Stanford Exploration Project, Report 120, May 3, 2005, pages 159–166

Flattening without picking faults

Jesse Lomask, Antoine Guitton, and Alejandro Valenciano[1]

ABSTRACT

We show that iteratively re-weighted least squares (IRLS) can flatten data cubes with vertically-oriented faults without having to pick the faults. One requirement is that the faults need to have at least part of their tip-lines (fault terminations) encased within the 3D cube. We demonstrate this method's flattening ability on a faulted 3D field data-set.

INTRODUCTION

Flattening algorithms (Lomask and Claerbout, 2002; Lomask, 2003; Lomask et al., 2005) are able to flatten seismic data cubes with faults by summing the dips into time shifts around the faults and ignoring the dips across the faults. In order to know which dips to ignore and which dips to honor during inversion, we require a fault indicator (data residuals weight that throw away bad dips). This indicator could be either manually picked or automatically generated by an automatic fault detector.

While the smoothness of the summed time shifts are often justifiable in non-faulted areas, the time shifts can change abruptly across the faults. In these situations, we desire an inversion technique that yields smooth time shifts in non-faulted areas while preserving sharp time shifts across the faults. In addition, no pre-defined fault indicator should be supplied.

In this paper, we present an automatic edge-preserving method for flattening faulted data without requiring an input fault model. The method uses iterative re-weighted least-squares (IRLS). A Geman-McClure weight function (fault indicator) of data residuals is computed at each non-linear iteration to allow outliers in the data residuals.

The only requirements are that part of the fault tip-lines are encased in the data and that the faults are oriented vertically. The resulting weight generated by this IRLS method is a fault indicator cube that best flattens the data. This is an important difference from many traditional automatic fault detectors that are defined by local discontinuities.

[1]**email:** lomask@sep.stanford.edu, antoine@sep.stanford.edu, valencia@sep.stanford.edu

METHODOLOGY

We modify the flattening algorithm described in Lomask et al. (2005). The key regression is this overdetermined system:

$$\nabla \boldsymbol{\tau} \quad = \quad \mathbf{p}. \tag{1}$$

This equation means that we need to find a time-shift field $\boldsymbol{\tau}(x,y,z)$ such that its gradient approximates the dip $\mathbf{p}(\mathrm{x},\mathrm{y},\boldsymbol{\tau})$. It is overdetermined because we are finding a single summed time shift field such that its two-dimensional gradient matches the dip in the x and y directions.

Imagine that after solving equation (1), the data residuals consist of spikes separated by relatively large distances. Then the estimated time shifts $\boldsymbol{\tau}$ would be piecewise smooth with jumps at the spike locations (fault locations), which is what we desire. However, in solving Equation (1), we use the least-squares criterion – minimization of the ℓ_2 norm of the residual. Large spikes in the residual tend to be attenuated. In the model space, the solver smooths the time shifts across the spike location.

It is known that the ℓ_1 norm is less sensitive to spikes in the residual (Claerbout and Muir, 1973; Darche, 1989; Nichols, 1994). Minimization of the ℓ_1 norm makes the assumption that the residuals are exponentially distributed and have a "long-tailed" distribution relative to the Gaussian function assumed by the ℓ_2 norm inversion.

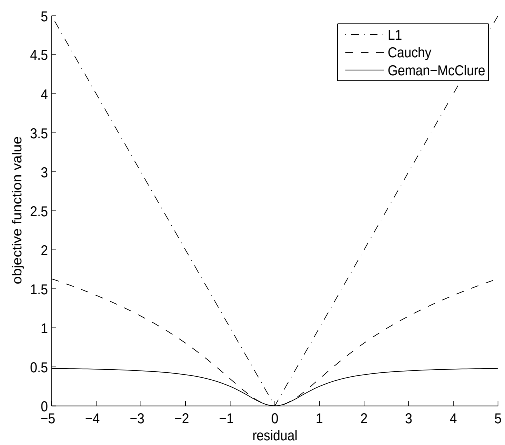

Figure 1: Comparison of the loss functions for the ℓ_1, Cauchy, and Geman-McClure functions. Notice that the Geman-McClure is the most robust. jesse2-2 [CR]

Here, we compute non-linear model residual weights which force a Geman-McClure distribution, another long-tailed distribution which approximates an exponential distribution. A comparison of the loss functions for the ℓ_1, Cauchy (Claerbout, 1999), and Geman-McClure functions is shown in Figure 1. Notice that the Geman-McClure is the most robust in that it is the least sensitive to outliers. Our weighted regression now is the following over-determined system:

$$\mathbf{W^{j-1}} \nabla \boldsymbol{\tau}^{\mathbf{j}} \quad = \mathbf{W^{j-1}} \mathbf{p}, \tag{2}$$

where $\mathbf{W^{j-1}}$ is the weight computed at the previous non-linear iteration.

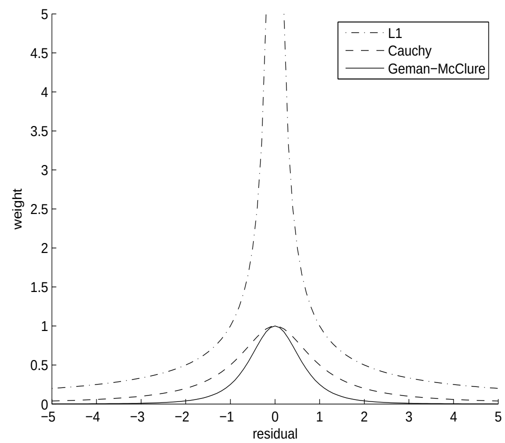

Figure 2: Comparison of weight functions for the ℓ_1, Cauchy, and Geman-McClure functions. The Geman-McClure is most restrictive, this may be desirable for weighting faults. jesse2-3 [CR]

Our method consists of recomputing the weights at each non-linear iteration, solving small piecewise Gauss-Newton linear problems. The IRLS algorithms converge if each minimization reaches a minimum for a constant weight. We perform the following non-linear iterations: starting with the weights $\mathbf{W^0} = \mathbf{I}$, at the j^{th} iteration the algorithm solves

iterate {

$$\mathbf{r} = \mathbf{W}[\nabla \boldsymbol{\tau}_\mathbf{k} - \mathbf{p}] \tag{3}$$

$$\Delta\boldsymbol{\tau} = (\nabla'\nabla)^{-1}\nabla'\mathbf{r} \tag{4}$$

$$\boldsymbol{\tau}_{k+1} = \boldsymbol{\tau}_k + \Delta\boldsymbol{\tau} \tag{5}$$

}

At every non-linear iteration j^{th} iteration we re-estimate our Geman-McClure weight function:

$$\mathbf{W}^{j-1} = \mathbf{diag}\left(\frac{1}{\left(1+(\mathbf{r}^{j-1})^2/\bar{r}^2\right)^2}\right) \tag{6}$$

where $\bar{r}$ is an adjustable parameter. A comparison of weight functions for the ℓ_1, Cauchy, and Geman-McClure functions is shown in Figure 2. Notice that the Geman-McClure creates the tightest, most precise weight. This should have the advantage of having a surgical-like effect of down-weighting spurious dips.

FIELD DATA EXAMPLE

We tested this method on the Chevron Shoal data shown in Figure 3. At first glance, the 2D in-line section does not appear to have much faulting. However, there is actually a nearly vertical fault near the center, and dip estimators will return incorrect estimates at faults.

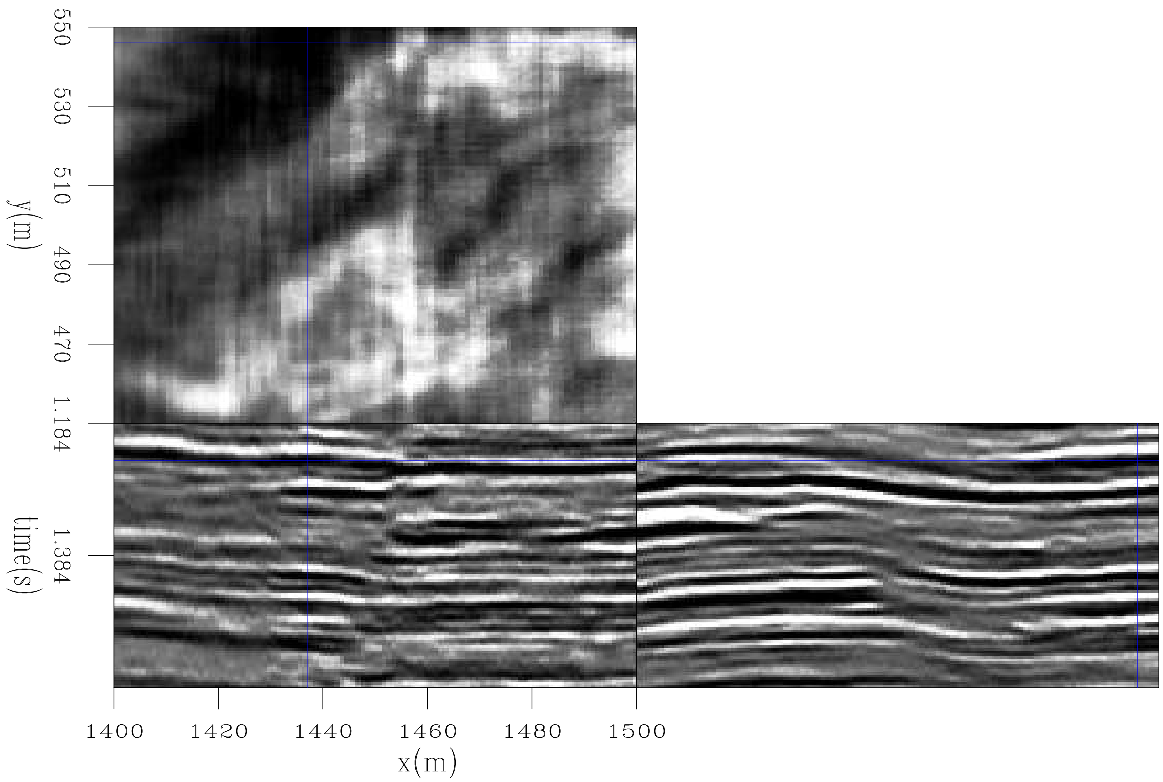

Figure 3: Faulted Chevron Gulf of Mexico data. Although it is difficult to see, the 2D vertical section shows fault with significant displacement, enough to cause our dip estimator to return erroneous values. jesse2-dat [ER]

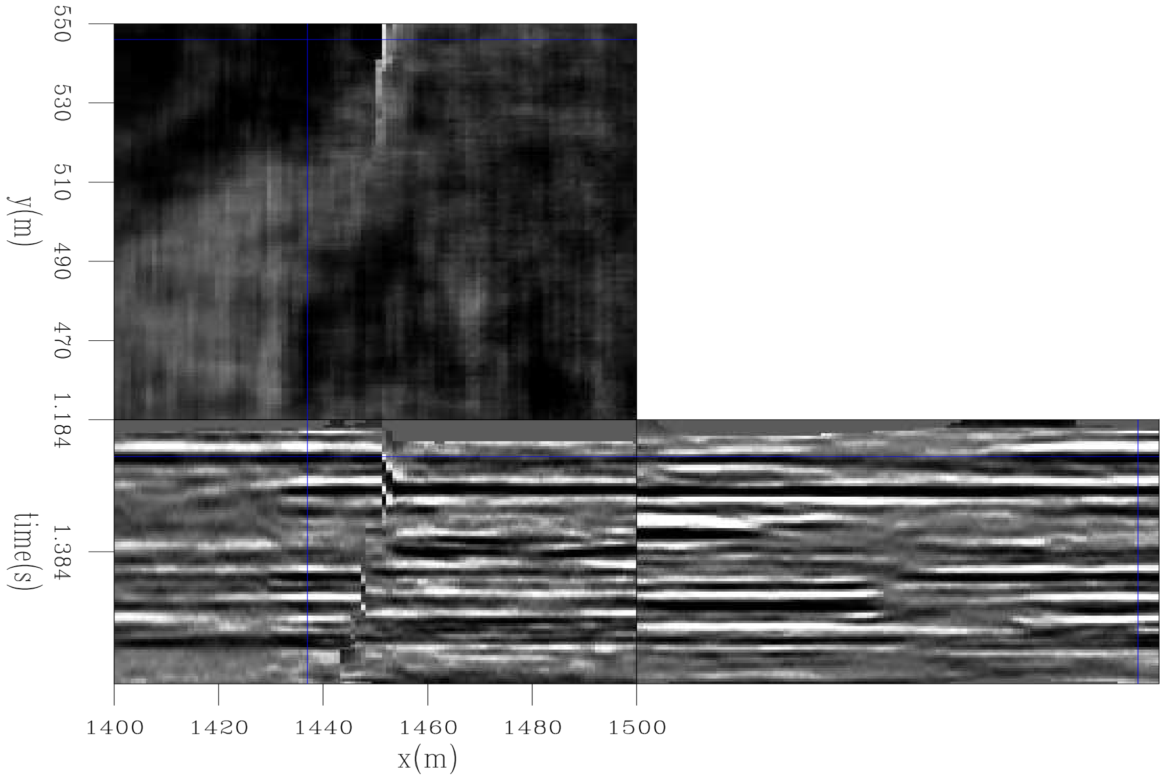

Figure 4: The result of flattening of Figure 3 jesse2-flat [ER]

We applied our IRLS flattening method to this data to get the result shown in Figure 4. The position of the fault is more obvious in this image because the data on either side were shifted by the flattening process.

The fault weight automatically created by this method is shown in Figure 5. It is, however, more insightful to overlay the fault weight on the original unflattened data as in Figure 6. Also, it is interesting to note that the fault appears to be segmented.

CONCLUSION

We presented an IRLS flattening approach able to flatten data cubes in the presence of laterally limited vertical faults. Our method uses iteratively re-weighted least-squares with the Geman-McClure function. The requirement that the faults terminate within the data is necessary so that dips can be summed around the faults in order to remove the structure.

For faults that do not terminate within the data cube, this method still may indicate their location. This is because dips estimated at faults tend to be more erratic than other dips away from the fault and are usually treated as outliers.

There are still several weaknesses and areas for improvement of this method. If the fault termination is close to the boundary of the data, this method has the unfortunate side effect

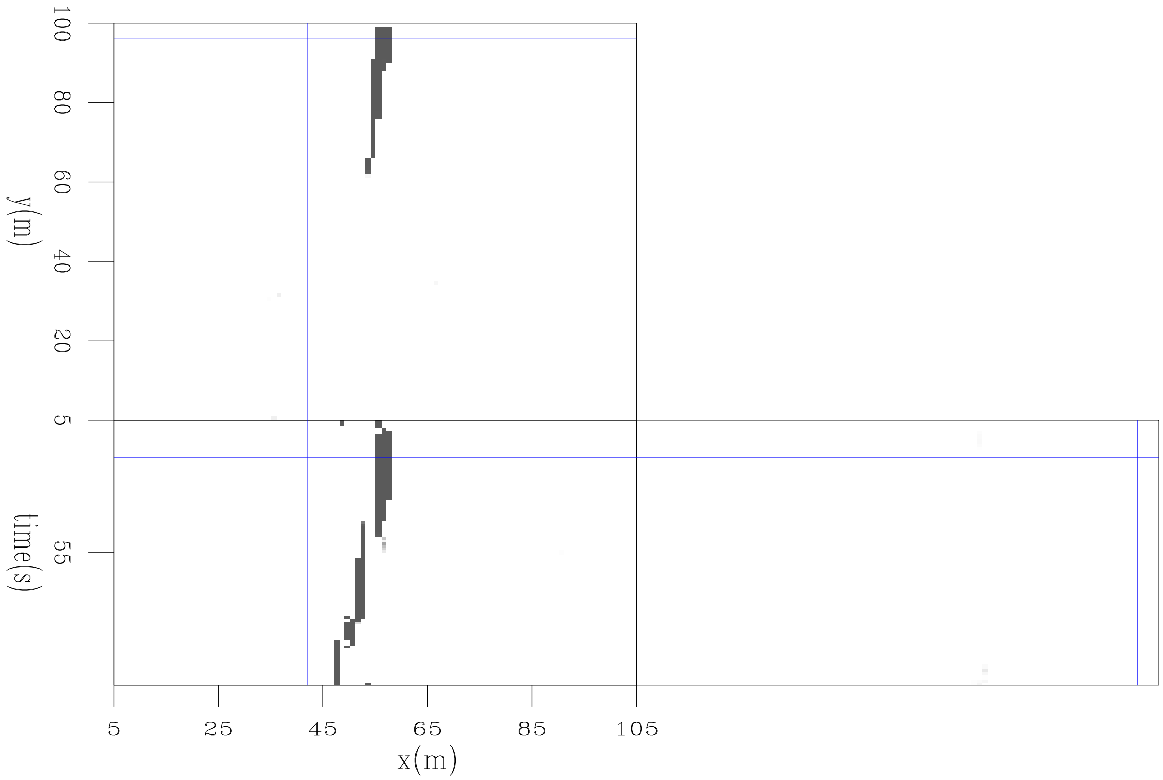

Figure 5: The fault weight automatically created by flattening Figure 3 jesse2-wt [ER]

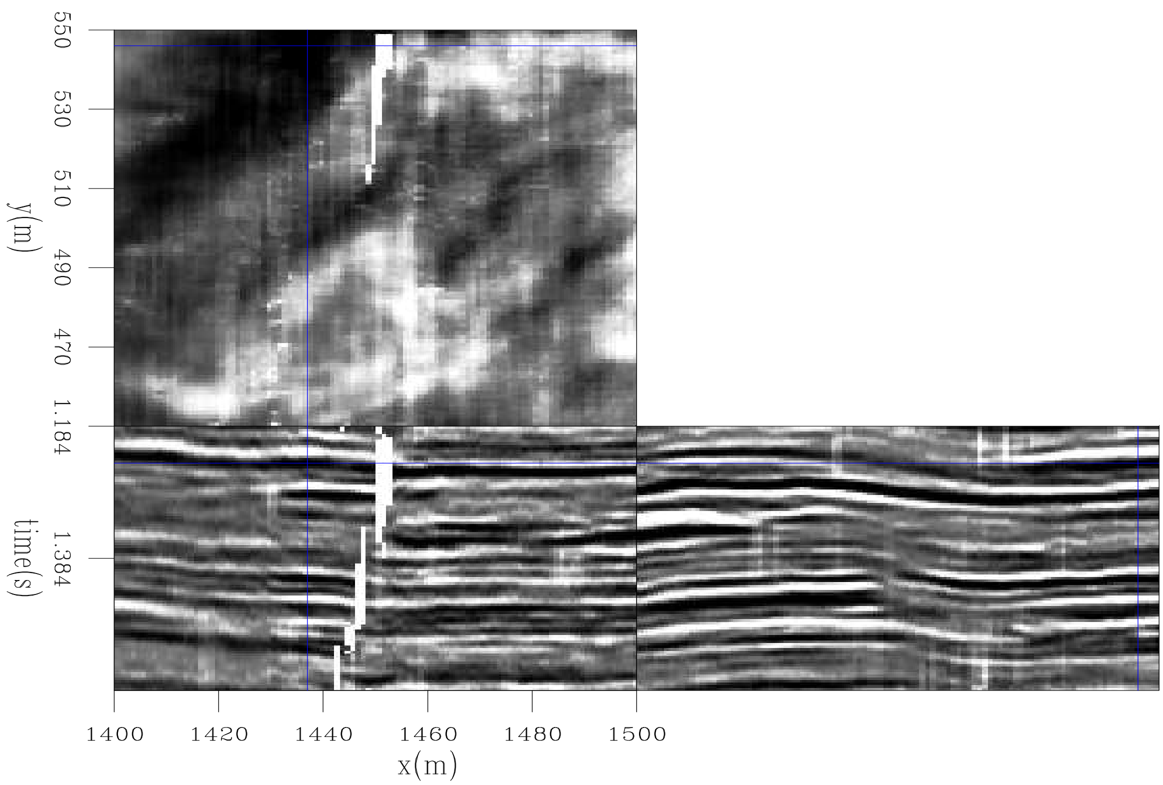

Figure 6: The input data with the jesse2-wt.dat [ER]

of connecting the fault to the data boundary, creating a separate fault block. It may also have the tendency to create false faults if there is significant noise. Lastly, we still may not have determined the best weight function.

ACKNOWLEDGMENT

We would like to thank Chevron for providing the data used in this paper.

REFERENCES

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

Claerbout, J., 1999, Geophysical estimation by example: Environmental soundings image enhancement: Stanford Exploration Project, `http://sepwww.stanford.edu/sep/prof/`.

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

Lomask, J., and Claerbout, J., 2002, Flattening without picking: SEP–**112**, 141–150.

Lomask, J., Guitton, A., Fomel, S., and Claerbout, J., 2005, Update on flattening without picking: SEP–**120**, 137–158.

Lomask, J., 2003, Flattening 3-D data cubes in complex geology: SEP–**113**, 247–260.

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