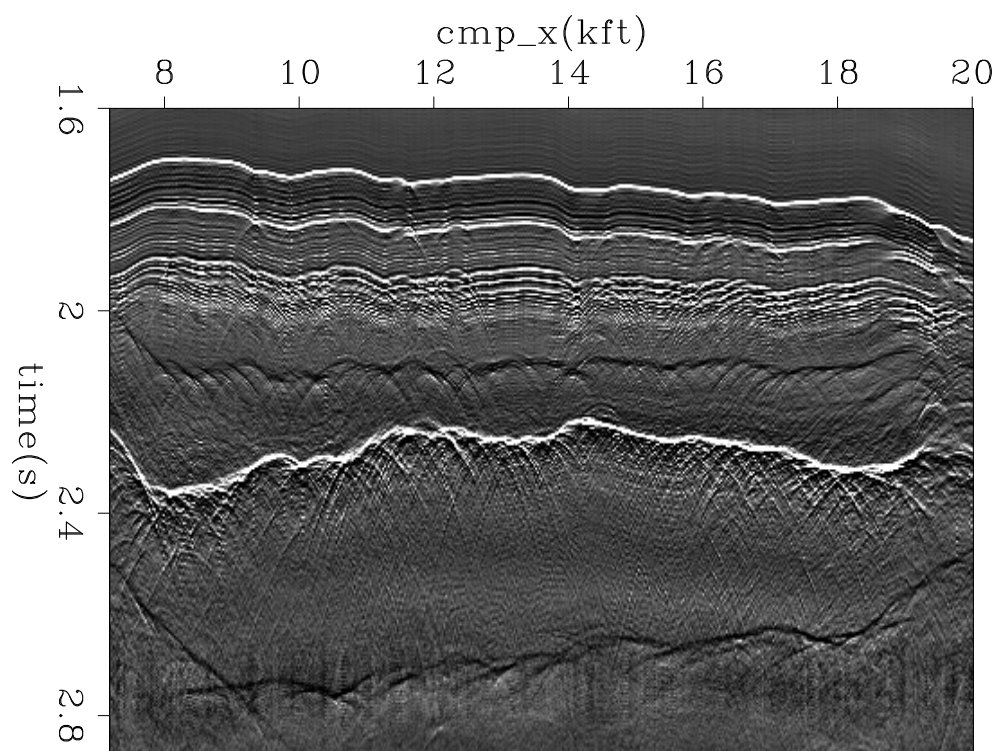
PEF versus GALI-PEF preconditioning

In this subsection, we test the PEF preconditioning and GALI-PEF preconditioning on bidirectional deconvolution. Fu et al. (2011) shows that the method proposed by Claerbout et al. (2011) produces most stable result among the three bidirectional deconvolution methods considered above. Therefore, we use Claerbout et al. (2011) to compare these two preconditionings to make the comparison reliable.

We use the same field data shown in the previous subsection for this example. First, we convolve the data with PEF and GALI-PEF preconditioning respectively, as is shown in Figure 10. Then we apply bidirectional deconvolution to the convolution results, as is displayed in Figure 11. We may draw the following conclusions from the comparison results.



(a)



(b)

Figure 8: Given the common offset data in Figure 7, (a) reflectivity model retrieved without preconditioning; (b) reflectivity model retrieved with PEF preconditioning. [ER]

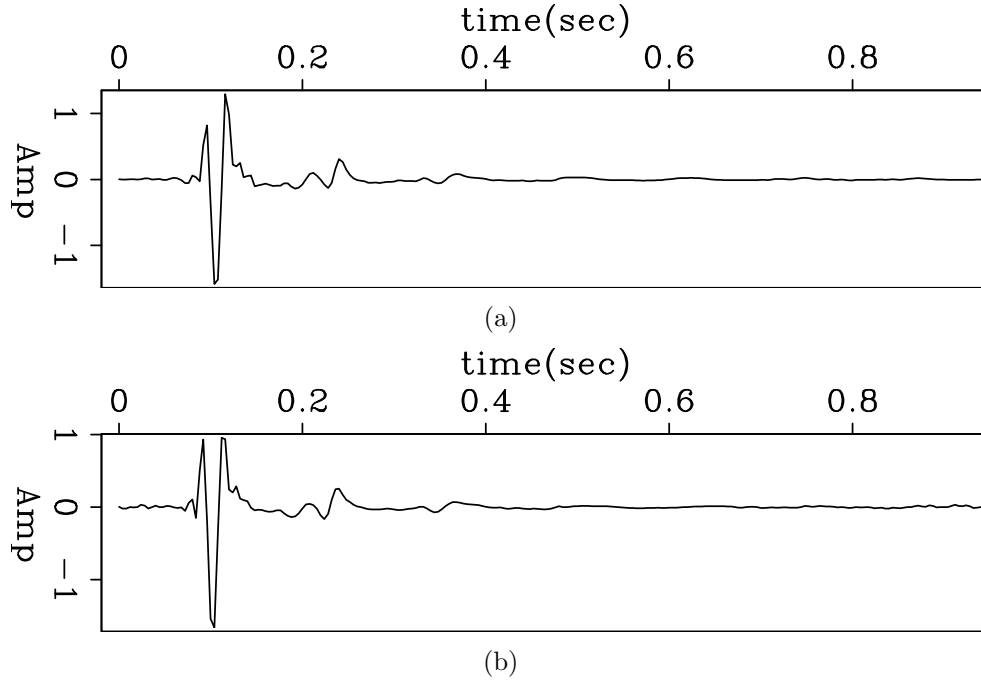


Figure 9: Given the common offset data in Figure 7, (a) shot wavelet estimated without preconditioning; (b) shot wavelet estimated with PEF preconditioning. [ER]

GALI-PEF preconditioning helps produce the spike in the central wavelet.

As the data shows, the events in Figure 7 look like Ricker wavelet, with two weak side lobes and one strong middle lobe. We expect to spike the result in the strong middle lobe. Because PEF is a causal filter with causal inverse, it produces the output in the first lobe of Ricker. Thus the polarity of the output is the same as the first lobe of Ricker. From panel (b) in Figure 10, to take the water bottom for example, the strong event is black. This polarity, as well as its output location, is the same as the first lobe of mixed-phase wavelet around 1.8 seconds shown in Figure 10. Focusing on the first lobe in preconditioning leads to same effect in the bidirectional deconvolution. Panel (b) in Figure 11 shows exactly the same outcome that the output is in the same location and polarity as the first lobe of Ricker wavelet. However, GALI-PEF preconditioning helps shift the time of output. Panel (c) in Figure 10 shows the event is produced in the same location and polarity as the middle Ricker. So are their bidirectional deconvolution results. To take the water bottom for example, the event appears white in both GALI-PEF preconditioning and its bidirectional deconvolution result, which is the same as the middle lobe of the wavelet. Besides helping produce the spike in the middle Ricker, GALI-PEF preconditioning can make the output in any position by changing the length of gap. In this case, the gap between the first ghost and first arrival is roughly 10-15 millisecond. If the gap in GALI-PEF preconditioning is longer than this separation, the output will move towards the second side lobe of the wavelet and visa versa.

Unfortunately, however, what disappoints us is that GALI-PEF preconditioning

does not improve the result compared to PEF preconditioning. Both the PEF and GALI-PEF preconditioning results are almost the same except for reversed polarity and time shift. In addition, the precursors in Figure 10(c) are strong, because of the anti-causal integration. From another perspective, although the GALI-PEF preconditioner produces a noisier, more resonant section than does PEF, that section illustrates polarity more clearly than does PEF. Also the interval between every two nearby precursors illustrates the gap between first ghost and first arrival.

Both preconditioning methods speed convergence. The convergence rates with and without preconditioning are shown in Figure 12. The average mismatch here is measured by using a hybrid penalty function (Claerbout (2010)):

$$\frac{\bar{r}}{R} = r(\bar{H}) = \sqrt{\left(\frac{1}{N} \sum_{i=1}^N \sqrt{1 + \frac{r_i^2}{R^2}}\right)^2 - 1} \quad (11)$$

where $H(r) = \sqrt{R^2 + r^2} - R$, and R is the threshold. This expression of the misfit is dimensionless and reflects the speed of convergence. We notice that these three convergence curves in Figure 12 originate from different points, because the average residual without preconditioning is calculated directly from the raw data, whereas the ones with the two preconditionings methods are calculated from the data transformed by PEF and GALI-PEF preconditionings respectively. Thus, we only consider the relative trend, not the absolute value, of the curves. We notice that the convergence rates drop somewhat with preconditioning, because either PEF or GALI-PEF already helps reduce the average mismatch. However, convergence is reached soon after 30 iterations with the help of preconditioning, whereas without preconditioning convergence takes more than 55 iterations. Therefore, preconditioning does reduce the computational cost.

Both methods of preconditioning improve bidirectional deconvolution. The logarithm bidirectional deconvolution proposed by Claerbout et al. (2011) is more stable than the one proposed by Shen et al. (2011), which estimates the filters in the Fourier domain. Thus the result depends less on preconditioning in the logarithm method. However, we still notice that both methods of preconditioning improve the results by reducing precursors. In addition, the unknown event around 2.4 sec in panel (a) of Figure 11 becomes weaker in the results with preconditioning, especially in bidirectional deconvolution with PEF preconditioning.

CONCLUSION

In this paper, we illustrate the importance of preconditioning on non-linear problems, and we apply preconditioning to bidirectional deconvolution. The results in three data examples show that wavelets are more spiky in the results with preconditioning than in those without preconditioning. However, the results with preconditioning in the 1D and 2D synthetic sections show slight ringing around the spike, which

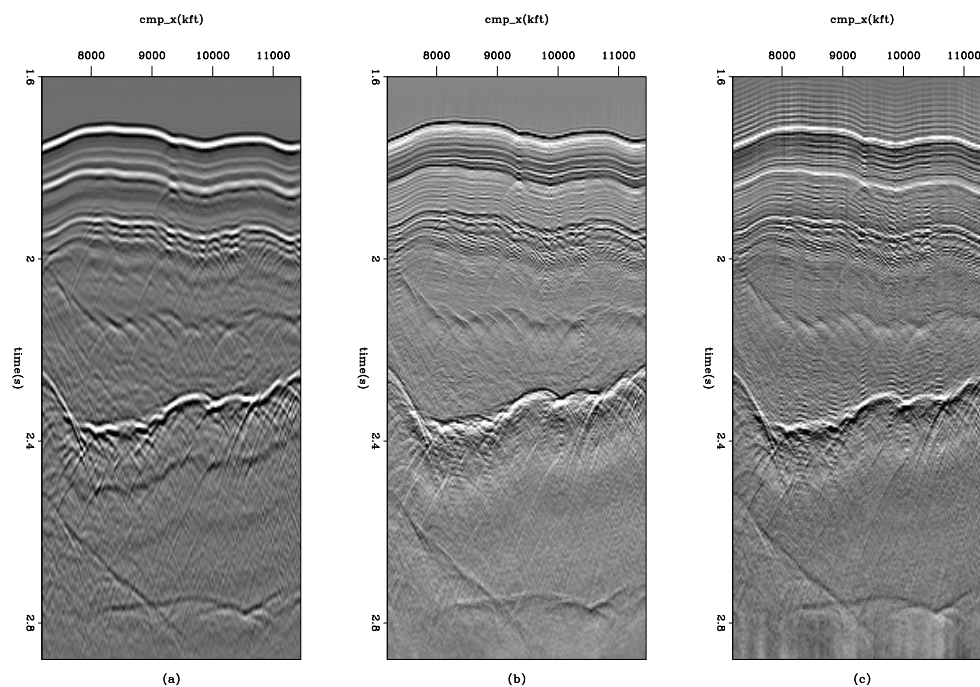


Figure 10: Given the common offset data in Figure 7, (a) 1/3 of original data; (b) data transformed by PEF preconditioning; (c) data transformed by GALI-PEF preconditioning. These three panels are the inputs to the bidirectional deconvolution output in Figure 11. [ER]

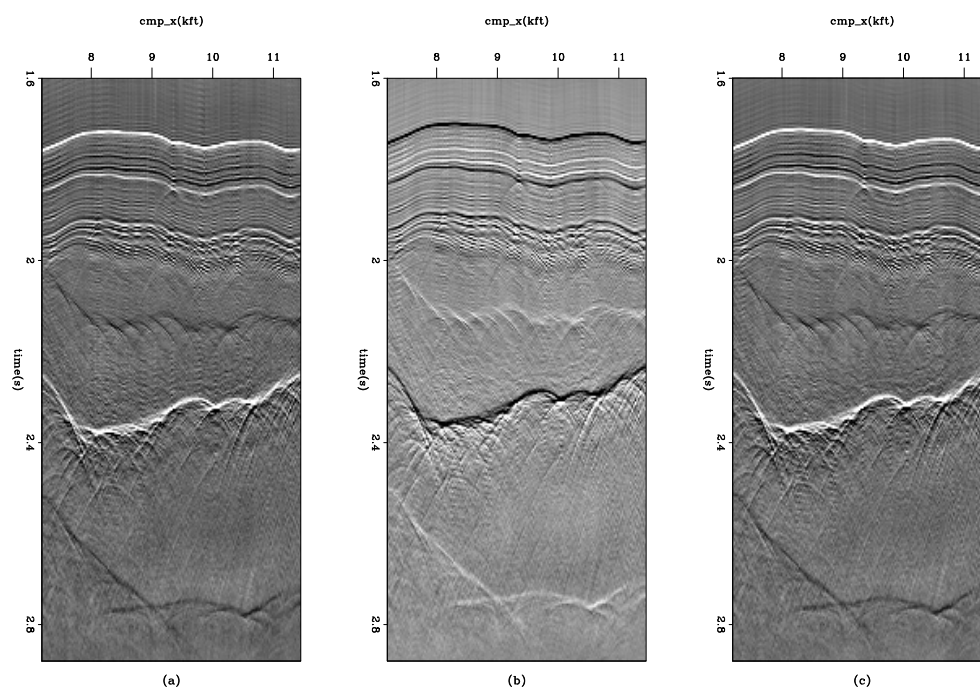
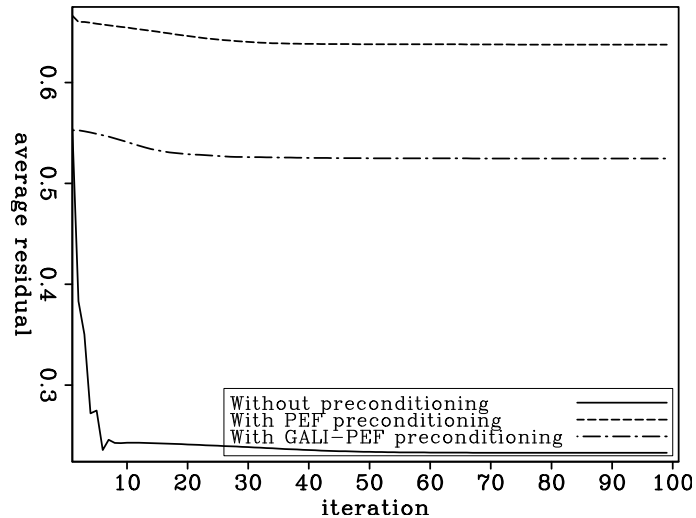


Figure 11: Given the common offset data in Figure 7, (a)bidirectional deconvolution without preconditioning; (b)bidirectional deconvolution with PEF preconditioning (c) bidirectional deconvolution with GALI-PEF preconditioning. [ER]

Figure 12: Convergence rate of the results in Figure 11. [ER]



may encourage us to use more prior knowledge in the preconditioning. For field data, the results with preconditioning have fewer precursors, a cleaner salt body, and a more symmetric wavelet than those without preconditioning. This proves that preconditioning, which utilizes prior information, can guide the gradient along sensible pathways, thus avoiding potential local minima, making the results more reliable, and speeding convergence.

In addition, we introduce two methods of preconditioning: PEF and GALI-PEF, and apply them to the field data. Both approaches improve the bidirectional deconvolution result and improve the convergence speed. But unlike PEF preconditioning, GALI-PEF preconditioning helps output the spike in the central wavelet or other positions of the wavelet if we change the length of gap. However, we have tested these two methods on only one set of field data. More experiments in other datasets are needed to illustrate the effectiveness and limitation of these two methods of preconditioning in our future work.

ACKNOWLEDGMENTS

The authors thank Shuki Ronen for his idea of applying a gap in anti-causal leaky integration, and we thank Dave Nichols, Robert Clapp, Yang Zhang, Antoine Guitton for fruitful discussion.

REFERENCES

- Claerbout, J., 2010, Image estimation by example.
- Claerbout, J., Q. Fu, and Y. Shen, 2011, A log spectral approach to bidirectional deconvolution: SEP-Report, **143**, 295–298.
- Fu, Q., Y. Shen, and J. Claerbout, 2011, Data examples of logarithm fourier domain bidirectional deconvolution: SEP-Report, **145**, ???–???

- Shen, Y., Q. Fu, and J. Claerbout, 2011, A new algorithm for bidirectional deconvolution: SEP-Report, **143**, 271–281.
- Zhang, Y. and J. Claerbout, 2010, A new bidirectional deconvolution method that overcomes the minimum phase assumption: SEP-Report, **142**, 93–103.