

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 highly dependent on the initial guess. Here we introduce preconditioning and show how it helps resolve in our current problem—bidirectional deconvolution. The results in three data examples show that preconditioning helps us get a more spiky result when compared with the results without preconditioning. Additionally field data results with preconditioning have fewer precursors, cleaner salt body, more symmetric wavelet, and faster convergence rate than those without preconditioning. In addition to the field data, we theoretically illustrate and practically apply two ways of preconditioning: Prediction-error filter (PEF) preconditioning and Gapped anti-causal leaky integration followed by PEF (GALIP) preconditioning. Unlike PEF preconditioning, GALIP helps produce the result 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 theory and techniques than any one person can master in a lifetime. You 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. First, avoid creating difficulties for yourself by always using a convex penalty function. If you don’t, you may quickly find yourself in a world where your final answer depends strongly on your first guess. Nobody likes that.

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 with linear equations you may start anywhere, while with nonlinear relationships, you had better have a sensible starting solution.

Before arriving here you have seen many linear regressions and likely saw prior information introduced in the form of regularization. Preconditioning is a well established technique with linear regressions to hasten convergence by utilizing prior

information. Preconditioning usually begins from regularization. Preconditioning steers the iterative descent along the path set out by a prior model, but 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. The truth is, every application offers us a choice of coordinate systems and ways to parameterize the model. The choice taken is arbitrary. 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 so long as its polarity is unchanged. That means any gradient can be multiplied by any diagonal matrix that contains 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 really is positive definite. At early stages of descent, you should scale that gradient large where you have confidence, and small where you do not. With linear regressions this has no effect on the solution. With nonlinear physics it steers 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 where the model will be findable. We need (in nonlinear cases) to be certain such regions do not distract our descent, especially in early iterations. So we should view our gradient both in the model space and in data space. Then choose any diagonal weighting you like, and choose any filter you like. Perhaps you are doubtful about Nyquist frequency, zero frequency, and long lags, especially anticausal lags. Dream up your own filter \mathbf{F} and weight \mathbf{W} . To your gradient apply either \mathbf{FW} or \mathbf{WF} . Then also apply its transpose so you will have applied either $(\mathbf{FW})^T(\mathbf{FW})$ or $(\mathbf{WF})^T(\mathbf{WF})$. This procedure throws away no information in the data. It merely selects what aspects of the data are used first. As the final solution is approached the gradient diminishes; your down-weighted regions eventually emerging in the gradient because they are the only things left. Now you are much closer to the ultimate solution where it is far less dangerous having your down-weighted regions affecting the solution.

THEORY

Preconditioning Offers Smart Directions

We start from fitting goals

$$\begin{aligned}\mathbf{0} &\approx \mathbf{F}\mathbf{m} - \mathbf{d} \\ \mathbf{0} &\approx \mathbf{A}\mathbf{m}\end{aligned}\tag{1}$$

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

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

Without preconditioning we have the search direction

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

and with preconditioning we have the search direction

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

The essential feature of preconditioning is not that we perform the iterative optimization in terms of the variable \mathbf{p} . The essential feature is 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.

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

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

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

The search direction (6) shows a positive-definite operator scaling the gradient. Each component of any gradient vector is independent of each other. All independently point to a direction for descent. Obviously, each can be scaled by any positive number. Now we have found 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 on Bidirectional Deconvolution

Bidirectional deconvolution (Zhang and Claerbout (2010), Yi Shen and Claerbout (2011) and Jon Claerbout and Shen (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 p_a and p_b using $a = A * p_a$ and $b = B * p_b$

$$d * A * B^r * p_a * p_b^r \approx 0 \quad (8)$$

Thus, our job focuses on estimating p_a and p_b instead of a and b . By applying the prior knowledge on the preconditioners A and B , we can keep away from the unwelcome local minima.

GALIP 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 A and B . Thus, we have various choices of preconditioners to indicate different prior knowledge. In the paper, we present two ways of preconditioning, Prediction-error filter (PEF) preconditioning and Gapped anti-causal leaky integration followed by PEF (GALIP) preconditioning.

On one hand, PEF, whose output is white, is widely used to do 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 A and a spike as the preconditioner B in PEF preconditioning. On the other hand, a PEF is a causal filter with a causal inverse. Theoretically, this property adds confidence to 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 mixed-phase. In most field data the wavelet is similar to a Ricker wavelet, such as band-limited marine seismic data, the land response of an accelerometer and so on. It is a danger to only deal with the causal part of the data by using PEF because it may mislead the bidirectional deconvolution to a wrong 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 $[(1 - 1/z)(1 - z)][(1 + 1/z)^N(1 + z)^N]$, where N is the order of the binomial coefficient. In real cases, to take the marine data for example, there is a time gap between the first ghost and first arrival, thus the numerical representation of the wavelet becomes $[(1 - (\rho/z)^g)(1 - (\rho z)^g)][(1 + 1/z)^N(1 + z)^N]$, 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 B . After convolving the data with B , we apply PEF to the convolution result and use this estimated PEF as preconditioner A . We hope this GALIP preconditioning leads the bi-directional deconvolution to a right phase spectrum and makes the result fall into the global minimum.

In this paper, we first use the PEF preconditioning on bidirectional deconvolution proposed by Yi Shen and Claerbout (2011) in which preconditioning is most needed among three methods (Zhang and Claerbout (2010), Yi Shen and Claerbout (2011) and Jon Claerbout and Shen (2011)) and illustrate the effectiveness and limitation of preconditioning. Then we test the PEF preconditioning and GALIP preconditioning on bidirectional deconvolution proposed by Jon Claerbout and Shen (2011) which produces most stable result among three methods, and a comparison is made between these two preconditionings.

NUMERICAL EXAMPLE

Single Wavelet

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 1, 2 and 3 show the zero-phase wavelet and its bidirectional deconvolution proposed by Yi Shen and Claerbout (2011), without and with preconditioning. The results show that the wavelet is not completely compressed into a spike without preconditioning, but it turns out to be a perfect spike with preconditioning. These results indicate that preconditioning steers the non-linear problem away from the unwelcome local minima. However, we can still see some small ringing around the spike in the preconditioning result. Such phenomenon indicates that PEF preconditioning does not fully guide the result into the global minimum, and thus we are encouraged to use more prior knowledge on the preconditioning.

Figure 1: Zero-phase wavelet. [ER]

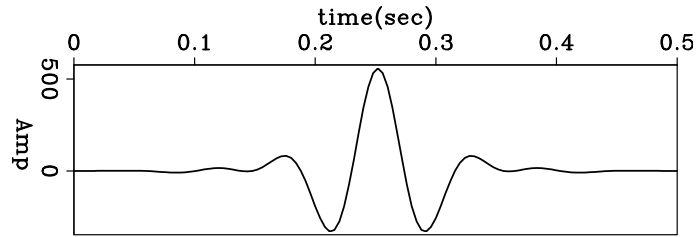
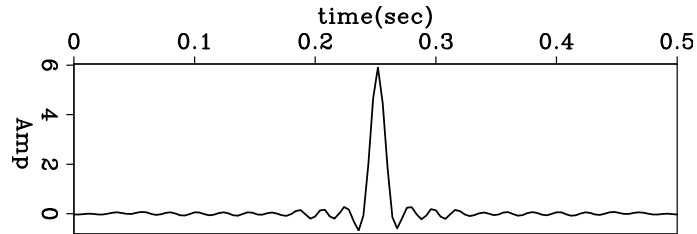


Figure 2: Deconvolution result without preconditioning. [ER]



2D Synthetic Data

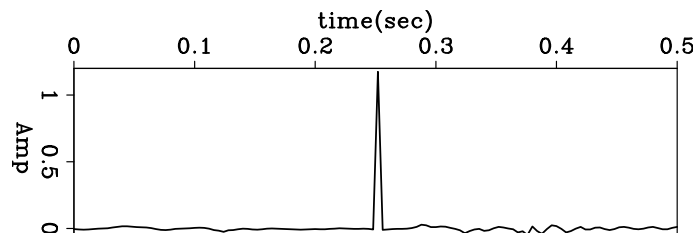
After applying deconvolution on the simple 1D case, we test preconditioning on more complicated 2D synthetic data. Figure 4(a) shows the starting reflectivity model. Figure 4(b) shows the data generated by convolving the reflectivity model with the zero-phase wavelet in the previous section. All traces use the same wavelet when we generate the data, and they share the same wavelet when we are doing the deconvolution.

Figures 5(a) and 5(b) show the bidirectional deconvolution proposed by Yi Shen and Claerbout (2011) without and with preconditioning. The deconvolution model with preconditioning is more spiky than the one without preconditioning. However, it has some small ringings around the events in the result with preconditioning. Such phenomena are just the same as the previous 1D synthetic example, because the same wavelet is used to generate the data in these two sections.

2D Field Data

The field data we use in this example is a common-offest section of marine field data. Figure 6 shows the input data. Figures 7(a) and 7(b) show the bidirectional deconvolution without and with preconditioning. Both methods perform well to retrieve

Figure 3: Deconvolution result with preconditioning. [ER]



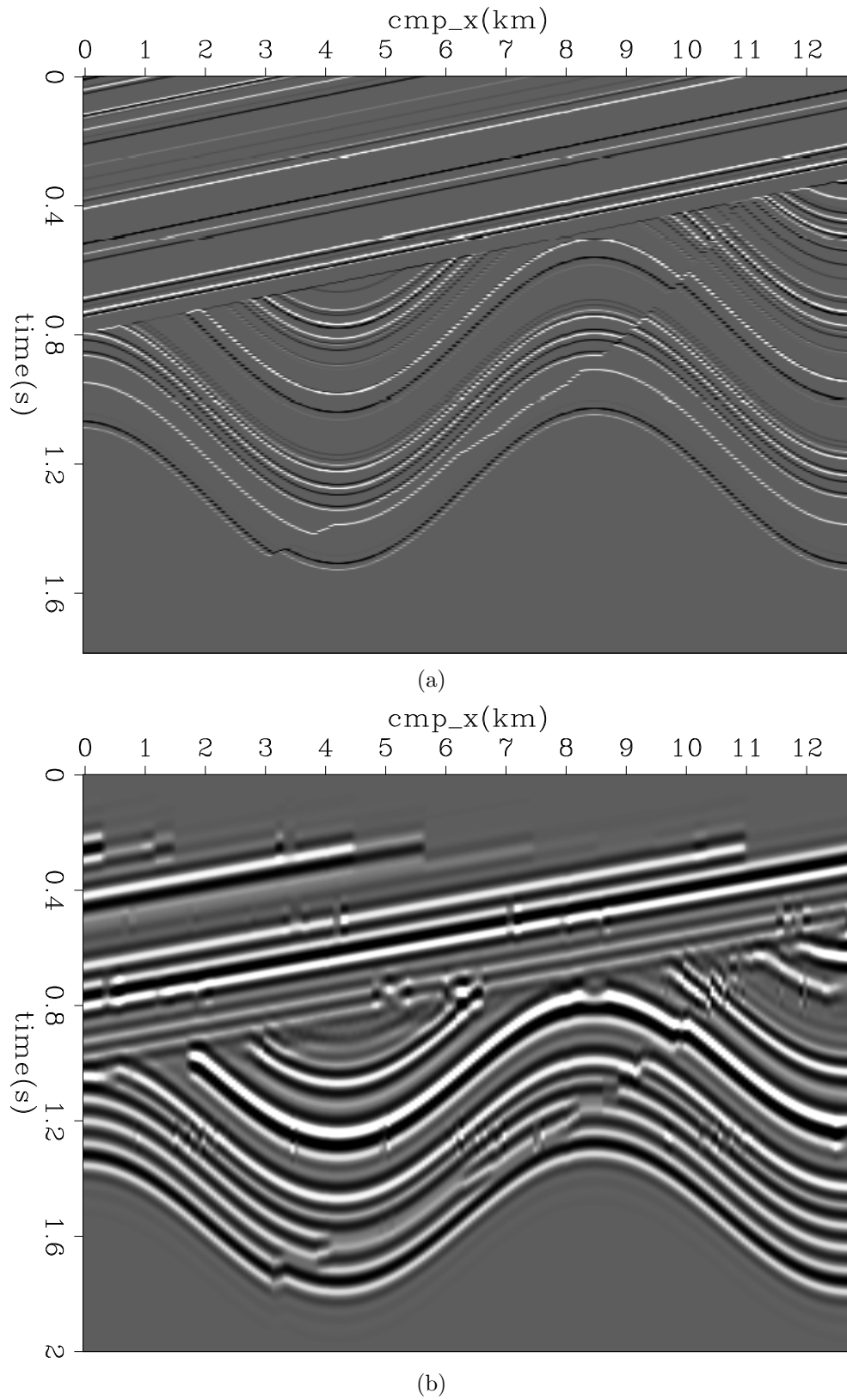
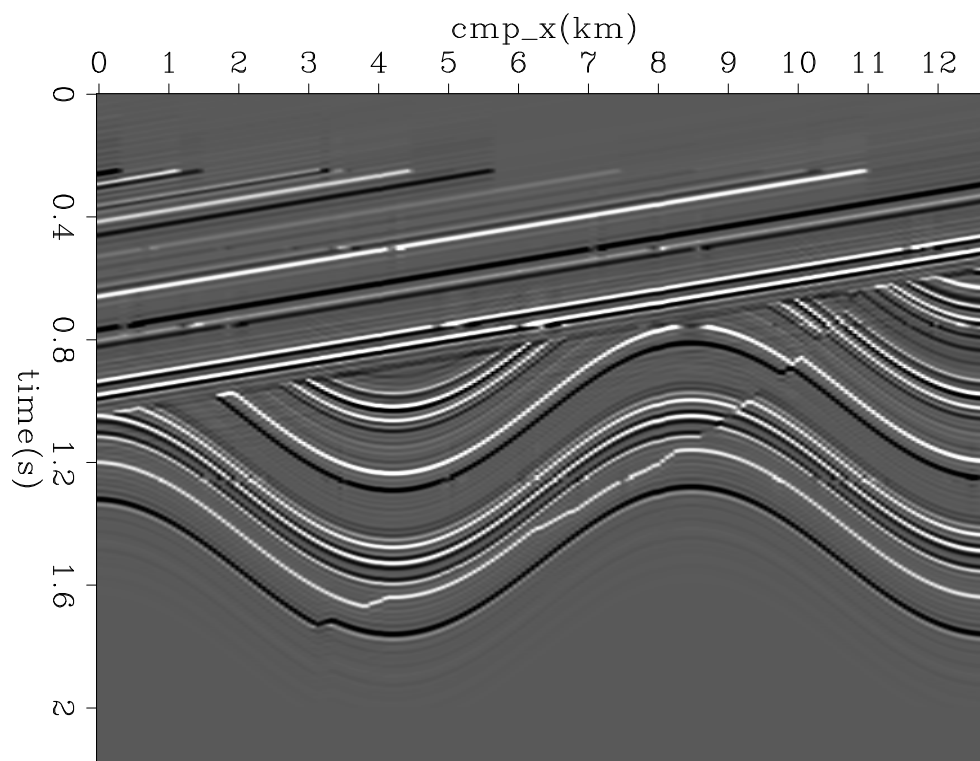
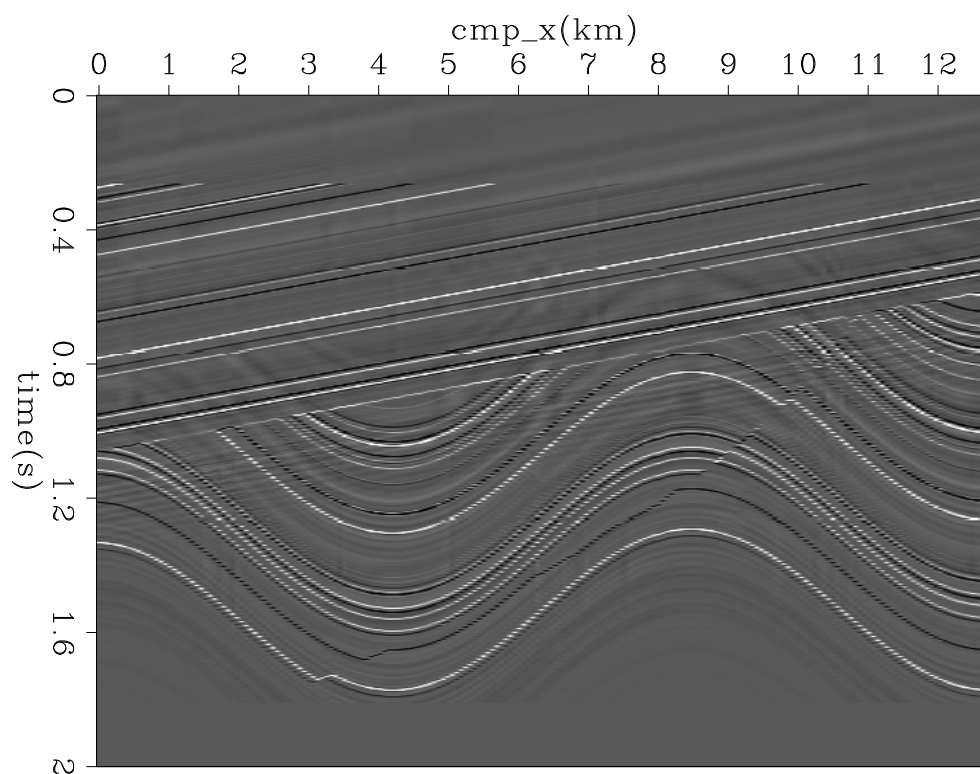


Figure 4: (a) The 2D synthetic reflectivity model; (b) the synthetic data generated using the zero-phase wavelet. [ER]



(a)



(b)

Figure 5: Given the 2D synthetic data in Figure 4(b), (a) reflectivity model retrieved without preconditioning; (b) reflectivity model retrieved with preconditioning. [ER]

the sparse reflectivity in this field data. However, the preconditioning result has less precursors and cleaner events than the one without preconditioning. Another important thing to notice is that around 2.4 sec there is an unknown event appearing in figure 7(a), but it disappears in Figure 7(b). Thus we get a cleaner salt body when we apply preconditioning on this field data.

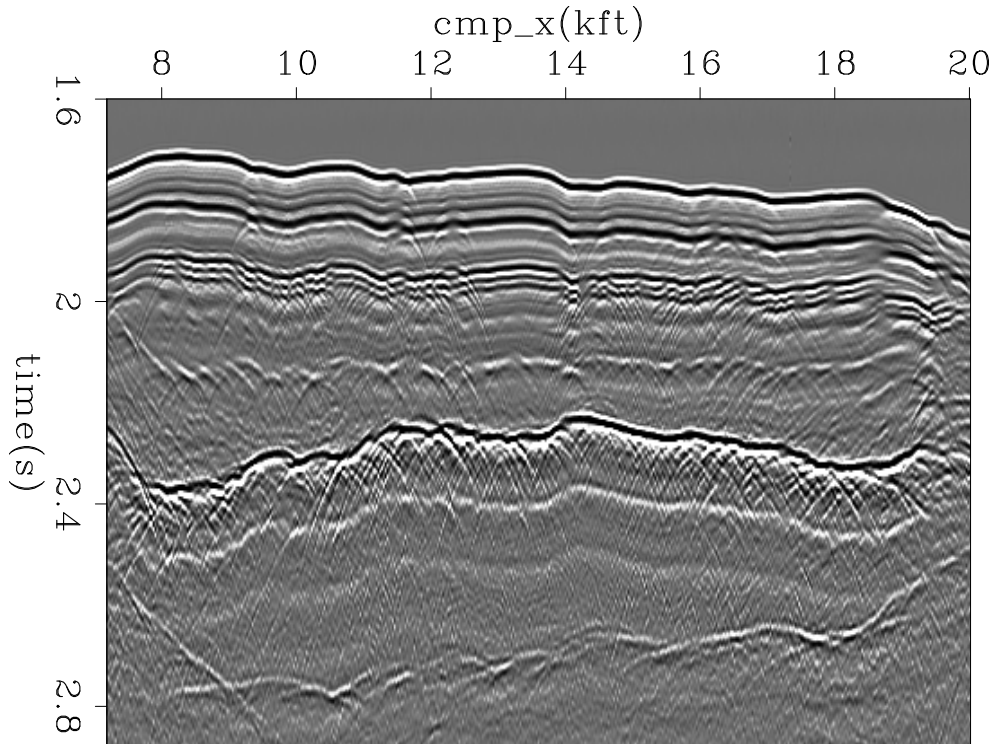
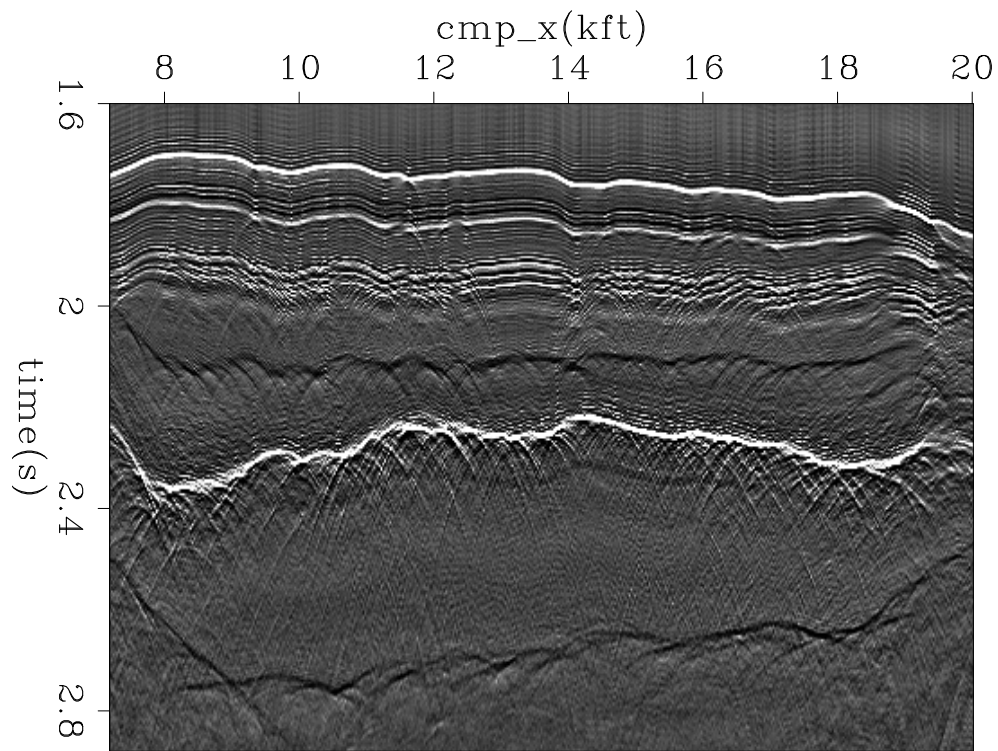


Figure 6: Input Common Offset data. [ER]

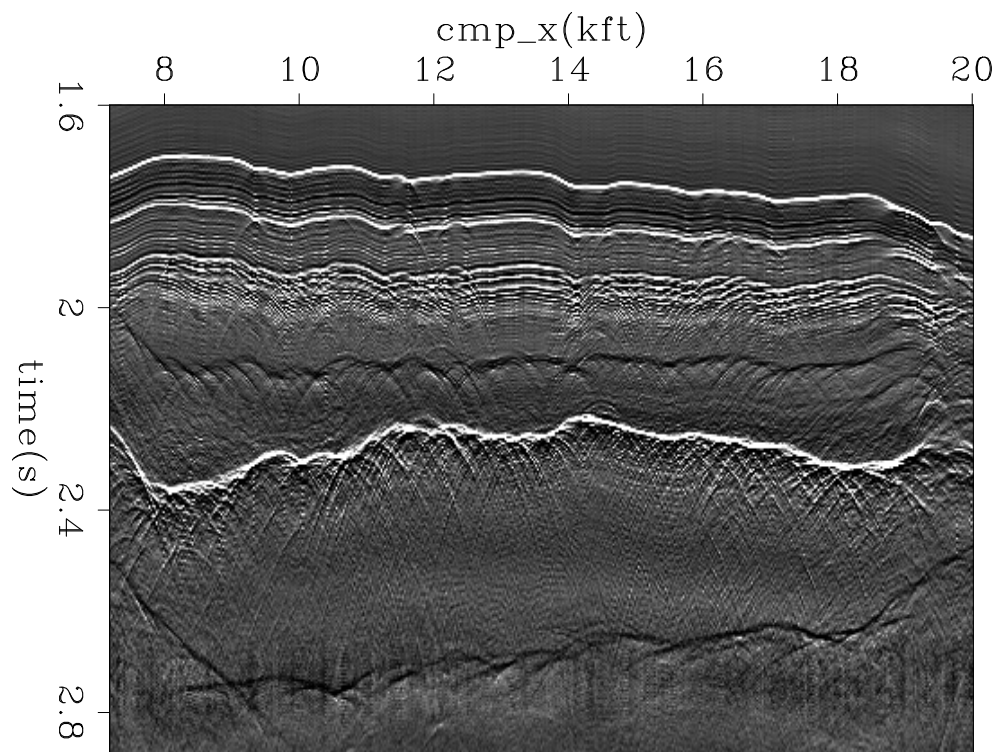
Figure 8(a) and 8(b) show the shot wavelet estimated without and with preconditioning. We notice that both results estimate the bubbles and the double ghost, which can be seen in the data. However, the estimated wavelet with preconditioning is more symmetric than the one without preconditioning. This symmetric quality meets our expectation, because we hope that the wavelet we invert looks like a Ricker wavelet, just as the data shows.

We also compare the convergence rate with and without preconditioning, which are shown in Figure 9. We notice that after preconditioning, the convergence rate drops a little and gets convergence soon after 60 iterations. However, the result without preconditioning gets converged after 250 iterations. Therefore, preconditioning makes the result converge faster and helps reduce the computational cost.

In this field data, we also test PEF and GALIP preconditioning on bidirectional deconvolution proposed by Jon Claerbout and Shen (2011) which produces more stable results than other two methods proposed by Zhang and Claerbout (2010) and Yi Shen and Claerbout (2011), and a comparison is made between these two preconditionings.



(a)



(b)

Figure 7: Given the common offset data in Figure 6, (a) reflectivity model retrieved without preconditioning; (b) reflectivity model retrieved with preconditioning. [ER]

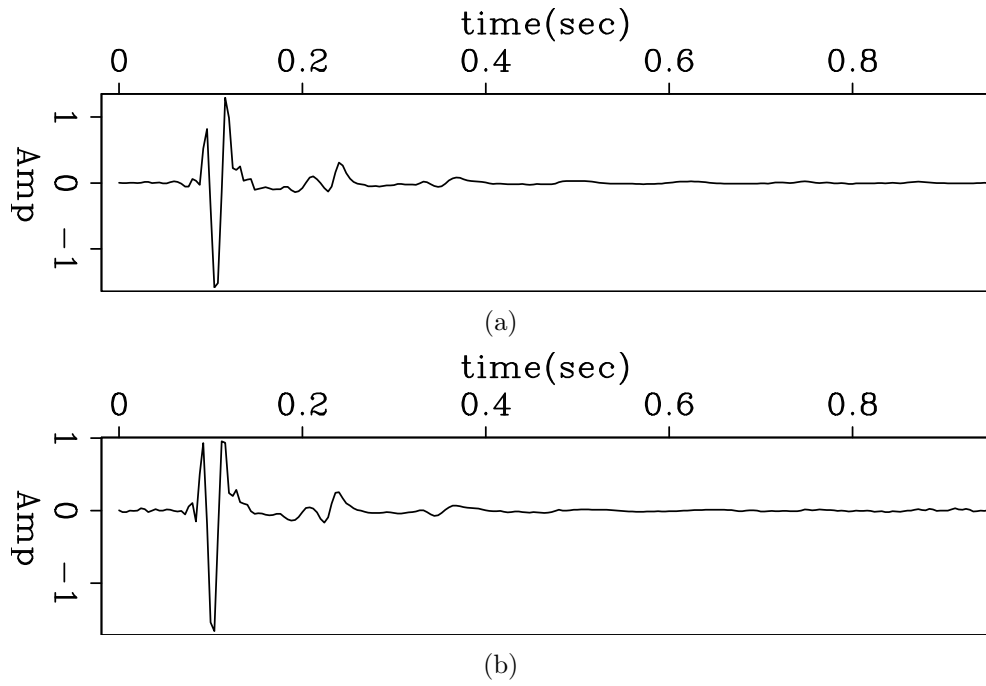
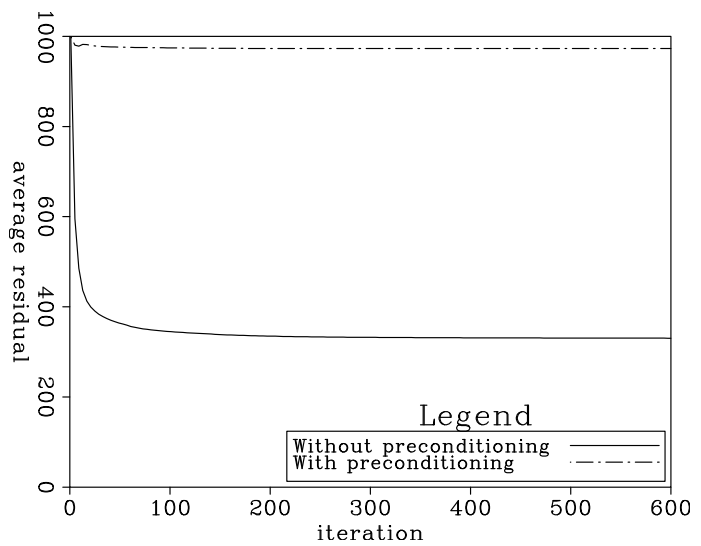


Figure 8: Given the common offset data in Figure 6, (a) shot wavelet estimated without preconditioning; (b) shot wavelet estimated with preconditioning. [ER]

Figure 9: Convergence rate with and without preconditioning. [ER]



We convolve the data with PEF and GALIP respectively as preconditioning, which are 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.

GALIP helps produce the result in the central wavelet. As the data shows, the events in Figure 6 look like Ricker wavelet, with two weak side lobes and one strong middle lobe. We expect the output of our result to be produced in 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, GALIP 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 is its bidirectional deconvolution result. To take the water bottom for example, the event appears white in both GALIP preconditioning and its bidirectional deconvolution result, which is the same as the middle lobe of the wavelet. But both of the PEF and GALIP results are almost the same if one of them is polarity flipped and time shifted. Besides helping produce the result in the middle Ricker, GALIP 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 millisecc. If the gap in GALIP is longer than this separation, the output will move towards the second side lobe of the wavelet and visa versa.

Both preconditionings improve bidirectional deconvolution. Because of the logarithm bidirectional deconvolution proposed by Jon Claerbout and Shen (2011), the result is more stable than the one proposed by Yi Shen and Claerbout (2011) by estimating the filters in Fourier domain. Thus the result depends less on preconditioning in the logarithm method. However, from the bidirectional deconvolution results, we still notice that both ways 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 problem, and apply preconditioning to bidirectional deconvolution. The results in three data examples show that wavelets are compressed into more spiky spikes in the results with preconditioning in comparison with those without preconditioning. However, the results with preconditioning in the 1D and 2D synthetic sectionsthere show some

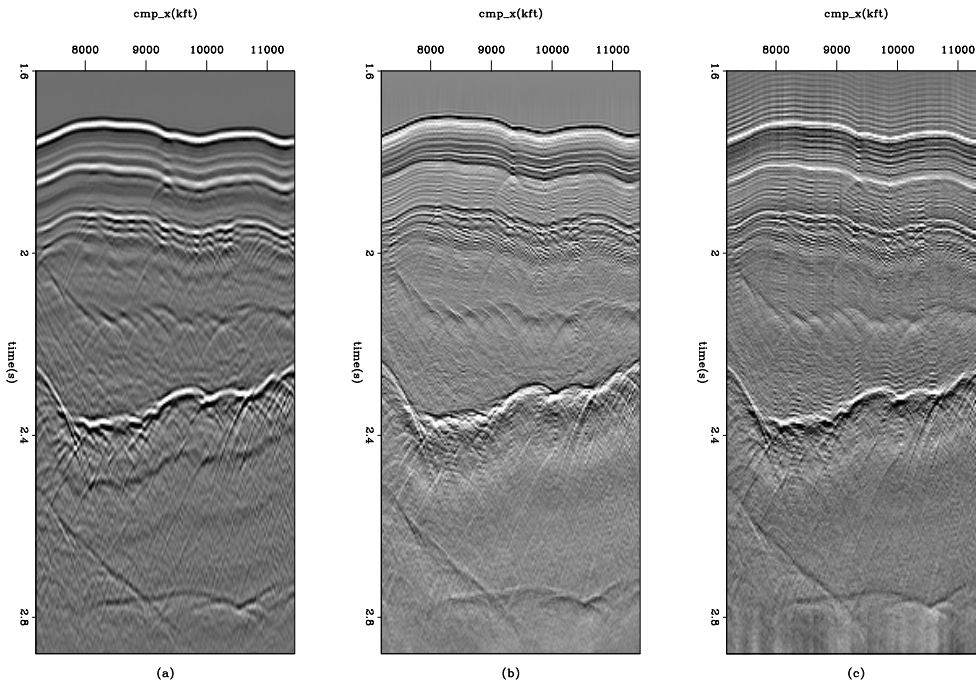


Figure 10: Given the common offset data in Figure 6, (a)one third of the data; (b)PEF preconditioning (c) GALIP preconditioning. [ER]

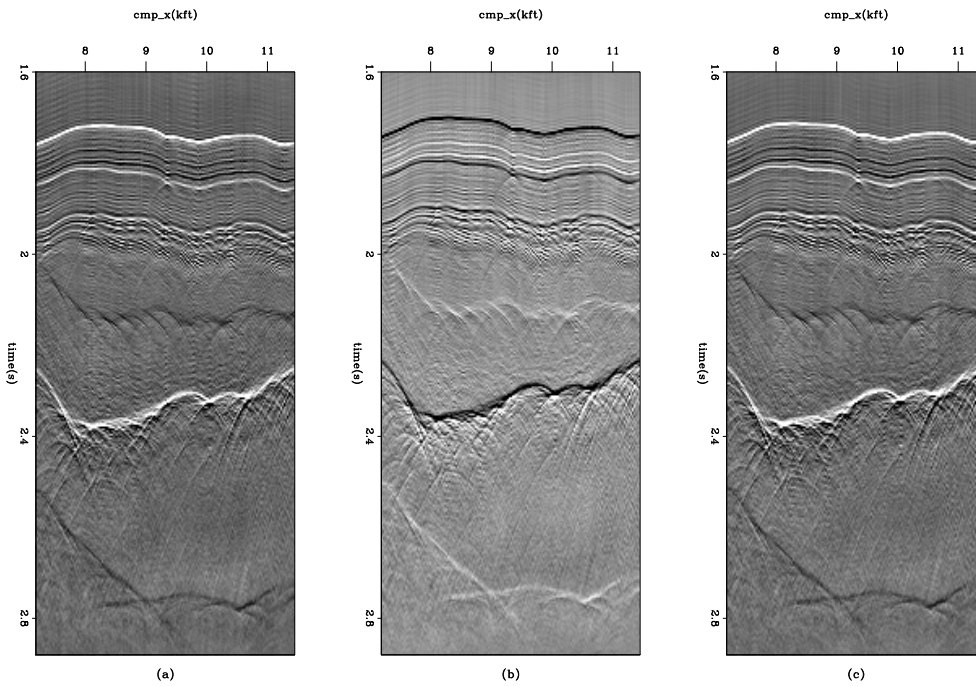


Figure 11: Given the common offset data in Figure 6, (a)bidirectional deconvolution without preconditioning; (b)bidirectional deconvolution with PEF preconditioning (c) bidirectional deconvolution with GALIP preconditioning. [ER]

small ringings around the spike, which may encourage us to use more prior knowledge on preconditioning. For field data, the results with preconditioning have less precursors, cleaner salt body, more symmetric wavelet and faster convergence rate than those without preconditioning. This proves that preconditioning, which utilizes prior information, can guide the gradient along sensible pathways thus avoiding potential local minima and make the results more reliable and converging faster.

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

ACKNOWLEDGMENTS

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

REFERENCES

- Jon Claerbout, Q. F. and Y. Shen, 2011, A log spectral approach to bidirectional deconvolution: SEP-Report, **143**, 295–298.
- Yi Shen, Q. F. 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.