GEOPHYSICS, VOL. 68, NO. 4 (JULY-AUGUST 2003); P. 1310–1319, 10 FIGS.
10.1190/1.1598124

Robust inversion of seismic data using the Huber norm

Antoine Guitton* and William W. Symes‡

ABSTRACT

The "Huber function" (or "Huber norm") is one of several robust error measures which interpolates between smooth (ℓ^2) treatment of small residuals and robust (ℓ^1) treatment of large residuals. Since the Huber function is differentiable, it may be minimized reliably with a standard gradient-based optimizer. We propose to minimize the Huber function with a quasi-Newton method that has the potential of being faster and more robust than conjugate-gradient methods when solving nonlinear problems. Tests with a linear inverse problem for velocity analysis with both synthetic and field data suggest that the Huber function gives far more robust model estimates than does a least-squares fit with or without damping.

INTRODUCTION

Robust error measures such as the ℓ^1 norm have found a number of applications in geophysics. As measures of data misfit, they show considerably less sensitivity to large measurement errors than least-squares (ℓ^2) measures. Since geophysical inverse problems are generally ill-posed, relatively noise insensitive misfit measures can yield far more stable estimates of earth parameters than does the ℓ^2 norm (Claerbout and Muir, 1973; Taylor et al., 1979; Chapman and Barrodale, 1983; Scales and Gersztenkorn, 1987; Scales et al., 1988). This insensitivity to large noise has a statistical interpretation: robust measures are related to long-tailed density functions in the same way that ℓ^2 is related to the short-tailed Gaussian density function (Tarantola, 1987).

A simple choice of robust measure is the ℓ^1 norm. Denoting the residual (misfit) components by r_i, $i = 1, \ldots . N$ (N being the number of data points), the ℓ^1 norm misfit function of the residual vector is $\sum_{i=1}^{N} |r_i|$. This function is not smooth: it is singular where any residual component vanishes. As a result, numerical minimization is difficult. Various approaches based, for example, on a linear programming viewpoint (Barrodale and Roberts, 1980) or iterative smoothing (Scales et al., 1988) have been used with success but require considerable tuning. Moreover, the singularity implies that small residuals are "taken as seriously" as large residuals, which may not be appropriate in all circumstances.

These drawbacks of the ℓ^1 norm have led to various proposals which combine robust treatment of large residuals with Gaussian treatment of small residuals. These proposals are known as "hybrid ℓ^1/ℓ^2" methods. For example, Bube and Langan (1997) apply an iteratively reweighted least-squares (IRLS) method to minimize a hybrid objective function on a tomography problem. More recently, Zhang et al. (2000) use an IRLS procedure to locate bed boundaries from electromagnetic data.

In this paper, we present a hybrid ℓ^1/ℓ^2 error measure (or norm) proposed by Huber (1973):

$$M_\epsilon(r) = \begin{cases} \dfrac{r^2}{2\epsilon} & 0 \le |r| \le \epsilon, \\ |r| - \dfrac{\epsilon}{2} & \epsilon < |r|, \end{cases} \tag{1}$$

where ϵ is the threshold between the ℓ^1 and ℓ^2 norms. We call $\sum_{i=1}^{N} M_\epsilon(r_i)$ the "Huber misfit function," or Huber function for short. Figure 1 shows the Huber norm as a function of the residual. It is smooth near zero residual, weights small residuals by mean square, and treats large residuals with ℓ^1. Because it is differentiable everywhere, it is reasonable to suppose that the Huber function is easier to minimize than ℓ^1 while still robust against large residuals.

Definition of the misfit via the Huber function results in a nonlinear optimization problem because any residual component r_i close to the threshold ϵ can oscillate between the ℓ^1 and ℓ^2 norms. In the first section following this introduction, we propose solving the optimization problem with a quasi-Newton method called limited-memory BFGS (Broyden, 1969; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970; Nocedal, 1980).

Presented at the 69th Annual Internation Meeting, Society of Exploration Geophysicists. Published on Geophysics Online January 30, 2003. Manuscript received by the Editor July 16, 2001; revised manuscript received January 6, 2003.
*Stanford University, Stanford Exploration Project, Geophysics Department, 397 Panama Mall, Stanford, California 94305-2215. E-mail: antoine@sep.stanford.edu.
‡Rice University, The Rice Inversion Project, Department of Computational and Applied Mathematics, Houston, Texas 77005. E-mail: symes@caam.rice.edu.

In the second section, we test our method and estimate the root mean square (rms) velocity (or slowness) of noisy common-midpoint (CMP) and shot gathers for synthetic and field seismic data. There, we compare the Huber norm with the ℓ^2 norm with and without regularization.

IMPLEMENTATION OF A NONLINEAR ALGORITHM

In this section, we propose minimizing any inverse problem involving the Huber norm with a quasi-Newton method. First, consider the linear system $\mathbf{Am} = \mathbf{d}$, where $\mathbf{m}$ is a vector of model parameters to be estimated, $\mathbf{d}$ a vector of observed data, and $\mathbf{A}$ a matrix or a seismic operator. We want to estimate $\mathbf{m}$ so that

$$f(\mathbf{m}) = |\mathbf{Am} - \mathbf{d}|_{Huber} = |\mathbf{r}|_{Huber} \sum_{i=1}^{N} M_\epsilon(r_i) \qquad (2)$$

is minimum. Because of the definition in equation (1), although $\mathbf{A}$ is linear, the minimization of $f(\mathbf{m})$ in equation (2) is not a linear problem; a particular point in the residual can oscillate between the ℓ^1 and ℓ^2 norms for different iterations. This difficulty can be overcome if we design a nonlinear solver that will converge to a minimum of $f(\mathbf{m})$. Some specially adapted Huber minimizers have been suggested (Ekblom and Madsen, 1989; Li and Swetist, 1998). One of our questions for our study was whether a standard nonlinear method, as opposed to a special solver, would perform satisfactorily in Huber estimation. We show that it does. In addition, these specially adapted minimizers work only when $\mathbf{A}$ is a matrix and not an operator.

In our implementation, we decide to use a quasi-Newton method. The quasi-Newton method is an iterative process where the solution to the problem is updated as follows:

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \lambda_k \mathbf{H}_k^{-1} \nabla f(\mathbf{m}_k), \qquad (3)$$

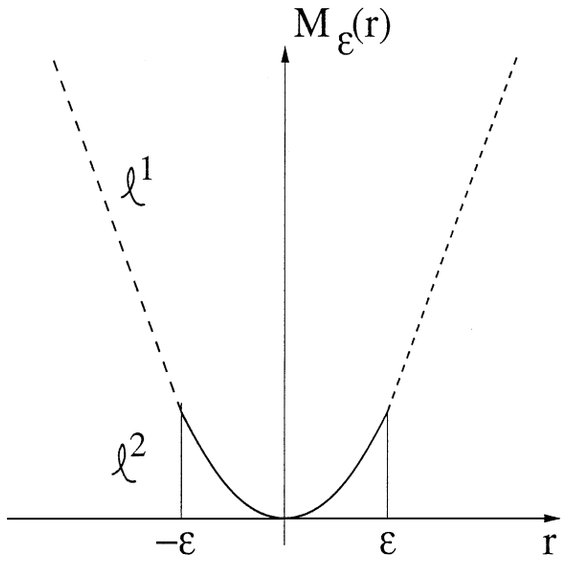

FIG. 1. Error measure proposed by Huber (Huber, 1973). The upper part above ϵ is the ℓ^1 norm (dashed line); the lower part is the ℓ^2 norm (solid line).

where $\mathbf{m}_{k+1}$ is the updated solution at iteration $k+1$, λ_k the step length computed by a line search that ensures a sufficient decrease of $f(\mathbf{m})$, and $\mathbf{H}_k$ is an approximation of the Hessian (or second derivative.)

Quasi-Newton methods intend to approximate the Hessian without explicitely computing it. The Hessian is then re-estimated and improved at each iteration of the descent method (Gill et al., 1986). Quasi-Newton methods can be equivalent to conjugate-gradient methods on a quadratic function, and they are often more robust than conjugate-gradient methods on general nonlinear problems. They also tend to require fewer iterations (Gill et al., 1986). However, one major drawback of the quasi-Newton methods is that they require the storage of the approximate Hessian in memory, which can be troublesome for large-scale problems.

In the Appendix, we present a computationally effective method that does not require the storage of the Hessian. This method is called limited-memory BFGS (L-BFGS) (Nocedal, 1980). With this technique, we store a limited number of solution-step and gradient-step vectors as the iterations go on. For each new iteration, these vectors are combined to form the approximate Hessian (see the Appendix for details). In the case where all the solution-step and gradient-step vectors are kept in memory at every iteration, the L-BFGS method becomes equivalent to the BFGS update (Nocedal, 1980). With the L-BFGS method, the storage needed is reduced compared to BFGS, and this makes it affordable to use for geophysical applications.

One computational burden is the line search algorithm that ensures sufficient decrease of the misfit function. This comes from the need to evaluate the misfit function many times before finding a good step length λ_k. In the Appendix, we propose testing the value $\lambda_k = 1$ first (Liu and Nocedal, 1989). This can save substantial computational time.

Another source of improvement is by scaling the Hessian at each iteration (Liu and Nocedal, 1989). This scaling greatly improves the performances of the quasi-Newton method. Liu and Nocedal (1989) show on numerical examples for large-scale problems that the L-BFGS method with the scaling of the Hessian and an appropriate line search algorithm (see the Appendix for details) is usually faster than conjugate-gradient methods using the Polak-Ribière formula (Kelley, 1999).

One problem with our choice of quasi-Newton method, however, is that the Huber function is not twice continuously differentiable. This assumption is at the heart of the convergence properties of the L-BFGS method. Nonetheless, the L-BFGS update requires the computation of the gradient only (see the Appendix). Furthermore, given that the approximate Hessian is not an exact representation of the real one, we expect the violation to this initial condition to be mild. Examples in the next section show that the method nontheless gives satisfying results and is robust to outliers, as expected.

In this section, we have proposed an efficient algorithm that will minimize any misfit function using the Huber norm. This algorithm is a limited-memory BFGS technique that saves computational time and memory requirement by (1) limiting the number of vectors kept in memory for the update of the Hessian $\mathbf{H}_k$, (2) testing a default value for the step length λ_k, and (3) scaling the Hessian at each iteration. In the next section, we test our algorithm on a geophysical problem and compare the Huber norm with the ℓ^2 norm with and without regularization.

APPLICATION OF THE HUBER NORM: VELOCITY ESTIMATION WITH NOISY DATA

In this section, we test our algorithm with a geophysical inverse problem which is velocity estimation with noisy data. Some possible applications of a robust solver are, for example, tomography (Bube and Langan, 1997) and deconvolution of noisy data (Chapman and Barrodale, 1983). Our goal is to demonstrate that the Huber function with the L-BFGS method gives a robust estimate of the model parameters when outliers are present in the data. The velocity estimation problem with the Huber norm has potential applications when multiples need to be separated from the signal in the velocity domain (Lumley et al., 1995; Kostov and Nichols, 1995). More conventional multiple attenuation techniques using the parabolic Radon transform (Kabir and Marfurt, 1999; Herrmann et al., 2000) can also benefit from using the Huber norm.

The "velocity domain" representation of seismic data using the hyperbolic Radon transform (HRT) is an alternative to the standard CMP gather. Transformation of CMP data into the velocity domain (producing a velocity model or panel of the data) exhibits clearly the moveout inherent in the data and, therefore, forms a convenient basis for velocity analysis as a linear inverse problem.

Thorson and Claerbout (1985) were the first to define the forward and adjoint operators of the HRT, formulating it as an inverse problem in which the velocity domain is the unknown space. In their approach the forward operator $\mathbf{A}$ maps the model space (velocity domain) into the data space (CMP gathers). This transformation is a superposition of hyperbolas in the data space. The adjoint operator $\mathbf{A}^\dagger$ (the HRT) maps the data space into the model space. This transformation is a summation over hyperbolic trajectories in the data space [related to the velocity stack as defined by Taner and Koehler (1969)]. With $d(t,\ x)$ being a CMP gather and $m(\tau,\ s)$ the corresponding velocity model, the forward operation is

$$d(t,x)=\sum_{s=s_{\min}}^{s_{\max}} w_o m\left(\tau=\sqrt{t^2-s^2x^2},s\right), \tag{4}$$

and the adjoint transformation is

$$m(\tau,s)=\sum_{x=x_{\min}}^{x_{\max}} w_o d\left(\tau=\sqrt{\tau^2+s^2x^2},x\right), \tag{5}$$

where x is the offset ($x_{\min}$ and $x_{\max}$ being the offset range), s the slowness ($s_{\min}$ and $s_{\max}$ being the range of slownesses investigated), τ the two-way zero offset traveltime, and w_o a weighting function that compensates to some extent for geometrical spreading and other effects (Claerbout and Black. 1997).

Having defined the forward operator $\mathbf{A}$ and its adjoint $\mathbf{A}^\dagger$, we can now pose the inverse problem. Inverse theory helps us to find a velocity panel which synthesizes a given CMP gather via the operator $\mathbf{A}$. In equations, given data $\mathbf{d}$ (CMP gather), we want to solve for the model $\mathbf{m}$ (velocity panel)

$$\mathbf{Am}=\mathbf{d}, \tag{6}$$

which leads in a least-squares sense to the linear system ("normal equations")

$$\mathbf{A}^\dagger\mathbf{Am}=\mathbf{A}^\dagger\mathbf{d}. \tag{7}$$

This system is easy to solve if $\mathbf{A}^\dagger\mathbf{A}\approx\mathbf{I}$, i.e., if $\mathbf{A}$ is close to unitary. Unfortunately, $\mathbf{A}$ is far from an unitary operator (Sacchi and Ulrych, 1995; Kabir and Marfurt, 1999). In addition, the number of equations and unknowns may be large, making an iterative data-fitting approach reasonable.

Consequently, with E being a misfit measurement function, our goal is to iteratively calculate the model $\mathbf{m}$ that minimizes the misfit function

$$f(\mathbf{m})=E(\mathbf{Am}-\mathbf{d}). \tag{8}$$

One possibility for E is the ℓ^2 norm (least-squares inversion). The misfit function is then usually minimized with conjugate-gradient methods. Another possible approach is, of course, by taking the Huber norm along with the L-BFGS method introduced in the preceding section. The two norms are compared in the next section for velocity estimation problems. The results show that the Huber norm gives the expected ℓ^1 behavior when outliers (non-Gaussian noise) are present in the data.

Synthetic data results

Figure 2 displays the synthetic model. In Figure 2a, we show the ideal velocity space with five events at different slownesses. In Figure 2b, we show the five corresponding hyperbolas in the CMP domain. Finally in Figure 2c, we add four spikes to the CMP gather in Figure 2b to make Gaussian statistics unsuitable. The energy of the four spikes is five times the energy of the five hyperbolas. Our goal is to find the velocity field $\mathbf{m}$ that will best fit a CMP gather $\mathbf{d}$ via the HRT.

Figure 3 shows the result of the inversion when the ℓ^2 norm is used for the noise-free data. In Figure 3a, we display the velocity space with its five events. The focusing is not perfect and some artifacts appear (Sacchi and Ulrych, 1995). Figure 3b shows the remodeled data after inversion, and Figure 3c displays the residual (difference between the input and remodeled data). We conclude that the inversion reached a minimum since no coherent energy is left in the residual; the data fitting is very good. Because the input data have Gaussian statistics, the performance of the least-squares inversion was expected.

We now use the same inversion scheme with the ℓ^2 norm but with the "contaminated" CMP gather (Figure 2c). Notice that we do not apply any regularization on the least-squares method. The final result is shown in Figure 4. In this case, as expected, the ℓ^2 inversion creates a number of artifacts both in the model and data space. In Figure 5, we use least-squares with a simple damping in the regularization. The model and data space are cleaner, but the difference between the input and the remodeled data or residual is still big (compare Figure 5c to Figure 3c.) In addition, we see artifacts in the inverted slowness field and the reconstructed data.

Figure 6 displays the result of the inversion with the Huber norm. The outcome of the inversion is insensitive to the spiky events, just like a pure ℓ^1 norm misfit function. The residual (Figure 6c) exhibits the four spikes very clearly. This result demonstrates that our algorithm, although not specifically designed to minimize the Huber function, converges to a satisfying solution. The next section shows inversion results with noisy field data.

Field data results

We now test our algorithm with a field data example. We use a shot gather from a land-data survey in the Middle East. The trajectories of the events in Figure 7 look "hyperbolic" enough to be inverted with our method. Note that, in theory, we should resort the data into CMP gathers before doing the inversion. This data set is particularly interesting because it has amplitude anomalies at short offset (the red dots in Figure 7 show the clipped values) and a low-velocity coherent noise that is probably due to guided energy in the near surface. We could get rid of the amplitude anomalies by applying an automatic gain control (AGC) on the data, but AGC is a nonlinear process that we should not use if we want to preserve the prestack amplitudes of the data.

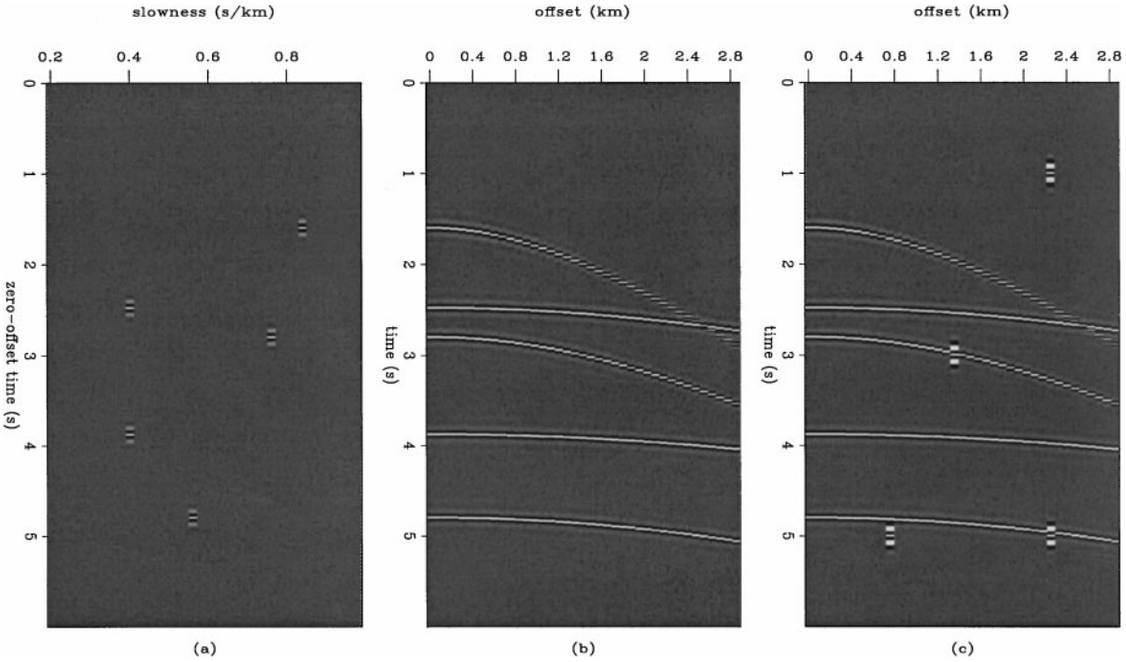

FIG. 2. Synthetic data used for the inversion. (a) The true velocity model represented in slowness. (b) The noise-free input data used for the inversion. (c) The same data but with noise added (four spikes).

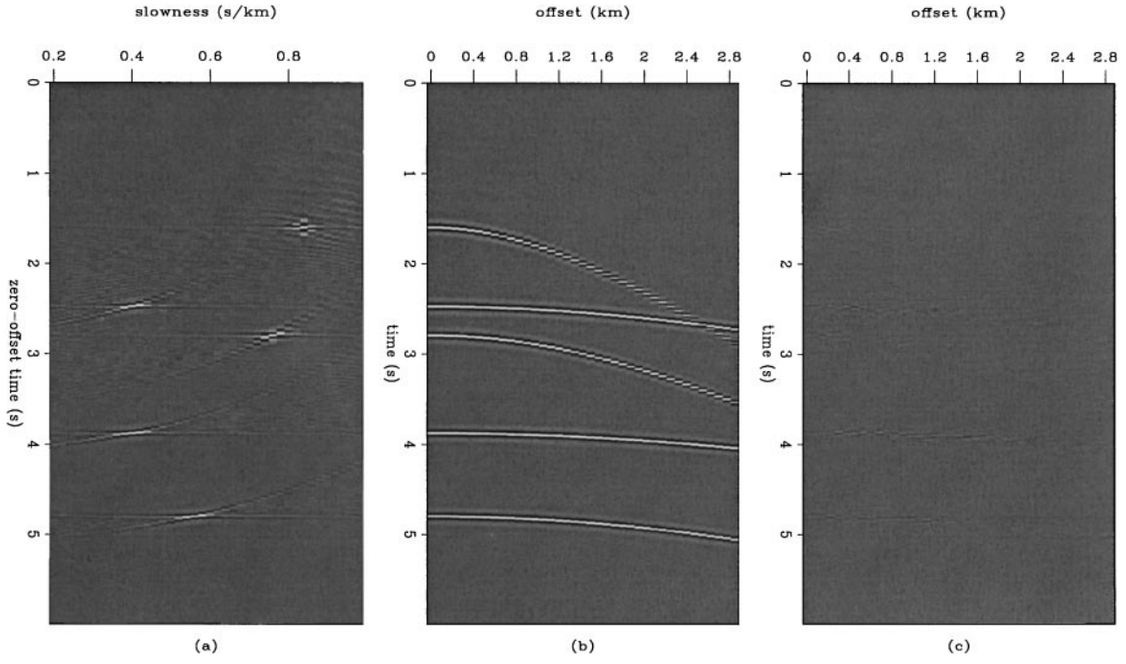

FIG. 3. Result of the inversion with the ℓ^2 norm for the noise-free data. (a) Inverted slowness field. (b) Remodeled data. (c) Difference between the input (Figure 2b) and the remodeled data.

We first invert this gather with the ℓ^2 norm without regularization (Figure 8). Figure 8a displays the velocity domain obtained after the least-squares inversion. The main velocity event is masked with horizontal stripes coming from the short-offset amplitude anomalies. The reconstructed data (Figure 8b) show spurious noise at large offset and other inversion artifacts. We now show in Figure 9 the result of the damped least-squares fit. The inverted slowness field is much cleaner, but the horizontal stripes remain. We notice that we have the same velocity from the top to the bottom in Figure 9a. This shows that our

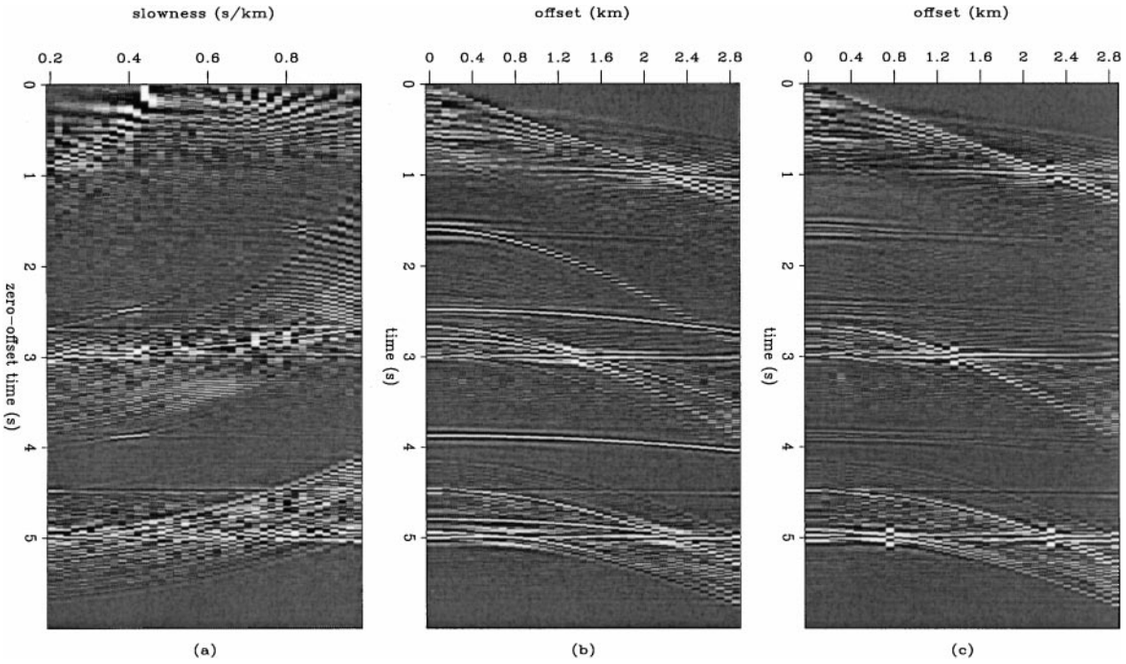

FIG. 4. Result of the inversion with the ℓ^2 norm for the data with noise. (a) Inverted slowness field. (b) Remodeled data. (c) Difference between the input (Figure 2c) and the remodeled data. The four spikes create artifacts in both the inverted model and the reconstructed data space.

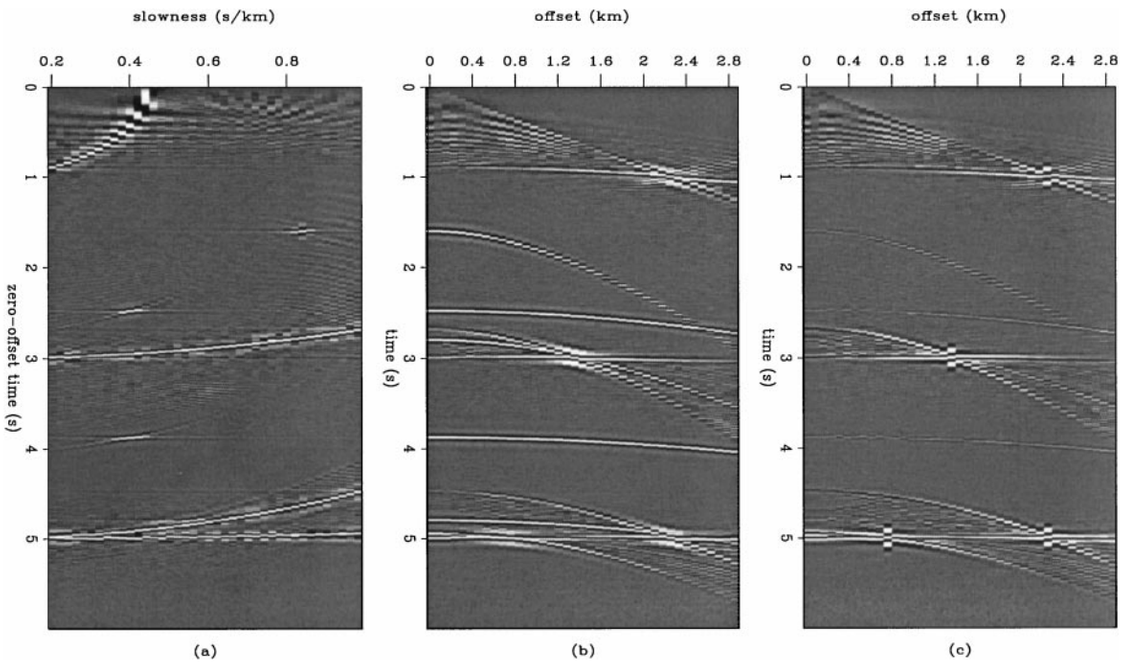

FIG. 5. Result of the inversion with the ℓ^2 norm with damping for the data with noise. (a) Inverted slowness field. (b) Remodeled data. (c) Difference between the input (Figure 2c) and the remodeled data. The three panels are cleaner than in Figure 4, but some artifacts remain, however.

data are contaminated with multiples generated in the near surface. Figure 10 displays the inversion result with the Huber norm and demonstrates the robustness of our method. We obtain a very well focused velocity corridor as opposed to the ℓ^2 result in Figure 8a. In addition, the horizontal stripes have disappeared.

The synthetic and field data examples demonstrate that the Huber norm can be an efficient alternative to the ℓ^2 norm when outliers (or non-Gaussian noise) are present in the data.

CONCLUSION

Since geophysical inverse problems are often ill-posed due to the presence of inconsistent data, high amplitude anomalies, and outliers, relative insensitivity to noise is a desirable characteristic of an inversion method. The Huber function is a compromise misfit measure between the ℓ^1 and ℓ^2 norms. It not only improves robustness in the presence of noise and outliers with an ℓ^1 measure, but also keeps smoothness for small residuals with an ℓ^2 measure.

In this paper, we have proposed minimizing the Huber function with a quasi-Newton method called limited-memory BFGS. This method has the potential of being faster and more robust than conjugate-gradient methods for solving nonlinear problems. Tests with noisy synthetic and field data examples demonstrate that our method is robust to outliers present in the data space, as expected.

Finally, we think that the Huber norm and the quasi-Newton method are a possible alternative to the more traditional IRLS method for robust inversion of seismic data. Therefore, one value of the Huber norm is that it opens new horizons in the design of robust solvers.

ACKNOWLEDGMENTS

The authors thank the sponsors of the Stanford Exploration Project and of the Rice Inversion Project for their financial support. We also thank the associate editor and the reviewers for their constructive comments.

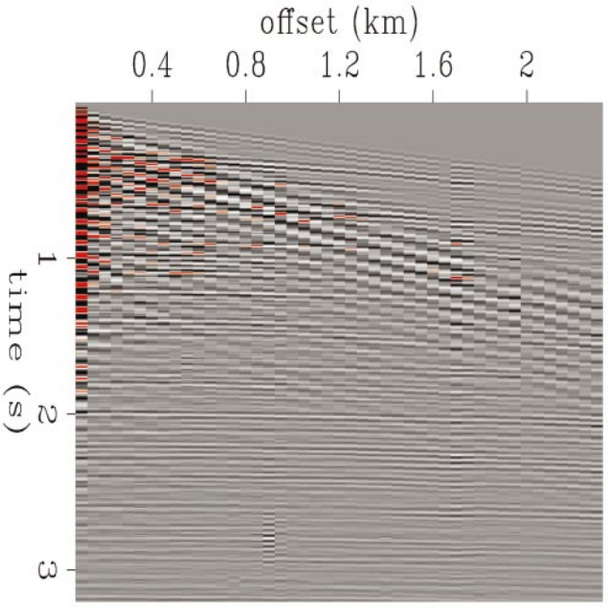

FIG. 7. The field data used for the inversion. Notice the amplitude anomalies at near offset and the time shift near offset 1.6 km. The red color shows the clipped value (clip = 0.3).

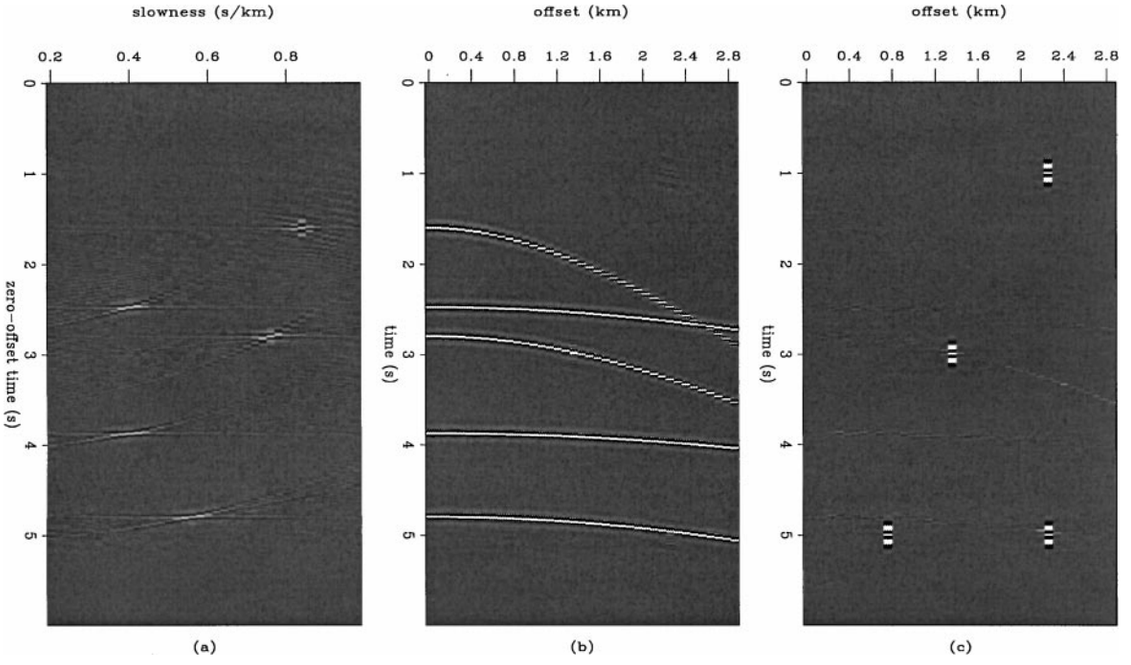

FIG. 6. Result of the robust inversion with the Huber norm for the data with noise. (a) Inverted slowness field. (b) Remodeled data. (c) Difference between the input (Figure 2c) and the remodeled data. The Huber norm behaves like a pure ℓ^1 norm since all artifacts have disappeared.

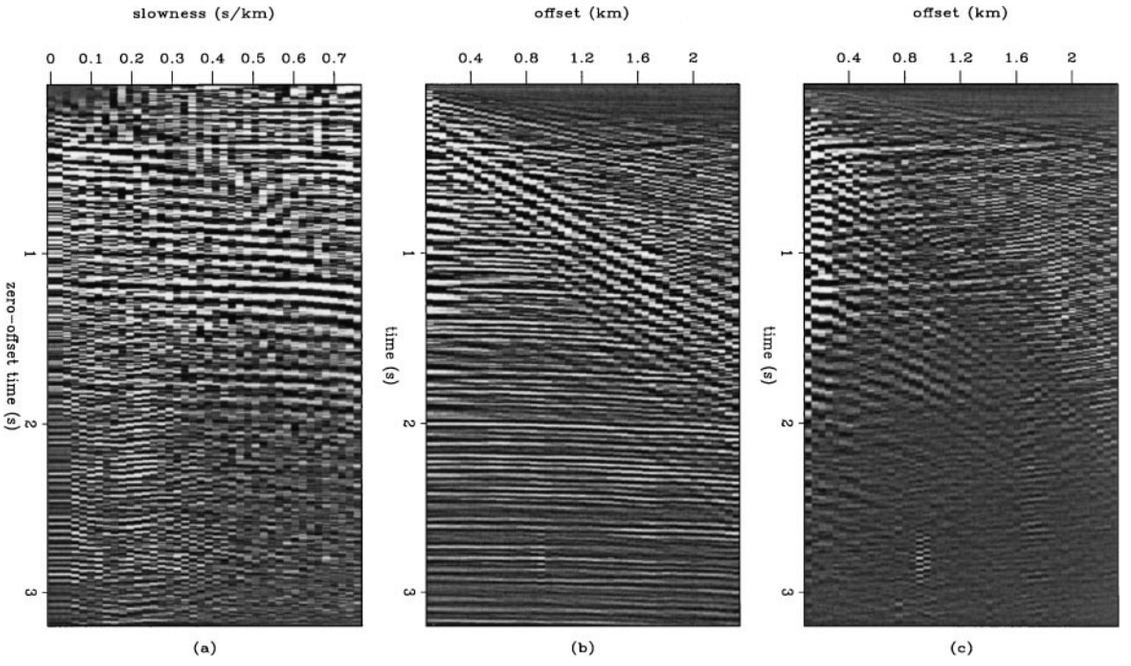

FIG. 8. The result of the inversion with the ℓ^2 norm for the field data. (a) Inverted slowness field. (b) Remodeled data. (c) Difference between the input (Figure 7) and the remodeled data. The horizontal stripes in the velocity panel are created by the amplitude anomalies at short offsets.

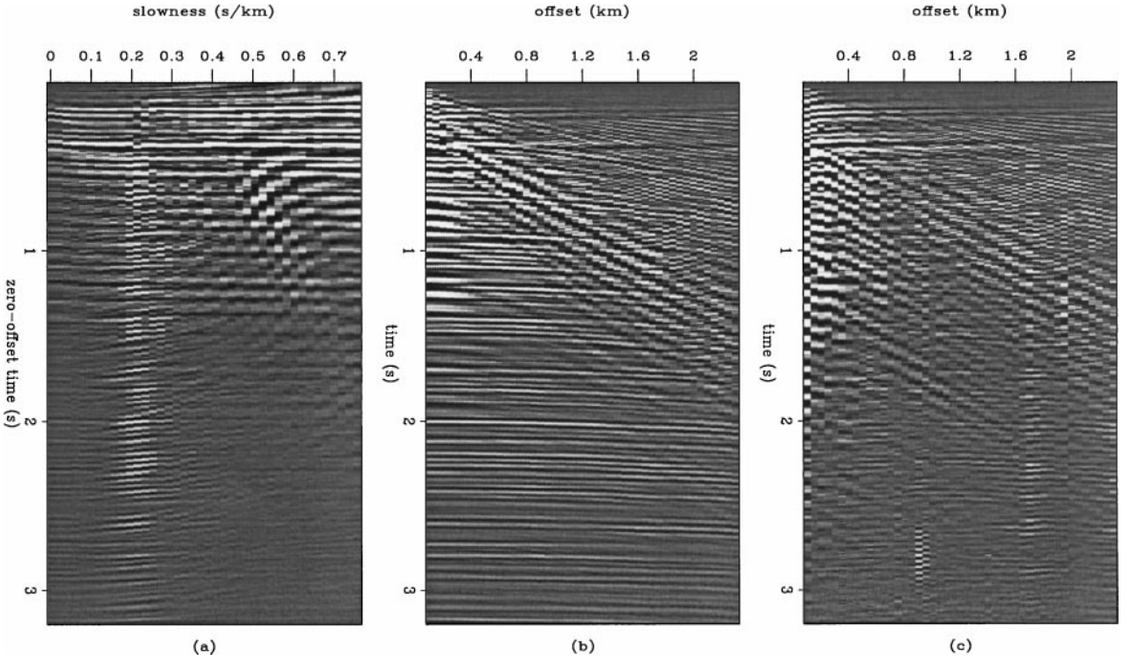

FIG. 9. The result of the inversion with the ℓ^2 norm and regularization for the field data. (a) Inverted slowness field. (b) Remodeled data. (c) Difference between the input (Figure 7) and the remodeled data. The model is much cleaner than in Figure 8a, but the horizontal events remain. The shot gather in Figure 7 is contaminated with multiples generated in the near surface, which explains the single velocity trend in the inverted slowness field.

REFERENCES

Barrodale, I., and Roberts, F. D. K., 1980, Algorithm 552: Solution of the constrained $\ell 1$ linear approximation problem: ACM Trans. Mathematical Software, **6**, 231–235.

Broyden, C. G., 1969, A new double-rank minimization algorithm: AMS Notices, **16**, 670.

Bube, K. P., and Langan, R. T., 1997, Hybrid λ_1/λ_2 minimization with applications to tomography: Geophysics, **62**, 1183–1195.

Chapman, N. R., and Barrodale, I., 1983, Deconvolution of marine seismic data using the $\ell 1$ norm: Geophys., J. Roy. Astr. Soc., **72**, 93–100.

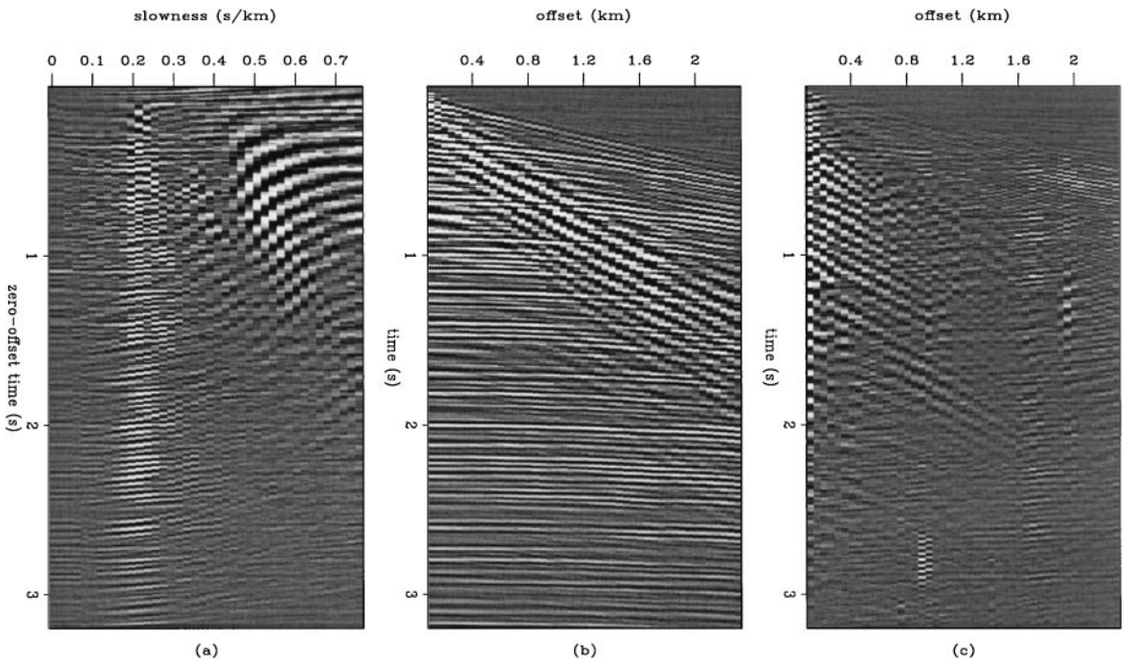

FIG. 10. The result of the robust inversion with the Huber norm for the field data. (a) Inverted slowness field. (b) Remodeled data. (c) Difference between the input (Figure 7) and the remodeled data. The velocity space displays a very well focused corridor as a result of the robust inversion.

Claerbout, J. F., and Black, J. L., 1997, Basic earth imaging: Class notes, http://sepwww.stanford.edu/sep/prof/index.html.

Claerbout, J. F., and Muir, F., 1973, Robust modeling with erratic data: Geophysics, **38**, 826–844.

Darche, G., 1989, Iterative L1 deconvolution: Stanford Exploration Project Annual Report, **61**, 281–301.

Ekblom, H., and Madsen, K., 1989, Algorithms for non-linear Huber estimation: BIT, **29**, 60–76.

Fletcher, R., 1970, A new approach to variable metric methods: Comput. J., **13**, 317–322.

Gill, P. H., Murray, W., and Wright, M. H., 1986, Practical optimization: Academic Press.

Goldfarb, D., 1970, A family of variable metric methods derived by variational means: Math. Comp., **24**, 23–26.

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

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

Kabir, N. M. M., and Marfurt, K. J., 1999, Toward true amplitude multiple removal: The Leading Edge, **18**, 66–73.

Kelley, C. T., 1999, Iterative methods for optimization: Soc. Ind. Appl. Math.

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

Li, W., and Swetist, J. J., 1998, The linear 11 estimator and the Huber M-estimator: SIAM J. Optim., **8**, 457–475.

Liu, D. C., and Nocedal, J., 1989, On the limited memory BFGS method for large scale optimization: Mathematical Programming, **45**, 503–528.

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

More, J. J., and Thuente, J., 1994. Line search algorithms with guaranteed sufficient decrease: ACM Trans. Mathematical Software, **20**, 286–307.

Nocedal, J., 1980, Updating quasi-Newton matrices with limited storage: Math. Comp., **95**, 339–353.

Sacchi, M. D., and Ulrych, T. J., 1995, High-resolution velocity gathers and offset space reconstruction: Geophysics, **60**, 1169–1177.

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

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

Shanno, D. F., 1970, Conditioning of quasi-Newton methods for function minimization: Math. Comp., **24**, 647–657.

Taner, M. T., and Koehler, F., 1969, Velocity spectra—Digital computer derivation and applications of velocity functions: Geophysics, **34**, 859–881.

Tarantola, A., 1987, Inverse problem theory: Elsevier.

Taylor, H. L., Banks, S. C., and McCoy, J. F., 1979, Deconvolution with the L1 norm: Geophysics, **44**, 39–52.

Thorson, J. R., and Claerbout, J. F., 1985, Velocity stack and slant stochastic inversion: Geophysics, **50**, 2727–2741.

Zhang, Z., Chunduru, R. K., and Jervis, M. A., 2000, Determining bed boundaries from inversion of EM logging data using general measures of model structure and data misfit: Geophysics, **65**, 76–82.

APPENDIX

PROPOSED ALGORITHM FOR MINIMIZING THE HUBER FUNCTION

We present a method for solving nonlinear problems that we later use to minimize the Huber function.

A quasi-Newton method for solving nonlinear problems

The method we present in this paper is suitable for smooth functions where local minima exist. It is not a method for global optimization where the global minimum is sought. We define $\mathbf{m}^*$ a local minimizer for $f(\mathbf{m})$, and we assume that $f(\mathbf{m})$ and $\mathbf{m}^*$ satisfy the "standard requirements" (1) f is twice differentiable, (2) $\nabla f(\mathbf{m}^*) = 0$, and (3) $\nabla^2 f(\mathbf{m}^*)$ is positive definite, i.e., $\mathbf{m}^\dagger \nabla^2 f(\mathbf{m}^*)\mathbf{m} > 0$ for all $\mathbf{m} \in \Re^N$ (†denotes the adjoint), where N is the dimension of the model vector $\mathbf{m}$ and $\Re^N$ the real space for the model vector $\mathbf{m}$. Any vector $\mathbf{m}^*$ that satisfies the standard requirements is a local minimizer of $f(\mathbf{m})$.

Newton's method is an iterative process where the solution to the problem is updated as follows:

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \lambda_k \mathbf{H}_k^{-1} \nabla f(\mathbf{m}_k), \tag{A-1}$$

where $\mathbf{m}_{k+1}$ is the updated solution at iteration $k+1$, λ_k the step length computed by a line search that ensures a sufficient decrease of $f(\mathbf{m})$, and $\mathbf{H}_k = \nabla^2 f(\mathbf{m}_k)$, the Hessian (or second derivative). In many circumstances, the inverse of the Hessian can't be computed directly. It happens, for example, when the matrix $\mathbf{H}$ is too big or when operators are used for $\mathbf{A}$ rather than matrices. Fortunately, we might be able to compute an approximation of the Hessian of $f(\mathbf{m})$. This strategy gives birth to quasi-Newton methods, where the way in which the Hessian is computed determines the method (Kelley, 1999).

A possible update of the Hessian is given by the BFGS technique (Broyden, 1969; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970). The BFGS update is given by

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{y}\mathbf{y}^\dagger}{\mathbf{y}^\dagger \mathbf{s}} - \frac{(\mathbf{H}_k \mathbf{s})(\mathbf{H}_k \mathbf{s})^\dagger}{\mathbf{s}^\dagger \mathbf{H}_k \mathbf{s}}, \tag{A-2}$$

where $\mathbf{s} = \mathbf{m}_{k+1} - \mathbf{m}_k$, and $\mathbf{y} = \nabla f(\mathbf{m}_{k+1}) - \nabla f(\mathbf{m}_k)$. In practice, however, we rather write the previous equation in terms of the inverse matrices. We have then

$$\mathbf{H}_{k+1}^{-1} = \left(\mathbf{I} - \frac{\mathbf{s}\mathbf{y}^\dagger}{\mathbf{y}^\dagger \mathbf{s}}\right) \mathbf{H}_k^{-1} \left(\mathbf{I} - \frac{\mathbf{y}\mathbf{s}^\dagger}{\mathbf{y}^\dagger \mathbf{s}}\right) + \frac{\mathbf{s}\mathbf{s}^\dagger}{\mathbf{y}^\dagger \mathbf{s}}. \tag{A-3}$$

In addition, we use the history of the iterations to compute the new Hessian rather than a full storage of the matrix $\mathbf{H}_k^{-1}$. This requires that a gradient step vector $\mathbf{y}$ and a solution step vector $\mathbf{s}$ are kept in memory after each iteration. Consequently this method might not be affordable with large data and model spaces. In the next section, we propose a modified version of the BFGS method that limits the storage needed to compute the update of the Hessian.

The limited memory BFGS method

Nocedal (1980) derives a technique that partially solves the storage problem caused by the BFGS update. Instead of keeping all the $\mathbf{s}$ and $\mathbf{y}$ from the past iterations, we update the Hessian using the information from the ℓ previous iterations, where ℓ is given by the end user. This implies that when the number of iterations is smaller than ℓ. we have the usual BFGS update, and when it is larger than ℓ, we have a limited memory BFGS (L-BFGS) update.

We give the updating formulas of the Hessian as presented by Nocedal (1980). First, we define

$$\rho_i = 1 \big/ \mathbf{y}_i^\dagger \mathbf{s}_i, \quad \mathbf{v}_i = \left(I - \rho_i \mathbf{y}_i \mathbf{s}_i^\dagger\right) \quad \text{and} \quad \mathbf{H}^{-1} = \mathbf{B}.$$

As described above, when k (the iteration number) obeys $k+1 \le \ell$, where ℓ is the storage limit, we have the BFGS update:

$$\begin{aligned} \mathbf{B}_{k+1} = {} & \mathbf{v}_k^\dagger \mathbf{v}_{k-1}^\dagger \cdots \mathbf{v}_0^\dagger \mathbf{B}_0 \mathbf{v}_0 \cdots \mathbf{v}_{k-1} \mathbf{v}_k \\ & + \mathbf{v}_k^\dagger \cdots \mathbf{v}_1^\dagger \rho_0 \mathbf{s}_0 \mathbf{s}_0^\dagger \mathbf{v}_1 \cdots \mathbf{v}_k \\ & \vdots \\ & + \mathbf{v}_k^\dagger \rho_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^\dagger \mathbf{v}_k \\ & + \rho_k \mathbf{s}_k \mathbf{s}_k^\dagger. \end{aligned} \tag{A-4}$$

For $k+1 > \ell$, we have the limited memory update:

$$\begin{aligned} \mathbf{B}_{k+1} = {} & \mathbf{v}_k^\dagger \mathbf{v}_{k-1}^\dagger \cdots \mathbf{v}_{k-\ell+1}^\dagger \mathbf{B}_0 \mathbf{v}_{k-\ell+1} \cdots \mathbf{v}_{k-1} \mathbf{v}_k \\ & + \mathbf{v}_k^\dagger \cdots \mathbf{v}_{k-\ell+2}^\dagger \rho_{k-\ell+1} \mathbf{s}_{k-\ell+1} \mathbf{s}_{k-\ell+1}^\dagger \mathbf{v}_{k-\ell+2} \cdots \mathbf{v}_k \\ & \vdots \\ & + \mathbf{v}_k^\dagger \rho_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^\dagger \mathbf{v}_k \\ & + \rho_k \mathbf{s}_k \mathbf{s}_k^\dagger. \end{aligned} \tag{A-5}$$

These equations show how the update of the Hessian is calculated.

Usually, the L-BFGS method is implemented with a line search for the step length λ_k to ensure a sufficient decrease of the misfit function. Convergence properties of the L-BFGS method are guaranteed if λ_k in equation (A-1) satisfies the Wolfe conditions (Kelley, 1999):

$$f(\mathbf{x}_k + \lambda_k \mathbf{d}_k) \le f(\mathbf{x}_k) + \mu \lambda_k \nabla f(\mathbf{x}_k)^\dagger \mathbf{d}_k, \tag{A-6}$$

$$\left|\nabla f(\mathbf{x}_k + \lambda_k \mathbf{d}_k)^\dagger \mathbf{d}_k\right| \ge \nu \left|\nabla f(\mathbf{x}_k)^\dagger \mathbf{d}_k\right|, \tag{A-7}$$

where ν and μ are constants to be chosen a priori, and $\mathbf{d}_k = -\mathbf{B}_k \nabla f(\mathbf{m}_k)$. For ν and μ, we set $\nu = 0.9$ and $\mu = 10^{-4}$, as proposed by Liu and Nocedal (1989). Equation (A-6) is a sufficient decrease condition that all line search algorithms must satisfy. Equation (A-7) is a curvature condition. The line search algorithm has to be carefully designed since it absorbs most of the computing time. We programmed a line search based on the More and Thuente (1994) method. Because the line search is time-consuming, the step length $\lambda_k = 1$ is always tested first. This procedure saves a lot of computing time and is also recommended by Liu and Nocedal (1989). We now give the algorithm used to minimize any objective function involving nonlinear problems.

An efficient algorithm for solving nonlinear problems

The solver works as follows:

1) Choose $\mathbf{m}_0$, ℓ, $\mathbf{B}_0$. Set $k = 0$.
2) Compute

$$\mathbf{d}_k = -\mathbf{B}_k \nabla f(\mathbf{m}_k), \tag{A-8}$$

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \lambda_k \mathbf{d}_k, \tag{A-9}$$

   where λ_k meets the Wolfe conditions.
3) Let $\hat{\ell} = \min\{k, \ell - 1\}$. Update $\mathbf{B}_0 \hat{\ell} + 1$ times using the pairs $\{\mathbf{y}_i,\ \mathbf{s}_i\}_{j=k-\hat{\ell}}^{k}$, i.e., let

$$\begin{aligned} \mathbf{B}_{k+1} = {} & \mathbf{v}_k^\dagger \mathbf{v}_{k-1}^\dagger \cdots \mathbf{v}_{k-\hat{\ell}}^\dagger \mathbf{B}_0 \mathbf{v}_{k-\hat{\ell}} \cdots \mathbf{v}_{k-1} \mathbf{v}_k \\ & + \mathbf{v}_k^\dagger \cdots \mathbf{v}_{k-\hat{\ell}+1}^\dagger \rho_{k-\hat{\ell}} \mathbf{s}_{k-\hat{\ell}} \mathbf{s}_{k-\hat{\ell}}^\dagger \mathbf{v}_{k-\hat{\ell}-1} \cdots \mathbf{v}_k \end{aligned} \tag{A-10}$$

$$\begin{aligned} & \vdots \\ & \mathbf{v}_k^\dagger \rho_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^\dagger \mathbf{v}_k \\ & + \rho_k \mathbf{s}_k \mathbf{s}_k^\dagger. \end{aligned} \tag{A-11}$$

4) Set $k = k + 1$ and go to 2 if the residual power is not small enough.

The update $\mathbf{B}_{k-1}$ is not formed explicitly; instead, we compute $\mathbf{d}_k = -\mathbf{B}_k \nabla f(\mathbf{x}_k)$ with an iterative formula (Nocedal, 1980). Liu and Nocedal (1989) propose scaling the initial symmetric positive definite $\mathbf{B}_0$ at each iteration, as follows:

$$\mathbf{B}_k^0 = \frac{\mathbf{y}_k^\dagger \mathbf{s}_k}{\|\mathbf{y}_k\|_2^2}\mathbf{B}_0. \tag{A-12}$$

This scaling greatly improves the performances of the method. Liu and Nocedal (1989) show that the storage limit for large-scale problems has little effects. A common choice for ℓ is $\ell = 5$. In practice, the initial guess $\mathbf{B}_0$ for the Hessian is the identity matrix $\mathbf{I}$; then it might be scaled as proposed in equation (A-12). The nonlinear solver as detailed in the previous algorithm converges to a local minimizer $\mathbf{m}^*$ of $f(\mathbf{m})$.

Minimizing the Huber function

So far, we have introduced a general method for solving nonlinear problems. In this section, we show how this algorithm can be used when the Huber function is used for measuring the data misfit. In fact, we only need to derive the gradient of the objective function in equation (2). The gradient can be written in the following compact form (Li and Swetist, 1998):

$$\nabla f(\mathbf{m}) = \mathbf{A}^\dagger(\mathbf{A}\mathbf{m} - \mathbf{d})^\epsilon_{-\epsilon}. \tag{A-13}$$

where $\mathbf{z}^\epsilon_{-\epsilon}$ is a vector with ith component

$$z_i \leftarrow \max\{-1, \min\{1, z_i/\epsilon\}\}. \tag{A-14}$$

A last difficulty arises in the choice of the threshold ϵ in equation (1). This value remains constant during the iterations; ϵ is also the only parameter to choose a priori for different problems. We have not derived any analytical expression for ϵ, but based on previous works with IRLS methods (Darche, 1989), it seems that

$$\epsilon = \frac{\max|\mathbf{d}|}{100} \tag{A-15}$$

is a good practical choice. Another possible solution is to set ϵ at the 98th percentile of the data (J. Claerbout, 2000, Personal communication).