Stanford Exploration Project, Report 115, May 22, 2004, pages 315–336

Conjugate-guided-gradient (CGG) method for robust inversion and its application to velocity-stack inversion

Jun Ji[1]

ABSTRACT

This paper proposes a modified conjugate-gradient (CG) method, called the conjugate-guided-gradient (CGG) method, as an alternative iterative inversion method that is robust and easily manageable. The CG method for solving least-squares (LS) (i.e. L^2-norm minimization) problems can be modified to solve for a different norm or different minimization criteria by guiding the gradient vector appropriately. The guiding can be achieved by iteratively weighting either the residual vector or the gradient vector during iteration steps. Weighting the residual vector can guide the solution to the minimum L^p-norm solution, and weighting the gradient vector can guide the solution to one constrained by *a priori* information imposed in the model space. In both cases, the minimum solutions are found in a least-squares sense along the gradient direction guided by the weights. Therefore, the solution found by the CGG method can be interpreted as the LS solution located in the guided gradient direction. I applied the CGG method to the velocity stack inversion, and the results suggest that the CGG method gives a far more robust model estimation than the standard L^2-norm solution, with results comparable to, or better than, an L^1-norm IRLS (Iteratively Reweighted LS) solution.

INTRODUCTION

The inverse problem has received considerable attention in various geophysical applications. One of the most popular inverse solutions is the least-squares (LS) solution. The LS solution is a member of a family of generalized L^p-norm solutions that are deduced from a maximum-likelihood formulation. This formulation allows the design of various statistical inversion solutions. Among the various L^p-norm solutions, the L^1-norm solution is more robust than the L^2-norm solution, being less sensitive to spiky, high-amplitude noise (Claerbout and Muir, 1973; Taylor et al., 1979; Scales and Gersztenkorn, 1987; Scales et al., 1988). However, the implementation of the algorithm to find L^1-norm solutions is not a trivial task. Iterative inversion algorithms called IRLS (Iteratively Reweighted LS) (Gersztenkorn et al., 1986; Scales et al., 1988) are a good choice for solving L^p-norm minimization problems for $1 \le p \le 2$. If the number of unknown model values is very large, LS problems are often solved by iterative solvers like the popular conjugate-gradient (CG) method. IRLS approaches for nonlinear inversion can be adapted to solve linear inverse problems by modifying the CG method (Darche,

[1]**email:** jun@sep.stanford.edu

1989; Nichols, 1994; Claerbout, 2004). This paper introduces a way to modify the CG method so that it can not only handle the general L^p-norm problem but also impose *a priori* constraints on the solution space. This method is called conjugate-guided-gradient (CGG), and is achieved by guiding the gradient vector during the iteration step. In the first section, I review the conventional CG method for solving LS problems and show how the IRLS approach differs from the standard LS approach. Next, I explain the CGG method and contrast it with both LS and IRLS. Finally I test the proposed CGG method on velocity-stack inversions with both noisy synthetic data and real data. I compare the results of the CGG method with conventional LS and L_1-norm IRLS results.

CG METHOD FOR LS AND IRLS INVERSION

Most inversion problems start by formulating the forward problem, which describes the forward operator, **L**, that transforms the model vector **m** to the data vector **d** :

$$\mathbf{d} = \mathbf{L}\mathbf{m}. \tag{1}$$

In general, the measured data **d** may be inexact, and the forward operator **L** may be ill-conditioned. In that case, instead of solving the above equation directly, different approaches are used to find an optimum solution **m** for a given data **d**. The most popular method is finding a solution that minimizes the misfit between the data **d** and the modeled data **Lm**. The misfit, or the residual vector, **r**, is described as follows:

$$\mathbf{r} = \mathbf{L}\mathbf{m} - \mathbf{d}. \tag{2}$$

In least-squares inversion, the solution **m** is the one that minimizes the squares of the residual vector as follows:

$$min_m(\mathbf{r}\cdot\mathbf{r}) = min_m(\mathbf{L}\mathbf{m} - \mathbf{d})^T(\mathbf{L}\mathbf{m} - \mathbf{d}). \tag{3}$$

Iterative solvers for the LS problem search the solution space for a better solution in each iteration step, along the gradient direction (in steepest-descent algorithms), or on the plane made by the current gradient vector and the previous descent-step vector (in conjugate-gradient algorithms). Following Claerbout (1992), a conjugate-gradient algorithm for the LS solution can be summarized as follows:

$$
\begin{array}{lll}
 & \mathbf{r} & \longleftarrow \quad \mathbf{L}\mathbf{m} - \mathbf{d} \\
\text{iterate } \{ & & \\
 & \Delta\mathbf{m} & \longleftarrow \quad \mathbf{L}^T\,\mathbf{r} \\
 & \Delta\mathbf{r} & \longleftarrow \quad \mathbf{L}\,\Delta\mathbf{m} \\
 & (\mathbf{m},\mathbf{r}) & \longleftarrow \quad \text{cgstep}(\mathbf{m},\mathbf{r},\Delta\mathbf{m},\Delta\mathbf{r}) \\
 & \} \,, &
\end{array}
$$

where the subroutine cgstep() remembers the previous iteration descent vector, $\Delta\mathbf{s} = \mathbf{m}_i - \mathbf{m}_{i-1}$, where i is the iteration step, and determines the step size by minimizing the quadrature function composed from $\Delta\mathbf{r}$ (the conjugate gradient) and $\Delta\mathbf{s}$ (the previous iteration descent vector), as follows (Claerbout, 1992):

$$Q(\alpha,\beta) = (\mathbf{r} - \alpha\Delta\mathbf{r} - \beta\Delta\mathbf{s})^T(\mathbf{r} - \alpha\Delta\mathbf{r} - \beta\Delta\mathbf{s}).$$

Notice that the gradient vector ($\Delta\mathbf{m}$) in the CG method for LS solution is the gradient of the squared residual and is determined by taking the derivative of the squared residual (i.e. the L^2-norm of the residual, $\mathbf{r}\cdot\mathbf{r}$, with respect to the model $\mathbf{m}^T$):

$$\Delta\mathbf{m} = \frac{\partial}{\partial\mathbf{m}^T}(\mathbf{Lm}-\mathbf{d})^T(\mathbf{Lm}-\mathbf{d}) = \mathbf{L}^T\mathbf{r}. \tag{4}$$

Iteratively Reweighted Least Squares (IRLS)

Instead of L^2-norm solutions obtained by the conventional LS solution, L^p-norm minimization solutions, with $1 \le p \le 2$, are often tried. Iterative inversion algorithms called IRLS (Iteratively Reweighted Least Squares) algorithms have been developed to solve these problems, which lie between the least-absolute-values problem and the classical least-squares problem. The main advantage of IRLS is to provide an easy way to compute the approximate L^1-norm solution. L^1-norm solutions are known to be more robust than L^2-norm solutions, being less sensitive to spiky, high-amplitude noise (Claerbout and Muir, 1973; Taylor et al., 1979; Scales and Gersztenkorn, 1987; Scales et al., 1988). The problem solved by IRLS is a minimization of the weighted residual in the least-squares sense :

$$\mathbf{r} = \mathbf{W}(\mathbf{Lm}-\mathbf{d}). \tag{5}$$

The gradient for the weighted residual in the least-squares sense becomes

$$\mathbf{L}^T\mathbf{W}\mathbf{r} = \frac{\partial}{\partial\mathbf{m}^T}(\mathbf{Lm}-\mathbf{d})^T\mathbf{W}^T\mathbf{W}(\mathbf{Lm}-\mathbf{d}). \tag{6}$$

The particular choice for $\mathbf{W}$ is the one that results in minimizing the L^p norm of the residual. Choosing the i^{th} diagonal element of $\mathbf{W}$ to be a function of the i^{th} component of the residual vector as follows:

$$\mathrm{diag}(\mathbf{W}) = |\mathbf{r}|^{(p-2)/2}, \tag{7}$$

the norm of the weighted residual is then

$$\mathbf{r}^T\mathbf{W}^T\mathbf{W}\mathbf{r} = \mathbf{r}^T|\mathbf{r}|^{(p-2)}\mathbf{r} = |\mathbf{r}|^p. \tag{8}$$

Therefore, this can be thought of as a method that estimates the gradient in the L^p-norm of the residual. This method is valid for norms where $1 \le p \le 2$. When the L^1-norm is desired, the weighting is as follows:

$$\mathrm{diag}(\mathbf{W}) = |\mathbf{r}|^{-1/2}.$$

This will reduce the contribution of large residuals and improve the fit to the data that is already well-estimated. Thus, the L^1-norm-based minimization is robust, less sensitive to noise bursts in the data. Huber proposed a hybrid L^1/L^2-norm (Huber, 1973) that treats the small residuals in an L^2-norm sense and the large residuals in an L^1-norm sense. This approach deals with both bursty and Gaussian-type noise, and can be realized by weighting as follows:

$$\mathrm{diag}(\mathbf{W}) = \begin{cases} |\mathbf{r}|^{-1/2}, & |\mathbf{r}| > \epsilon \\ 1, & |\mathbf{r}| \le \epsilon \end{cases}$$

where ϵ is a value that is used as a threshold between L^1 and L^2 -norms. IRLS can be easily incorporated in CG algorithms by including a weight $\mathbf{W}$ such that the operator $\mathbf{L}$ has a post-multiplier $\mathbf{W}$ and the the adjoint operator $\mathbf{L}^T$ has a premultiplier $\mathbf{W}^T$ (Claerbout, 2004). Even though we do not know the real L^p-norm residual vector at the beginning of the iteration, we can approximate the residual with a residual of the previous iteration step, and it will converge to a residual that is very close to the L^p-norm residual as the iteration step continues. This can be summarized as follows:

$$\begin{aligned} & \mathbf{r} \longleftarrow \mathbf{Lm} - \mathbf{d} \\ \text{iterate } \{ & \\ & \mathbf{W} \longleftarrow \mathbf{diag}[f(\mathbf{r})] \\ & \Delta\mathbf{m} \longleftarrow \mathbf{L}^T\mathbf{W}^T\,\mathbf{r} \\ & \Delta\mathbf{r} \longleftarrow \mathbf{WL}\,\Delta\mathbf{m} \\ & (\mathbf{m}, \mathbf{r}) \longleftarrow \mathrm{cgstep}(\mathbf{m}, \mathbf{r}, \Delta\mathbf{m}, \Delta\mathbf{r}) \\ & \}. \end{aligned}$$

CONJUGATE GUIDED GRADIENT(CGG) METHOD

Whithin the CG method, the IRLS algorithm can be considered as the LS method, but with its operator, $\mathbf{L}$, modified by the weight, $\mathbf{W}$. The only change that distinguishes the IRLS algorithm from the LS one is the substitution of $\mathbf{LW}$ and $\mathbf{W}^T\mathbf{L}^T$ for $\mathbf{L}$ and $\mathbf{L}^T$, respectively. Instead of modifying the operator, we can choose a way to guide the minimizing search to find the minimum $\mathbf{L}^2$-norm in a specific model subspace so as to obtain a solution that meets a user's specific criteria. The specific model subspace could be guided by a specific $\mathbf{L}^p$-norm's gradient or constrained by an *a priori* model. Such guiding of the model vector can be realized by weighting the residual vector or gradient vector in the CG algorithm.

CGG with iteratively reweighted residual

If we apply the same weight $\mathbf{W}$ we used in the IRLS, but do not change the operator from $\mathbf{L}$ to $\mathbf{WL}$, the weight affects only the gradient direction. This corresponds to guiding the gradient direction with a weighted residual, and the resultant weighted gradient will be the same gradient as we used in the IRLS method. This algorithm can be implemented as follows:

$$\begin{aligned} & \mathbf{r} \longleftarrow \mathbf{Lm} - \mathbf{d} \\ \text{iterate } \{ & \end{aligned}$$

$$
\begin{array}{lcl}
\mathbf{W} & \longleftarrow & \mathbf{diag}[f(\mathbf{r})] \\
\Delta\mathbf{m} & \longleftarrow & \mathbf{L}^T \mathbf{W}^T\ \mathbf{r} \\
\Delta\mathbf{r} & \longleftarrow & \mathbf{L}\ \Delta\mathbf{m} \\
(\mathbf{m},\mathbf{r}) & \longleftarrow & \text{cgstep}(\mathbf{m},\mathbf{r},\Delta\mathbf{m},\Delta\mathbf{r}) \\
\}.
\end{array}
$$

Notice that the above algorithm is different from the original CG algorithm only at the step of gradient computation; the modification of the gradient is performed by changing the residual before the gradient is computed from it. By choosing the weight as a function of the residual of the previous iteration step, as we did in the IRLS, we can guide the gradient to the gradient of the L^p-norm. Thus the result obtained by weighting the residual could be interpreted as an LS solution located along the direction of the L^p-norm gradient, according to the weight applied. If, during the iteration, any intermediate solution is found at the minimum L^2-norm location in the model space, it will be the final solution of the algorithm, and it is the same as the solution of the conventional LS problem. However, the minimum L^2-norm location is unlikely to fall along the gradient of the different L^p-norm determined by the applied weight. Therefore, it is more likely that the solution will be close to the minimum L^p-norm location determined by the applied weight.

CGG with iteratively reweighted gradient

Another way to modify the gradient direction is to modify the gradient vector after the gradient is computed from a given residual. Since the gradient vector is in the model space, any modification of the gradient vector imposes some constraint in the model space. If we know some characteristics of the solution which can be expressed in terms of weighting in the solution space, we can use that *a priori* knowledge to redirect the gradient vector by applying a weight to it. This algorithm can be implemented as follows:

$$
\begin{array}{llcl}
 & \mathbf{r} & \longleftarrow & \mathbf{Lm}-\mathbf{d} \\
\text{iterate } \{ & & & \\
 & \mathbf{W} & \longleftarrow & \mathbf{diag}[f(\mathbf{m})] \\
 & \Delta\mathbf{m} & \longleftarrow & \mathbf{W}\mathbf{L}^T\ \mathbf{r} \\
 & \Delta\mathbf{r} & \longleftarrow & \mathbf{L}\ \Delta\mathbf{m} \\
 & (\mathbf{m},\mathbf{r}) & \longleftarrow & \text{cgstep}(\mathbf{m},\mathbf{r},\Delta\mathbf{m},\Delta\mathbf{r}) \\
 & \}\ . & &
\end{array}
$$

Even though weighting the gradient has different meaning from weighting the residual, the analysis is similar in both cases. As we redefined the contribution of each residual element by weighting it with the absolute value of itself to some power: we can do the same with each model element in the solution,

$$\mathbf{W} = |\mathbf{m}|^p, \tag{9}$$

where p is a real number that depends on the problem we wish to. When we have a finite model space we are applying a uniform weight to the finite model space and zero weight to the outlying space. If the operator used in the inversion is close to unitary, the solution obtained after the first iteration already closely approximates the real solution. Therefore, weighting the gradient with some power of the absolute value of the previous iteration means that we down-weight the importance of small model values and improve the fit to the data by emphasizing model components that already have large values.

CGG with iteratively reweighted residual and gradient

In the previous two subsections, we examined the meaning the of weighting the residual and the gradient vector, respectively. Since applying the weighting in both residual space and model space is nothing but changing the direction of the descent for the solution search, the weighting is not limited either to residual or to model space. We can weight both the residual and the gradient,

$$\begin{array}{lll} & \mathbf{r} & \longleftarrow \quad \mathbf{Lm}-\mathbf{d} \\ \text{iterate } \{ & & \\ & \mathbf{W}_r & \longleftarrow \quad \mathbf{diag}[f(\mathbf{r})] \\ & \mathbf{W}_m & \longleftarrow \quad \mathbf{diag}[f(\mathbf{m})] \\ & \Delta\mathbf{m} & \longleftarrow \quad \mathbf{W}_m\mathbf{L}^T\mathbf{W}_r\mathbf{r} \\ & \Delta\mathbf{r} & \longleftarrow \quad \mathbf{L}\,\Delta\mathbf{m} \\ & (\mathbf{m},\mathbf{r}) & \longleftarrow \quad \text{cgstep}(\mathbf{m},\mathbf{r},\Delta\mathbf{m},\Delta\mathbf{r}) \\ & \} \,. & \end{array}$$

Again, the above CGG algorithm is different from the conventional CG method only in the step of gradient computation. Whether we modify the gradient in the residual sense or in the model sense, it changes only the gradient direction, or the direction in which the solution is sought. Therefore the CGG algorithm always converges to a solution.

APPLICATION OF THE CGG METHOD IN VELOCITY-STACK INVERSION

In this section, the CGG method is tested on a velocity-stack inversion, which is useful not only for velocity analysis but also for various data processing. The applications of the velocity-stack inversion include separating multiples from the signal in the velocity domain (Lumley et al., 1995; Kostov and Nichols, 1995), multiple-attenuation techniques using parabolic Radon transforms (Kabir and Marfurt, 1999; Herrmann et al., 2000), missing-trace interpolation in the CMP domain (Ji, 1994), and so on. In these applications, the velocity-stack panels obtained by inversion are usually required to be as spiky and sparse as possible. Then the hyperbolic events represented by the isolated peaks in the velocity-stack panel are more easily distinguished from the rest of the noise. The conventional velocity stack is performed by summing or estimating semblance (Taner and Koehler, 1969) along the various hyperbolas in a CMP gather, resulting in a velocity-stack panel. Ideally a hyperbola in a CMP gather should be mapped onto a point in a velocity-stack panel. Summation along a hyperbola, or hyperbolic

Radon transform (HRT), does not give such resolution. To obtain a velocity-stack panel with better resolution, Thorson and Claerbout (1985) formulated it as an inverse problem in which the velocity domain is the unknown space. If we find an operator $\mathbf{H}$ that transforms a point in a model space (velocity-stack panel), $\mathbf{m}$, into a hyperbola in data space (CMP gather), $\mathbf{d}$,

$$\mathbf{d} = \mathbf{H}\mathbf{m}, \tag{10}$$

and also find its adjoint operator $\mathbf{H}^T$, we can pose the velocity stack problem as an inverse problem. Inverse theory helps us to find a velocity-stack panel which synthesizes a given CMP gather via the operator $\mathbf{H}$. The usual process is to implement the inverse as the minimization of a least-squares problem and calculate the solution by solving the normal equation:

$$\mathbf{H}^T\mathbf{H}\mathbf{m} = \mathbf{H}^T\mathbf{d}. \tag{11}$$

Since the number of equations and unknowns may be large, an iterative least-squares solver such as CG is usually preferred to solving the normal equation directly. The least-squares solution has some attributes that may be undesirable. If the model space is overdetermined and has bursty noise in data, the least-squares solutions usually will be spread over all the possible solutions. Other methods may be more useful if we desire a parsimonious representation. To obtain a more robust solution, Nichols (1994) used the IRLS method for L^1-norm minimization, and Guitton and Symes (2003) used the L-BFGS method for Huber-norm minimization. Another possibility is the CGG method proposed in the preceding section. In the next subsections the results of the CGG method for the velocity-stack inversion are compared with the results of conventional LS and L^1-norm IRLS.

Examples on synthetic data

To examine the performance of the proposed CGG method, a synthetic CMP data set with various types of noise is used. Figure 1 shows the synthetic data with three types of noise — Gaussian noise in the background, bursty noise, and a very noisy trace. Figure 1 (right) is the same data as Figure 1 (left), but displayed in wiggle format to clearly show the bursty noise that was not discernable because of the clipping in the raster-format display. The amplitudes of the three bursty spikes are eight times the maximum amplitude of the hyperbolas. Figure 2(a) shows the modeled data from the velocity-stack panel obtained using the conventional CG algorithm for LS solution. The inversion was obtained with 30 iterations and the same number of iteration was used for all the other examples including the real data cases. We can clearly see the limit of L^2-norm minimization. The noise with Gaussian statistics is removed quite well, but some spurious events are generated around the bursty noise spikes and noisy trace. Figure 2(b) shows the modeled data from the velocity-stack panel obtained using the IRLS algorithm in an L^1-norm sense. In Figure 2(b) we can see the robustness of L^1-norm minimization. The nursty noise is reduced significantly, but the removal of the background noise seems to be worse than the result of L^2-norm minization. Figure 2(c) shows the modeled data from the velocity-stack panel obtained using the CGG method, with the iteratively reweighted residual in an L^1-norm sense. The result is comparable to the IRLS inversion (Figure 2(b)). This tells us that guiding the gradient vector toward the L^1-norm gradient gives a solution similar to the L^1-norm solution with the IRLS method. Figure 2(d) shows the modeled data from

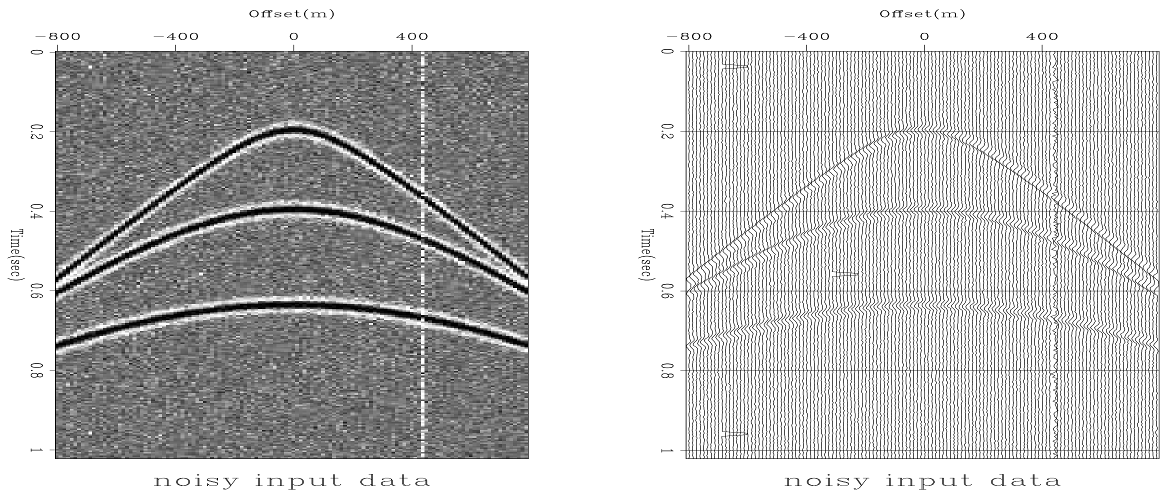


Figure 1: Synthetic input data with various noise types in raster format (left) and wiggle format (right). jun1-fig1 [ER]

the velocity-stack panel obtained using the CGG method with iteratively reweighted gradient as follows:

$$\mathrm{diag}(\mathbf{W}_i) = |\mathbf{m}_{i-1}|^{1.5},$$

that is the diagonal element of the i-th iteration weighting matrix is the absolute value of the model vector from the previous iteration, raised to the power 1.5. The result shows that the Gaussian background noise and the very noisy trace are better removed than with any of the L^1-norm approaches, either in CGG or IRLS. However, some spurious events around the bursty spikes still exist. This is because of the high amplitude of the noise. Since the weight is dependent on the amplitude of the model vector, and because the high amplitude in the CMP gather is also mapped to a high amplitude in the velocity-stack panel, the bursty noise with high amplitude would have higher weighting than the noise with low amplitude. However, this kind of artifact would easily be removed if the bursty noise had an amplitude similar to the rest of signal, which is the case when AGC (automatic gain control) is applied to the data. Figure 3 shows the modeled data obtained using the CGG algorithm with both residual and gradient weightings. In this case, the result looks like modeled data without any noise, because the bursty noise is reduced with residual weighting (L^1-norm criteria), and background noise is removed with gradient weighting. The velocity stacks obtained from the various inversions are shown in Figure 4. From left to right, the velocity-stack panels correspond to the results from LS inversion, IRLS inversion, CGG with residual weighting, CGG with gradient weighting, and CGG with both residual and gradient weighting. From these velocity-stack panels, we can deduce why different inversion methods were successful with the different noise styles. If we want an application that distinguishes signals in the model space, we can see that gradient weighting is the preferred method, because it gives a more parsimonious representation than the others.

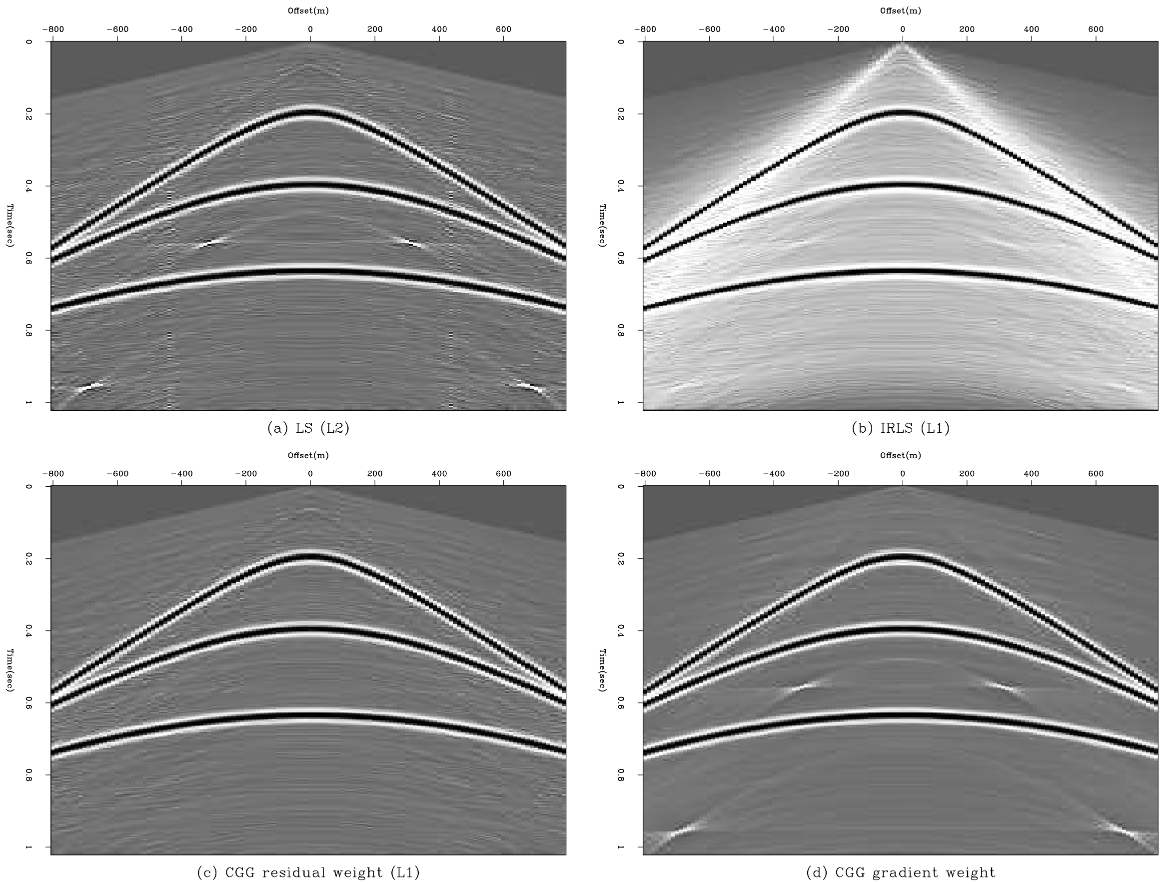


Figure 2: Remodeled data from the result of the LS inversion (a), from the result of the IRLS inversion with an L^1-norm minimization (b), from the CGG method with iteratively reweighted residuals (c) in an L^1-norm sense, and from the result of the CGG method with iteratively reweighted gradients (d). jun1-fig2 [ER]

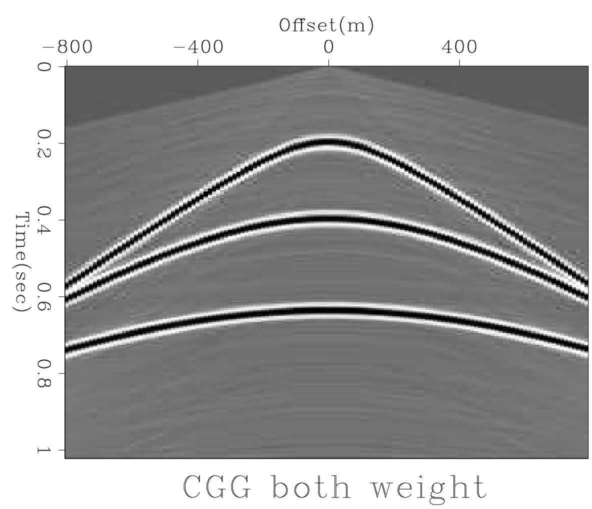


Figure 3: Remodeled data from the CGG method, with iteratively reweighted residual and gradient together. jun1-fig3 [ER]

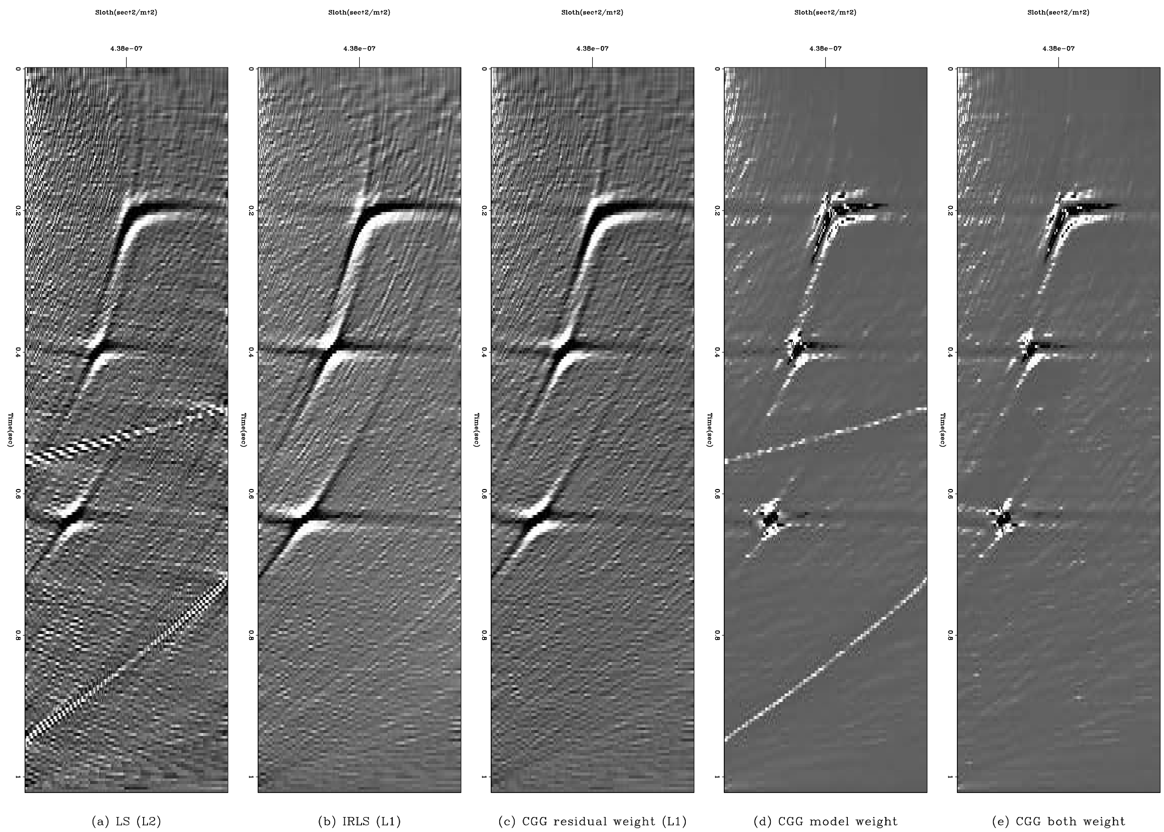


Figure 4: Velocity-stack panels obtained by various inversions. From left to right: LS inversion, IRLS, CGG with residual weighting, CGG with gradient weighting, and CGG with both residual and gradient weighting. jun1-fig3v [ER]

Examples on real data

I tested the proposed CGG method on two real data sets that contain various types of noise. The data are shot gathers from land surveys. The trajectories of the events from both data sets look "hyperbolic" enough to be tested with hyperbolic inversion. Figure 5 shows one of the real data sets tested. The noise in the data is mainly the strong ground roll, the amplitude anomalies early at near-offset and late at 0.8km offset, and the time shift around offsets 1.6 km and 2.0 km. Figure 6(a) shows the modeled data obtained using the IRLS algorithm in an L^1-norm sense. Figure 6(b) shows the modeled data obtained using the CGG algorithm with residual weighting in an L^1-norm sense. In both panels in Figure 6, the noise is greatly reduced, especially the ground roll, since the limited-size model space does not include the ground-roll velocity. However, we can see that the IRLS method reduce the signal at far offset too much and the signal is preserved better by CGG with residual weighting (Figure 6(b)) than with IRLS inversion (Figure 6(a)). Figure 7(a) shows the modeled data from the result obtained using the CGG algorithm with gradient weighting. The following weighting factor was used:

$$\mathrm{diag}(\mathbf{W}_i) = |\mathbf{m}_{i-1}|^{1.5},$$

that is the diagonal element of the i-th iteration weighting matrix is the absolute value of the (i-1)-th iteration model vector raised to the power 1.5. Comparing this with Figure 6(b) shows that weighting the gradient directly could result in better noise reduction at far offset than CGG with residual weighting. Figure 7(b) shows the modeled data obtained using the CGG algorithm with both residual and gradient weightings. In this case, the result is very similar to that of gradient-only weighting (Figure 7(a)), and it tells us that most of the noise could removed by weighting only the gradient. The velocity stacks obtained from the various inversions are shown in Figure 8. As we saw in the synthetic data example, the velocity stacks obtained by the CGG method that includes gradient weighting, Figure 8 (c) and (d), show more parsimonious representation than the others. Figure 9 shows another real data set used for testing. The noise in the data can be characterized by anomalous shifts and noisy amplitude at near offset around 1.5 sec, several junk traces around middle offset, and wide-spread noise. Figure 10(a) shows the modeled data obtained using the IRLS algorithm in an L^1-norm sense. Figure 10(b) shows the modeled data obtained using the CGG algorithm with residual weighting in an L^1-norm sense. In both results, most of the noise is greatly reduced. However, we can see that some of the noise characterized by anomalous amplitude is better reduced by CGG with residual weighting (Figure 10(b)) than by IRLS inversion (Figure 10(a)) and the IRLS inversion shows overly reduced amplitude at far offset signal, again. We can see that for the same iteration numbers, residual-only weighting is more effective than IRLS style residual weighting. Figure 11(a) shows the modeled data obtained using the CGG algorithm with gradient weighting. The following weighting factor was used:

$$\text{diag}(\mathbf{W}_i) = |\mathbf{m}_{i-1}|^2,$$

that is the diagonal element of the i-th iteration weighting matrix is the square of the (i-1)-th iteration model vector. Comparing this with Figure 10(b) shows the interesting result that weighting the gradient directly could result in noise reduction similar to that of residual weighting. Figure 11(b) shows the modeled data obtained using the CGG algorithm with both residual and gradient weightings. In this case, the result differs little from that of gradient-only weighting (Figure 11(a)), and most of the noise could removed by weighting only the gradient. The velocity stacks obtained from the various inversions are shown in Figure 12. As we saw in the synthetic data example, the velocity stacks obtained by the CGG method with gradient weighting show more parsimonious representation than the others.

CONCLUSIONS

The proposed CGG (Conjugate Guided Gradient) inversion method is a modified CG (Conjugate Gradient) inversion method, which guides the gradient vector during the iteration and allows the user to impose various constraints for residual and model. The guiding is implemented by weighting the residual vector and the gradient vector, either separately or together. Weighting the residual vector with the residual itself corresponds to guiding the solution search toward the L^p-norm minimization; weighting the gradient vector with the model itself corresponds to guiding the solution search toward *a priori* information imposed. Testing the CGG algorithm for the velocity-stack inversion of noisy synthetic and real data demonstrates that

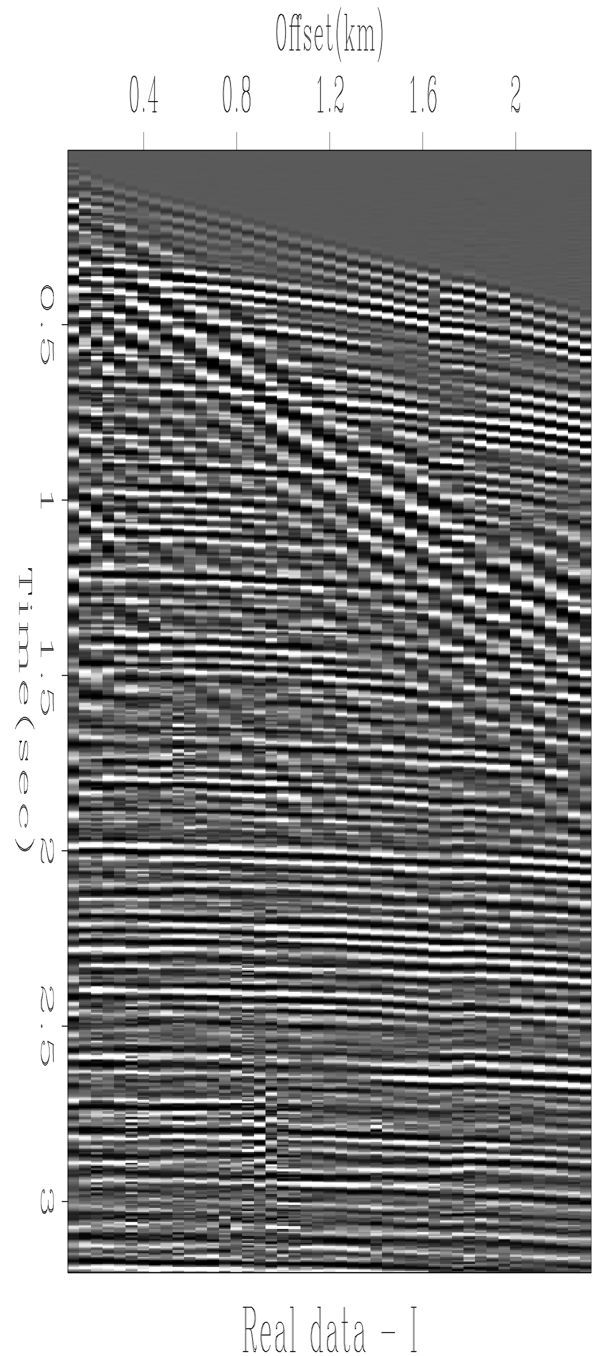


Figure 5: The real data used for the inversion. Notice the strong ground roll, the amplitude anomalies early at near-offset and late at 0.8km offset, and the time shift around offsets 1.6 km and 2.0 km. jun1-fig4-0 [CR]

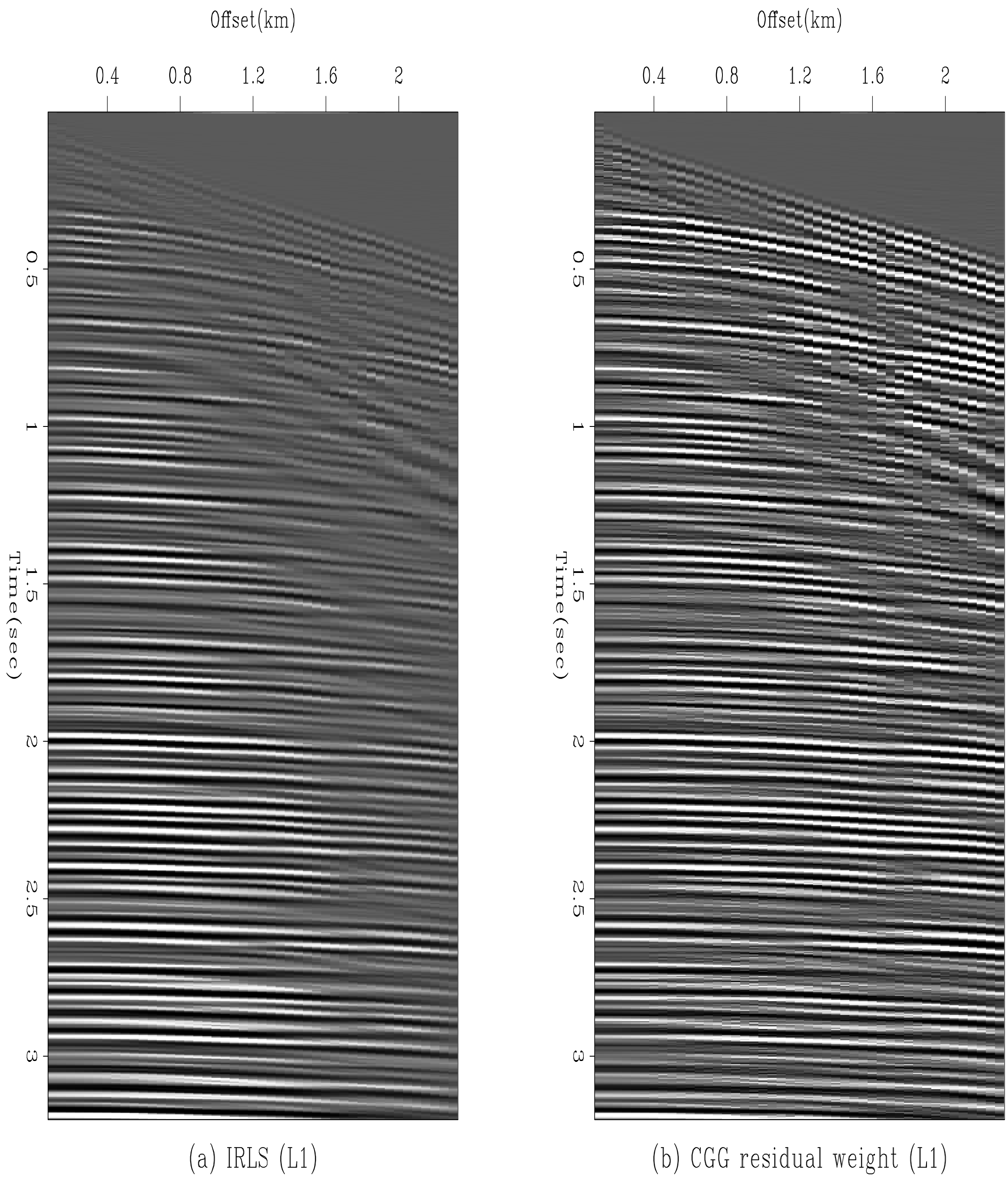
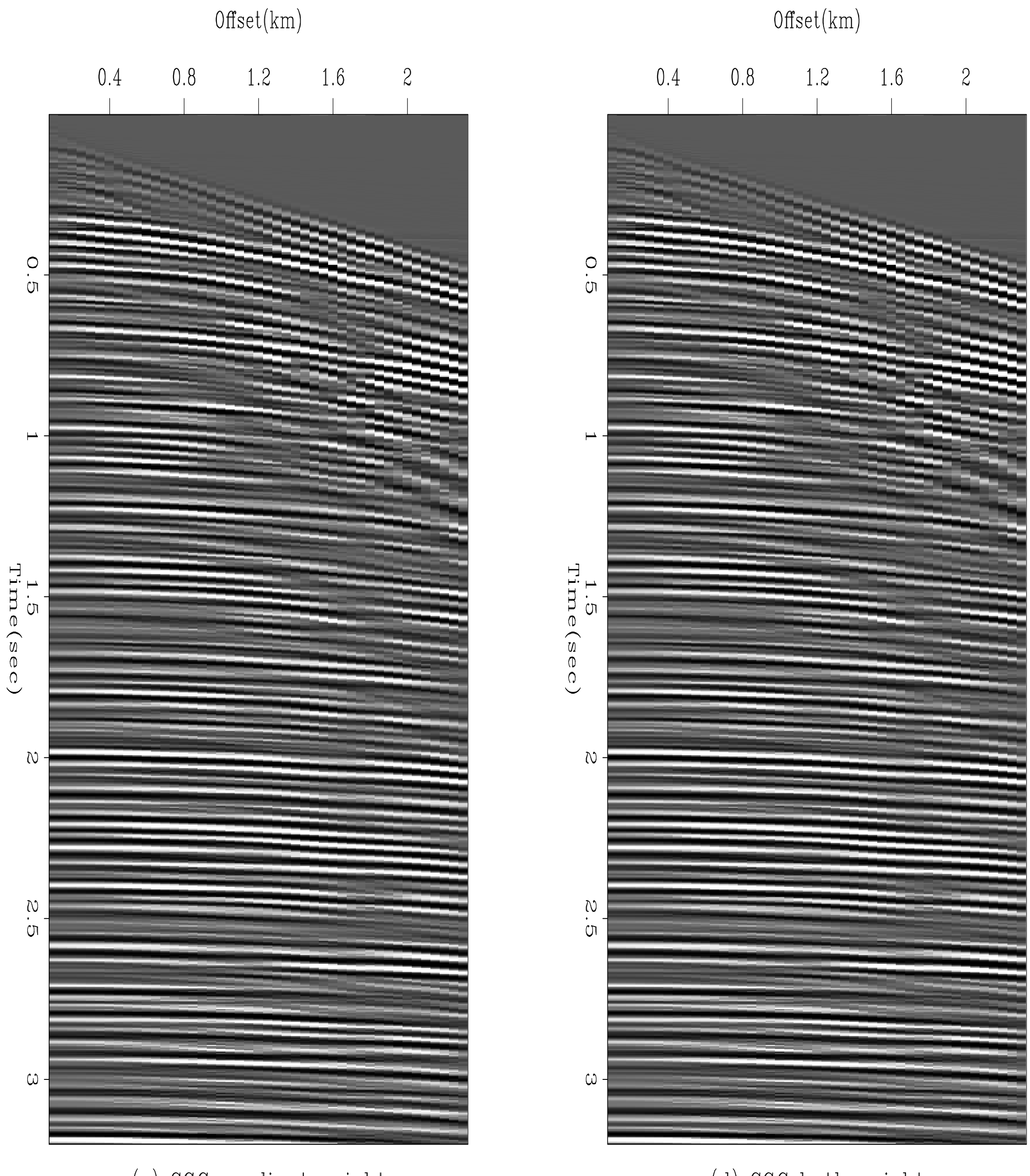


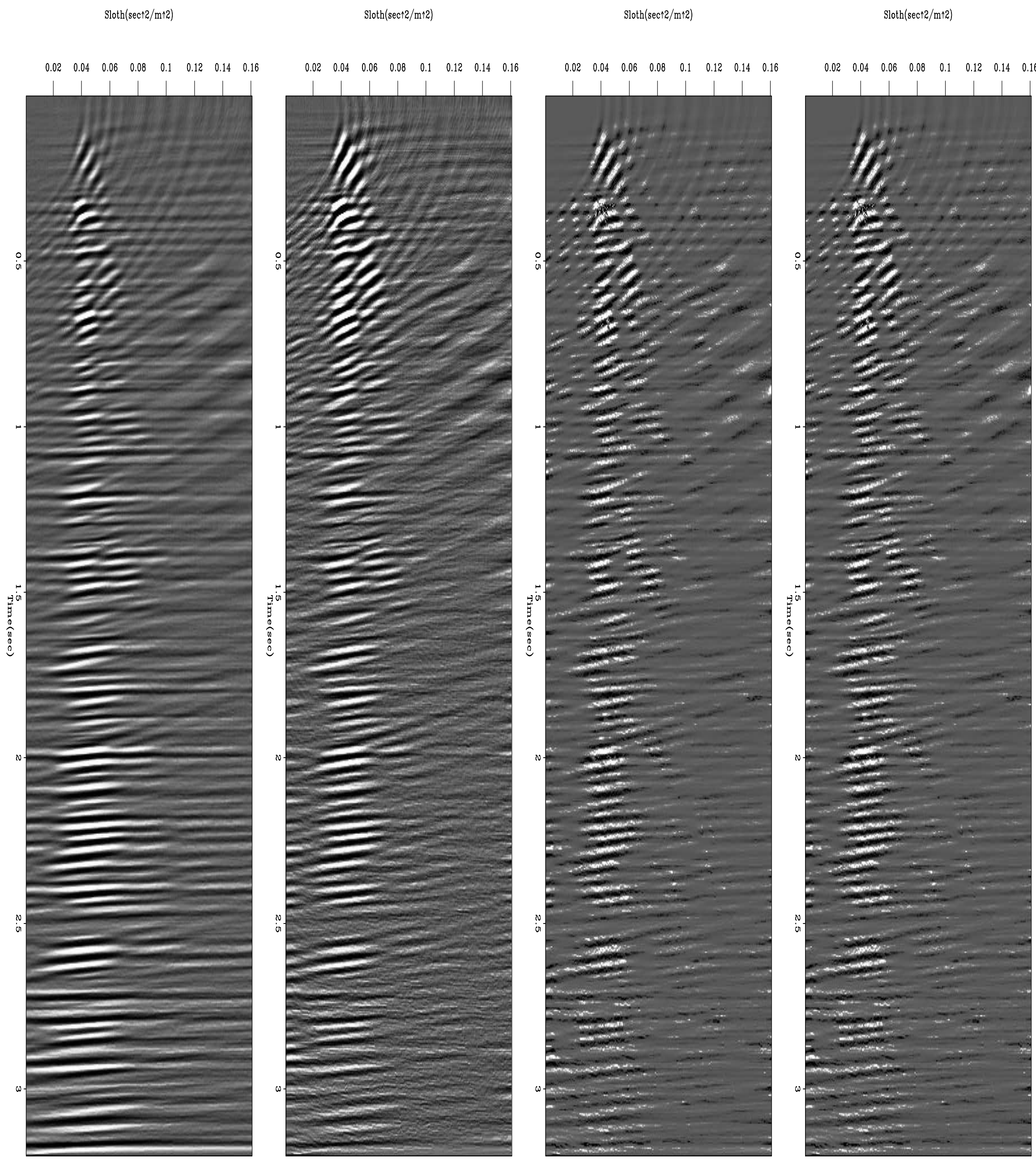
Figure 6: Remodeled data from the IRLS inversion with L^1-norm minimization (a) and from the result of the CGG method with iteratively reweighted residual in an L^1-norm sense (b). jun1-fig4-1 [CR]



(c) CGG gradient weight

(d) CGG both weight

Figure 7: Remodeled data from the CGG method with iteratively reweighted gradient (c) and from the CGG method with iteratively reweighted residual and gradient together (d). jun1-fig4-2 [CR]



(a) IRLS (L1) (b) CGG residual weight (L1) (c) CGG model weight (d) CGG both weight

Figure 8: Velocity-stack panels obtained by various inversions. From left to right: IRLS, CGG with residual weighting, CGG with gradient weighting, and CGG with both residual and gradient weighting. jun1-fig4v [CR]

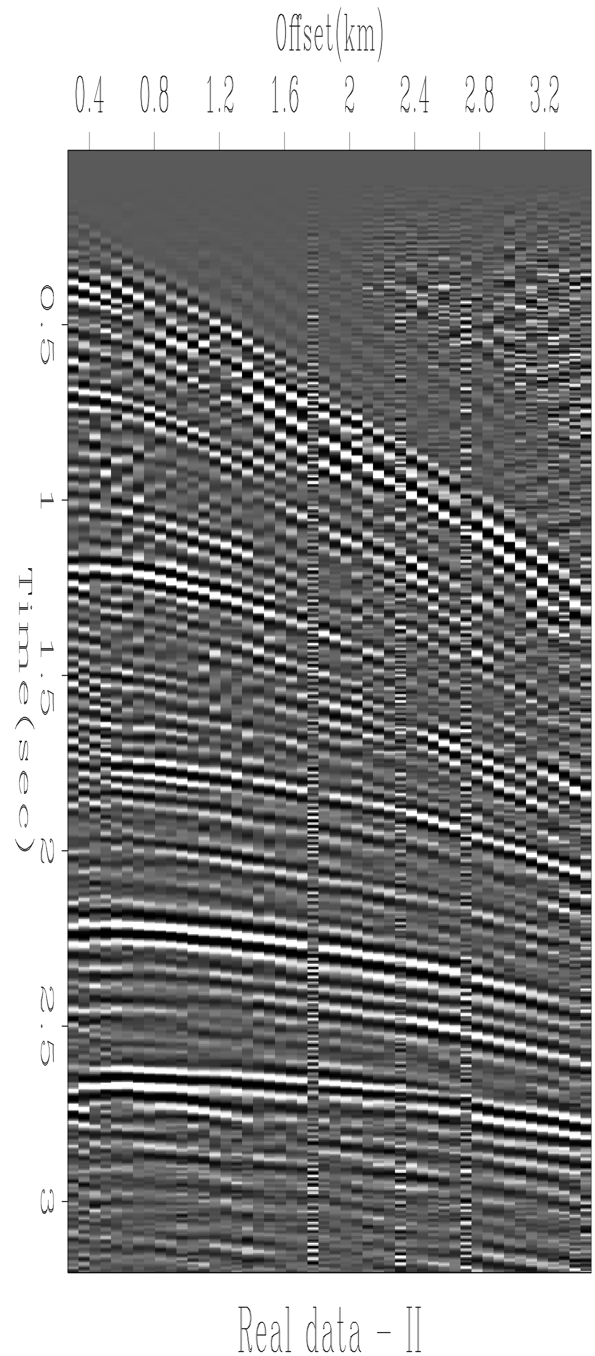


Figure 9: The real data used for the inversion. Notice the amplitude anomalies and time shift at near-offset traces, unrealistic junk traces at 1.8 km and 2.7 km, and widespread noise. jun1-fig5-0 [CR]

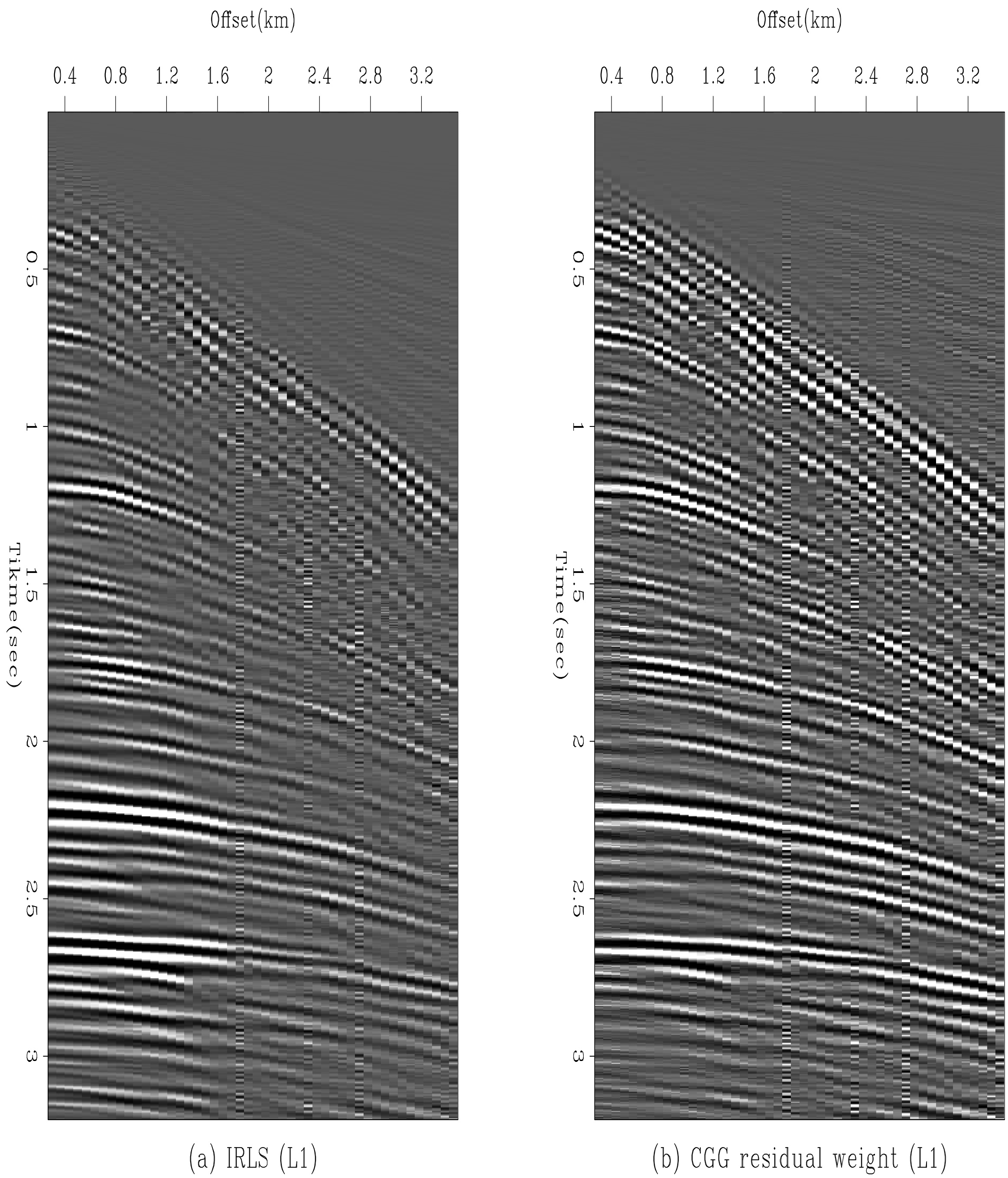


Figure 10: Remodeled data from the IRLS inversion with L^1-norm minimization (a), and remodeled data from the CGG method with iteratively reweighted residual in the L^1-norm sense (b). jun1-fig5-1 [CR]

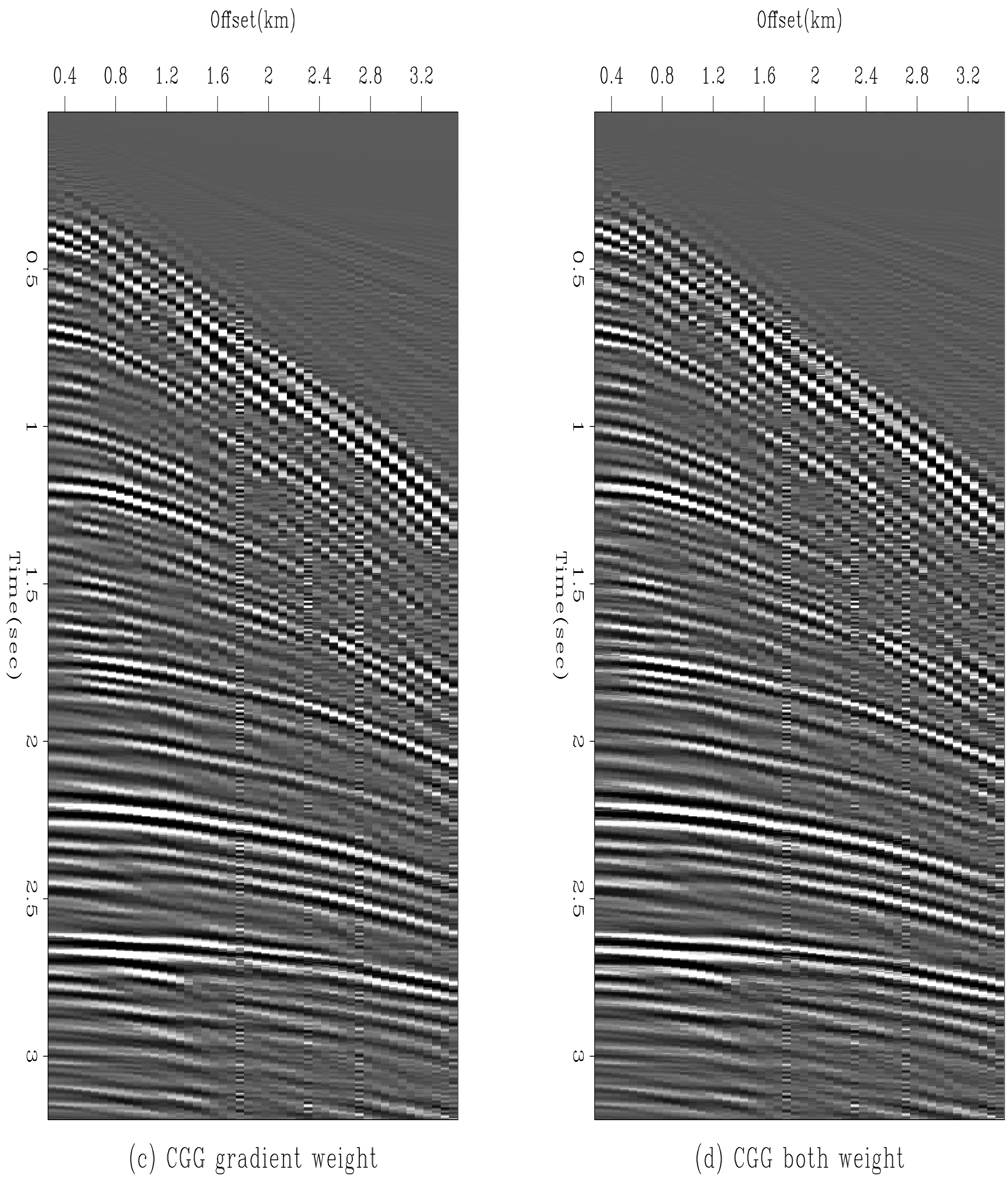


Figure 11: Remodeled data from the CGG method with iteratively reweighted gradient (c), and from the CGG method with iteratively reweighted residual and gradient together (d). jun1-fig5-2 [CR]

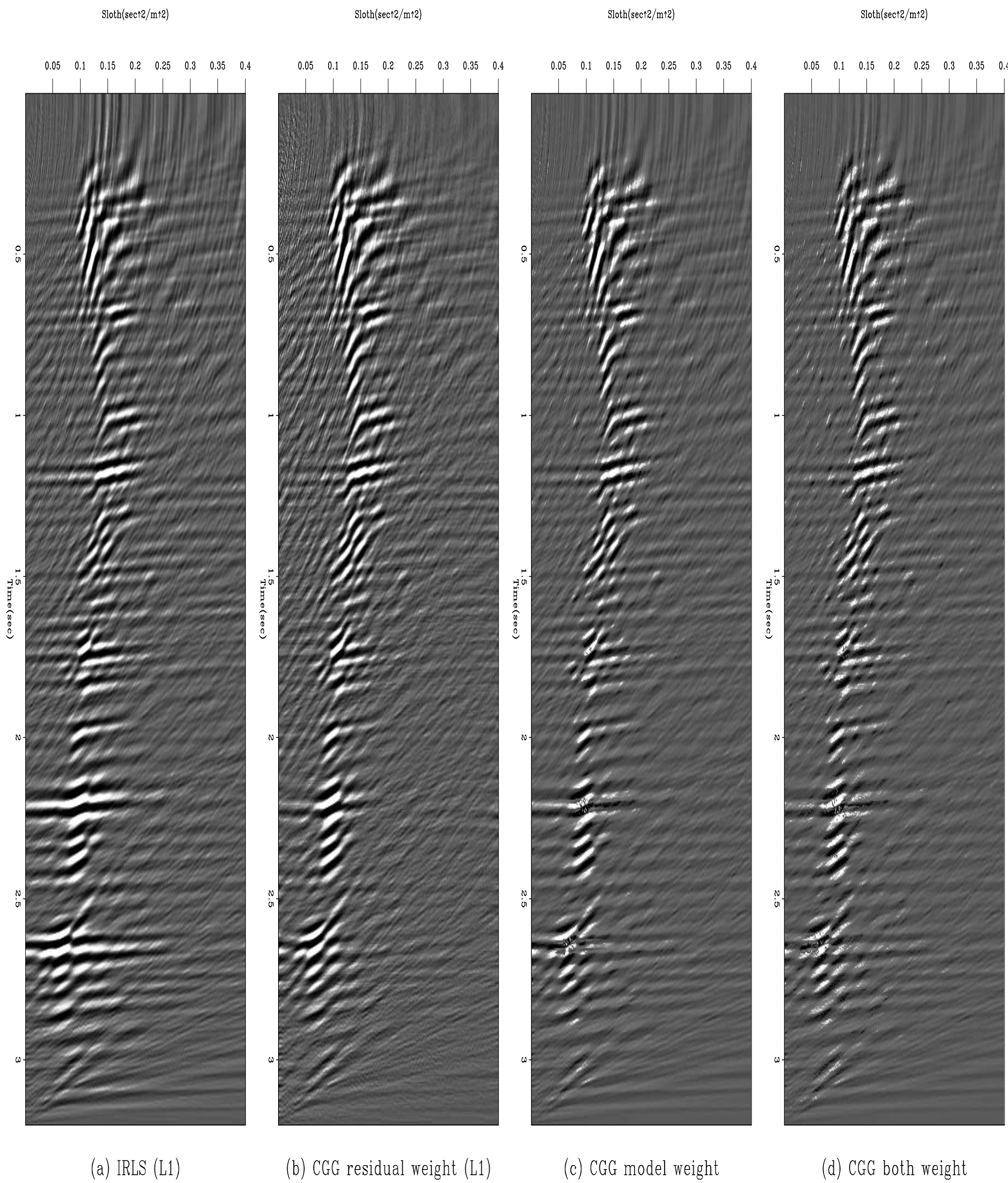


Figure 12: Velocity-stack panels obtained by various inversions. From left to right: IRLS, CGG with residual weighting, CGG with gradient weighting, and CGG with both residual and gradient weighting. jun1-fig5v [CR]

residual weighting appears to be comparable to or better than IRLS for the L^1-norm solution. Gradient weighting produces a more spiky velocity spectrum than any of the L^p-norm solutions, which are preferable for velocity picking. Therefore, I think the CGG method is a possible alternative to the more traditional IRLS method for robust inversion of seismic data.

REFERENCES

Claerbout, J. F., and Muir, F., 1973, Robust modeling with erratic data: Geophysics, **38**, no. 5, 826–844.

Claerbout, J. F., 1992, Earth Soundings Analysis, Processing versus Inversion: Blackwell Scientific Publication.

Claerbout, J. F., 2004, Image Estimation by Example: `http://sepwww.stanford.edu/sep/prof/index.html`.

Darche, G., 1989, Iterative l_1 deconvolution: SEP–**61**, 281–302.

Gersztenkorn, A., Bednar, J. B., and Lines, L. R., 1986, Robust iterative inversion for the one-dimensional acoustic wave equation: Geophysics, **51**, no. 02, 357–368.

Guitton, A., and Symes, W., 2003, Robust inversion of seismic data using the Huber norm: Geophysics, **68**, no. 4, 1310–1319.

Herrmann, P., Mojesky, T., Magesan, M., and Hugonnet, P., 2000, De-aliased, high-resolution Radon transforms: Soc. of Expl. Geophys., 70th Ann. Internat. Mtg, 1953–1956.

Huber, P. J., 1973, Robust regression: Asymptotics, conjectures, and Monte Carlo: Ann. Statist., **1**, 799–821.

Ji, J., 1994, Near-offset interpolation in wavefront synthesis imaging: SEP–**82**, 195–208.

Kabir, M. M. N., and Marfurt, K. J., 1999, Toward true amplitude multiple removal: The Leading Edge, **18**, no. 1, 66–73.

Kostov, C., and Nichols, D., 1995, Moveout-discriminating adaptive subtraction of multiples: Soc. of Expl. Geophys., 65th Ann. Internat. Mtg, 1464–1467.

Lumley, D. E., Nichols, D., and Rekdal, T., 1995, Amplitude-preserved multiple suppression: Soc. of Expl. Geophys., 65th Ann. Internat. Mtg, 1460–1463.

Nichols, D., 1994, Velocity-stack inversion using $\mathbf{L_p}$ norms: SEP–**82**, 1–16.

Scales, J. A., and Gersztenkorn, A., 1987, Robust methods in inverse theory *in* Scales, J. A., Ed., Geophysical imaging, symposium of geophysical society of Tulsa:: Soc. of Expl. Geophys., 25–50.

Scales, J. A., Gersztenkorn, A., Treitel, S., and Lines, L. R., 1988, Robust optimization methods in geophysical inverse theory: Soc. of Expl. Geophys., 58th Ann. Internat. Mtg, Session:S7.1.

Taner, M. T., and Koehler, F., 1969, Velocity spectra - Digital computer derivation and applications of velocity functions: Geophysics, **34**, no. 06, 859–881.

Taylor, H. L., Banks, S. C., and McCoy, J. F., 1979, Deconvolution with the L-one norm: Geophysics, **44**, no. 01, 39–52.

Thorson, J. R., and Claerbout, J. F., 1985, Velocity stack and slant stochastic inversion: Geophysics, **50**, no. 12, 2727–2741.