Short Note

AMO regularization: Effective approximate inverses for amplitude preservation

Robert G. Clapp[1]

INTRODUCTION

Amplitude preservation in imaging is becoming increasingly important. The irregularity of seismic data, particularly 3-D data, in both the model domain (in terms of subsurface position and reflection angle) and the data domain (in terms of midpoint, offset, and time) can have deleterious effects on amplitude behavior. There have been several general approaches to correct for this irregularity. The imaging problem is fundamentally an inverse problem, relating some model **m** to some data **d** through a linear operator **L**, which in this case is the adjoint of the migration operator. Ronen and Liner (2000); Duquet and Marfurt (1999); Prucha et al. (2000) cast the problem as such and then try to solve it with an iterative solver. These approaches have shown promise but are in many cases prohibitively expensive.

The problem is further complicated in that many migration algorithms assume the data is lying on regular mesh (downward continuation and finite difference schemes for example). Biondi and Vlad (2001) dealt with the problem of mapping the irregular data to a regular mesh for downward continuation migration. They set up an inverse problem relating the irregular input data to a regular model space. They regularized the problem by enforcing consistency between the various $(time, cmp_x, cmp_y)$ cubes. The consistency took two forms. In the first a simple difference between two adjacent inline offset cubes was minimized. In the second the difference was taken after transforming the cubes to the same offset through Azimuth Moveout (AMO) (Biondi et al., 1998). For efficiency the model was preconditioned with the inverse of the regularization operator (Fomel et al., 1997). Instead of solving the least squares inverse problem, the Hessian is approximated by a diagonal operator computed from a reference model (Claerbout and Nichols, 1994; Rickett, 2001; Clapp, 2003).

In this paper I examine and extend the work in Biondi and Vlad (2001). I show that approximating the inverse matrix with a simple diagonal operator is not sufficient. The resulting regular dataset has artificial amplitude anomalies. I replace the simple derivative operators with a filter that smooths along not only offset$_x$, but also offset$_y$. I conclude by discussing

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

how the problem can be effectively parallelized and the computational and storage challenges of various estimation schemes.

REVIEW

Most downward continuation methods require that the data lie on a regular mesh. To map the irregular recorded seismic data onto the regular mesh is a far from trivial exercise. A common approach in industry is to think of the problems in the same way we approach Kirchhoff migration, namely to loop over data space and spread into our regular model space. The spreading operation is governed by something like AMO (Biondi et al., 1998), which maps data from one offset vector to another. If we think of the AMO operator $\mathbf{T}$ as mapping from the regular model space $\mathbf{m}$ to the regular data space $\mathbf{d}$, our estimation procedure becomes,

$$\mathbf{m} = \mathbf{T}'\mathbf{d}. \tag{1}$$

This formulation suffers from all of the usual problems associated with applying an adjoint operation. We are spraying into a regular mesh, but the coverage is not regular. Areas with higher concentration of data traces will tend to map to artificially higher amplitudes in the model space. We can do some division by hit count to help minimize this effect but will still see some artifacts that come from approximating the inverse with an adjoint.

We can think of turning (1) into an inversion problem but, in addition to the high cost associated with the AMO operationm we face the same stability issues that setting up the migration problem as an inverse problem encounters. The null space of the imaging operator tends to put high frequency noise in the model space when cast as inverse problem.

Fomel (2001) suggested thinking of the problem more as a missing data problem. We can write the missing data problem in terms of the fitting goals

$$\begin{array}{rcl} \mathbf{d} & \approx & \mathbf{L}\mathbf{m} \\ \mathbf{0} & \approx & \epsilon\mathbf{A}\mathbf{m}, \end{array} \tag{2}$$

where $\mathbf{L}$ is a simple interpolation operator (nearest-neighbor, linear, etc) and the real work is done by the regularization operator $\mathbf{A}$ which describes the relationship between the irregular data and the regular sampled model.

We can speed up the convergence of (2) by preconditioning the model with $\mathbf{m} = \mathbf{A}^{-1}\mathbf{p} = \mathbf{B}\mathbf{p}$. Our new fitting goals become,

$$\begin{array}{rcl} \mathbf{d} & \approx & \mathbf{L}\mathbf{B}\mathbf{p} \\ \mathbf{0} & \approx & \epsilon\mathbf{p}. \end{array} \tag{3}$$

Biondi and Vlad (2001) suggested following the approach of Claerbout and Nichols (1994) and Rickett (2001). Instead of solving the inverse problem, they suggest filtering the adjoint solution with a diagonal operator. We obtain our filtering operator by first noting the least squares inverse of the interpolation problem,

$$\mathbf{m} = \mathbf{B}\left(\mathbf{B}'\mathbf{L}'\mathbf{L}\mathbf{B} + \epsilon^{2}\right)^{-1}\mathbf{B}'\mathbf{L}'\mathbf{d}. \tag{4}$$

We can think of equation (4) as filtering the adjoint solution with the matrix $\mathbf{W}^{-1}$ where,

$$\mathbf{W} = \mathbf{B}'\mathbf{L}'\mathbf{L}\mathbf{B} + \epsilon^{2}. \tag{5}$$

The weighting matrix is (np x np) where np is the size of our preconditioned model space. This matrix will be generally diagonally dominant. We can think of estimating a diagonal filtering operator $\mathbf{W}_{\rm diag}$ by using a reference model (in preconditioned model space) $\mathbf{p}$ and applying

$$\mathbf{W}_{\rm diag} = \frac{\mathbf{diag}\left[(\mathbf{B}'\mathbf{L}'\mathbf{L}\mathbf{B} + \epsilon^{2})\mathbf{p_{ref}}\right]}{\mathbf{diag}\left(\mathbf{p_{ref}}\right)}. \tag{6}$$

We can then get an estimate of our model through

$$\mathbf{m} = \mathbf{B}\mathbf{W}^{-1}\mathbf{B}'\mathbf{L}'\mathbf{d}. \tag{7}$$

Regularization

In the formulation above our model quality is now greatly determined by our choice of regularization operator. Biondi and Vlad (2001) implemented two different approach. The first was simply applying a derivative filter, $1 \quad -\rho$ along the offset axis. The ρ controls the length of the smoother. For symmetry we can cascade a left derivative $\mathbf{D_l}$ followed by a right derivative $\mathbf{D_r}$ for the combined regularization operator $\mathbf{A} = \mathbf{D_r}\mathbf{D_l}$.

Biondo and Vlad's (2001) second choice is more interesting. Vlad and Biondi (2001) describes a very fast implementation of AMO (based on the DMO formulation in the logstretch domain of Zhou et al. (1996)) on a regularly sampled mesh. They suggested instead of minimizing the difference between two offset cubes, to minimize the difference between the two cubes continued to the same offset through AMO. If we now imagine the filter operating on (t, cmp_x, cmp_y) cubes, the right derivative operation becomes $\mathbf{I} \;\; - \rho\mathbf{T_{h_{i+1,i}}}$ where $\mathbf{T_{h_{i,i+1}}}$ is the AMO transformation of the (t, cmp_x, cmp_y) at offset $i+1$ to offset i.

COMPLICATIONS

The procedure described in the last section has several problems when applied to a large 3-D dataset. In this section I cover some of these; issues, solutions, and compromises that need to be made for a practical implementation.

Approximate inverses

The approximation of equation (7) at first gives a visually appealing result. But if we take a closer look, we see that it isn't as close to the true inverse as we might hope. To see how, let's look at a simpler problem. The circles in Figure 1 show a series of irregular data points subsampled from the solid line curve. The dotted lines show the solution when applying equation (7). The dashed line represents the solution using fitting goals (3). Note how the approximate solution approach has the correct low frequency shape but varies significantly from the full inverse solution. With better choices of **p**, or additional diagonals (Guitton, 2003), a better solution might be possible, but the potential is limited. The full matrix can not be adequately described by the limited description we are allowing.

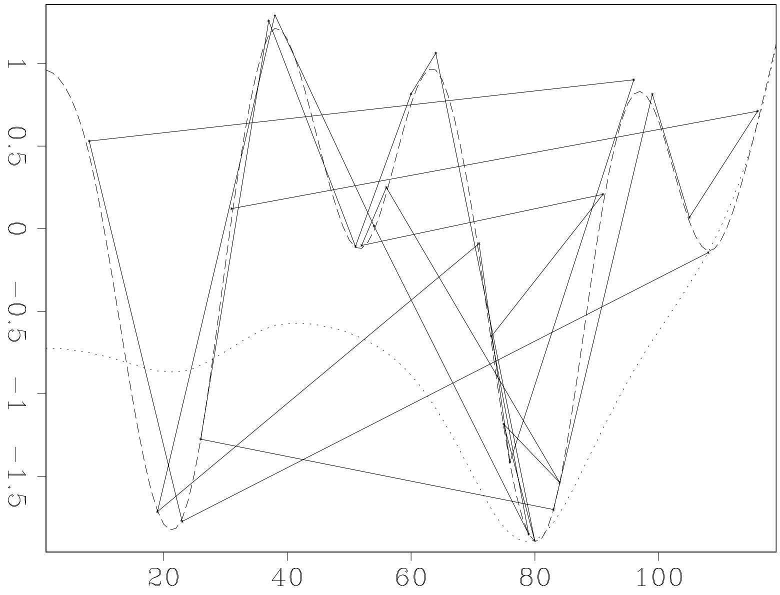

Figure 1: The solid line represents the input signal. The circles represent our data points. The dashed line shows the solution with fitting goals (3). The dotted lines show the solution using the approximate method in equation (7). bob1-miss [ER]

We can see the same effect in our regularization problem. Figure 2 shows the regularization result of a small portion of a 3-D land dataset. Figure 3 shows the result of doing five steps of conjugate gradient solving the fitting goals in (3). Note how we have a smoother, more believable, amplitude behavior as a function of midpoint. Unfortunately, the effect of the approximate solution translates directly to an effect on the amplitudes in our migration. Figure 4 is the result of migrating a single line from the 3-D dataset regularized by the method described in Biondi and Vlad (2001). Note the stripes of high and low amplitude indicated by ‘A’ and ‘B’. Wavefront healing helps minimize the effect at depth but there is still a noticable effect on the amplitude.

Solving the full inverse introduces its own problems. First, we have now significantly increased the cost. The approximate solution (7) required calling both **L** and **B** three times. Even using a minimal number of iteration (3-5) increases the cost by a factor of two or three. In addition, we have significantly increased our disk space requirement. If we set up the inverse problem shown in fitting goals (3), we now must store six copies of our model space. We are quickly approaching the point of impracticality even for a small dataset.

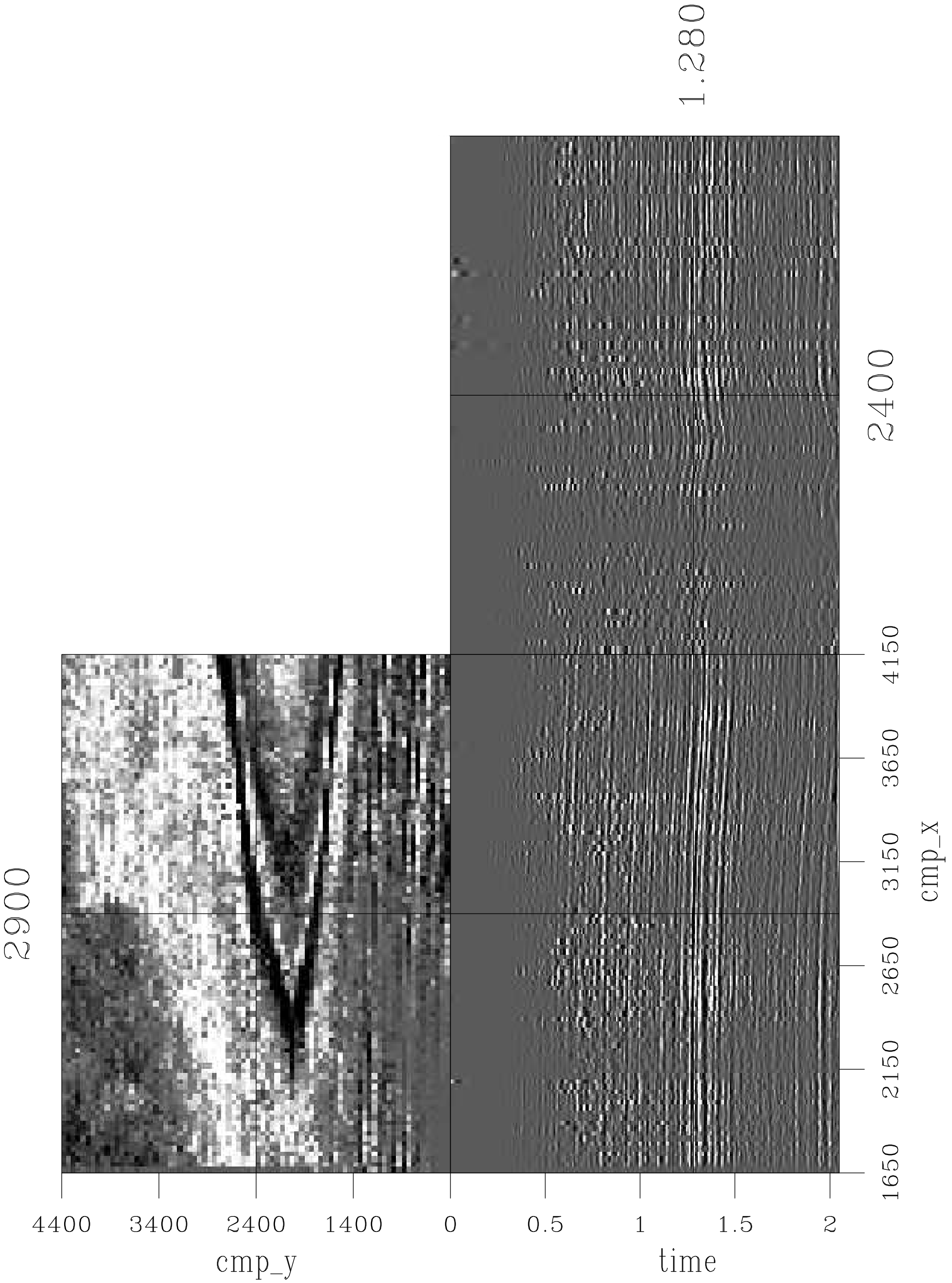

Figure 2: A portion of the regularized data, (t, cmp_x, cmp_y) cube, estimated using the approximate solution in equation (7). Note the dimming and brightening due to the irregular sampling of the input data. bob1-approx [CR]

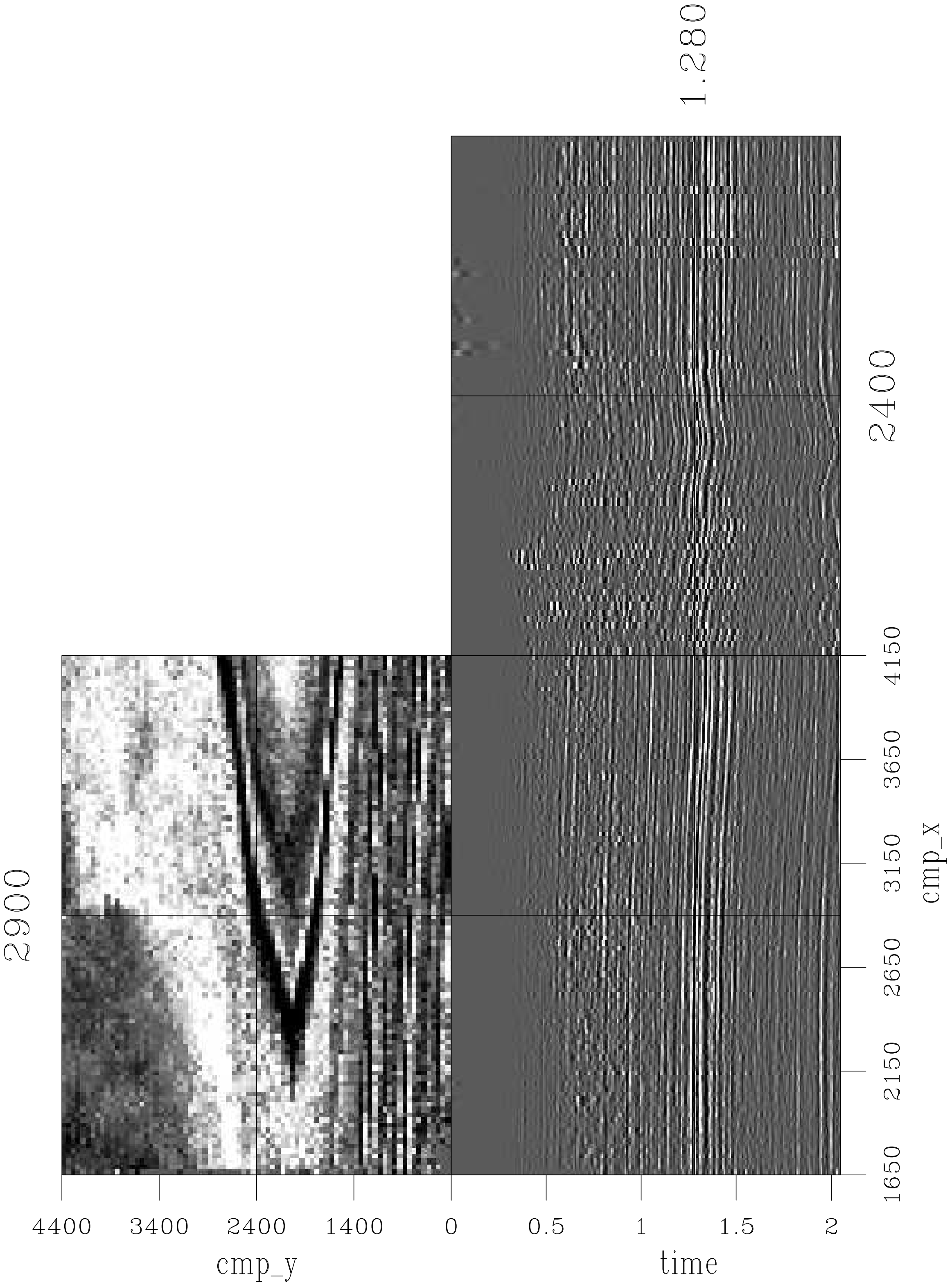

Figure 3: A portion of the regularized data, (t, cmp_x, cmp_y) cube, estimated by five conjugate gradient iterations using (3). Note how the improved amplitude continuity. bob1-inverse [CR]

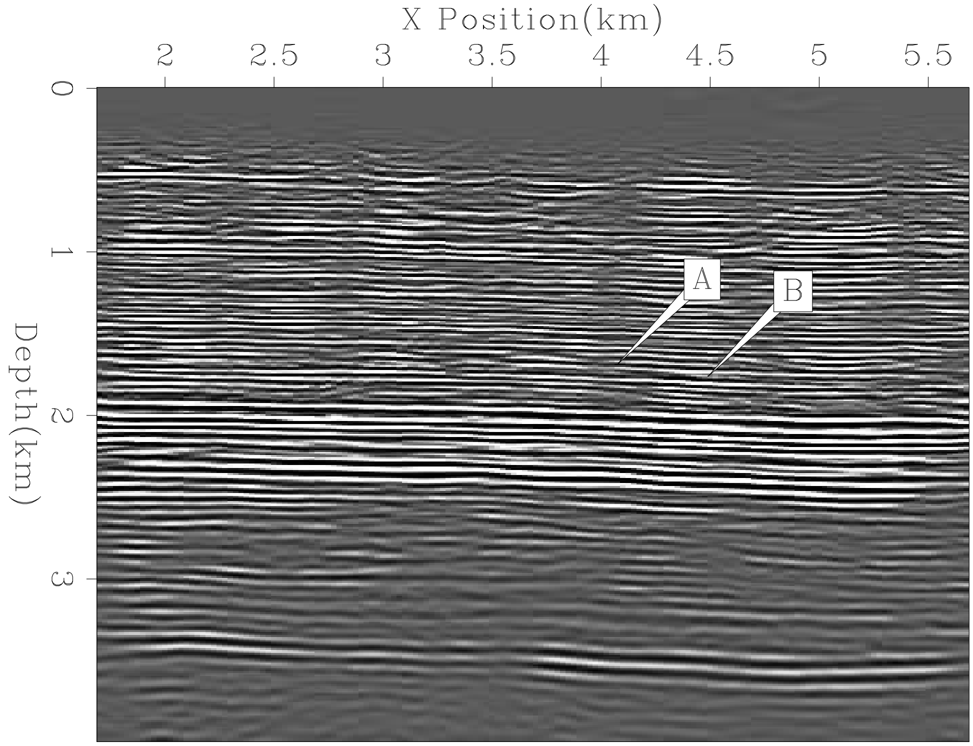

Figure 4: Migration of one line of dataset regularized with the approximation solution (7). Note the difference in amplitude at 'A' and 'B' caused by the approximate solution. bob1-mig [NR]

Dimensionality

A 3-D reflection dataset resides in a five dimensional space. Typically we describe this space in terms of $(t, \mathbf{s}, \mathbf{g})$ or $(t, \mathbf{h}, \mathbf{cmp})$ where t is time and $\mathbf{s}, \mathbf{g}, \mathbf{cmp}, \mathbf{h}$ are the source, receiver, midpoint, and offset vectors. In Biondi and Vlad (2001) the dimensionality of the dataset was decreased by one by describing offset by a scalar rather than a vector. This is far from an ideal solution, especially in the case of a land dataset and/or data over complicated geology. In both cases the earth being sampled at different azimuths can vary significantly. By stacking we are making an implicit assumption that there isn't any variation (or at least significant variation) due to azimuth. This can affect both our amplitudes and our ability to accurately estimate the model velocity (Clapp and Biondi, 1995).

If we use a five dimensional model space we must modify our estimation procedure. The solution is to perform individual estimations at different h_y or azimuths. Solving independent problems is not generally a workable solution. First we aren't imposing any smoothness over h_y or azimuth, something that we know should physically exist. Our estimation procedure is likely to produce an answer far from smooth over the added axes. Figure 5 and Figure 6 show fold maps for a portion of a 3-D land and marine datasets. The left panel shows the cmp_x, cmp_y, h_x cube, the latter the cmp_x, cmp_y, h_y cube at the same midpoint location. Note how in both cases the coverage varies significantly as a function of h_y. Take the marine case for

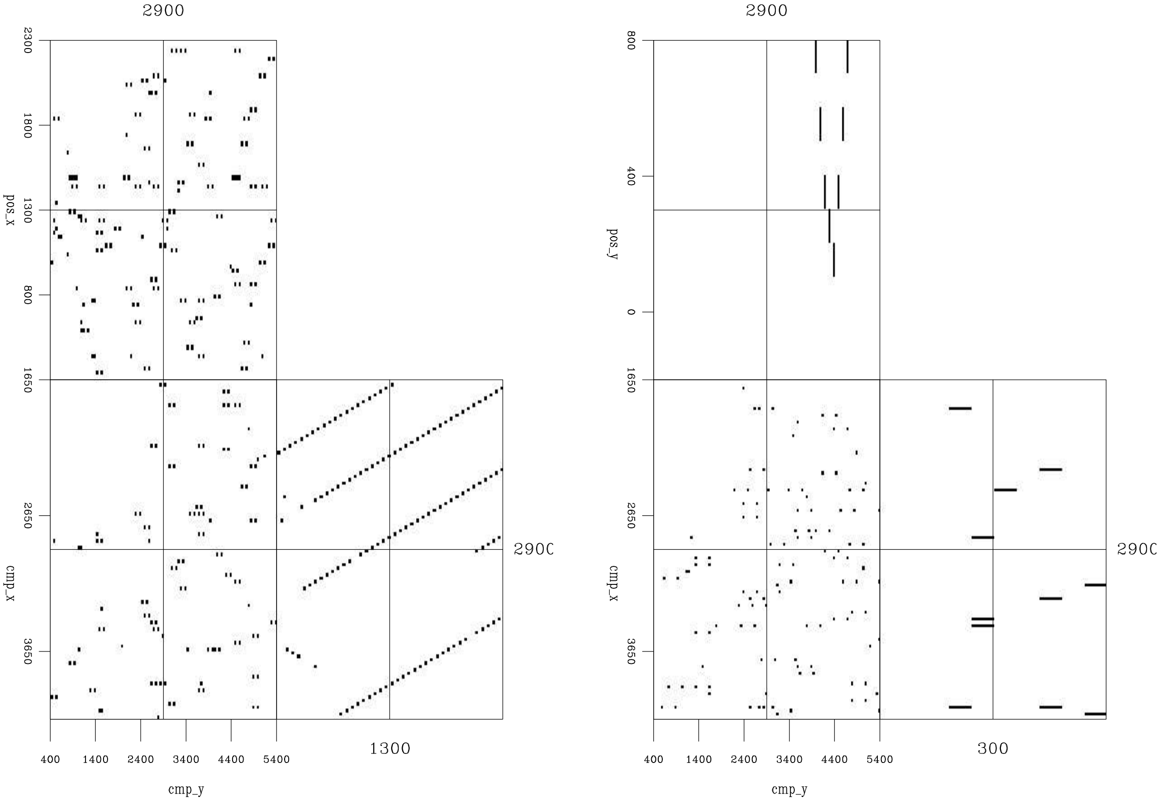

Figure 5: The fold from a portion of a 3-D land dataset. The left panel is a subset at a constant h_y. Three panels from the subset are shown. The right panel is a subset at a constant h_x. Both panels show the same cmp_x, cmp_y location. bob1-fold.land [CR,M]

example. Standard acquisitions techniques would lead to cmp_y (cross-line direction) locations to be banded along different h_y locations (caused by the multiple towed cables) and few large h_x at small h_y (due to cable feathering).

Data size

The estimation problem we have set up requires a model space larger than we will typically use in migration. As mentioned above, traditional implementation of AMO (Biondi et al., 1998) works like Kirchhoff migration. We define our model space (as sparse or as dense as we wish) and there sum in nearby traces with appropriate weights. The AMO procedure can be used as a fairly intelligent partial stack. By implementing the AMO as a regularization operator we are asking **L** to map the trace from the irregular data space to the the regular space that our model exists on. If we have too coarse of a sampling in our model space we end up mapping numerous data points to each model point. If we think about data's behavior as a function of offset (fairly variable even after NMO) the danger of making too large of bins becomes apparent.

The problem is that our full model space is enormous. A small to mid-size dataset might

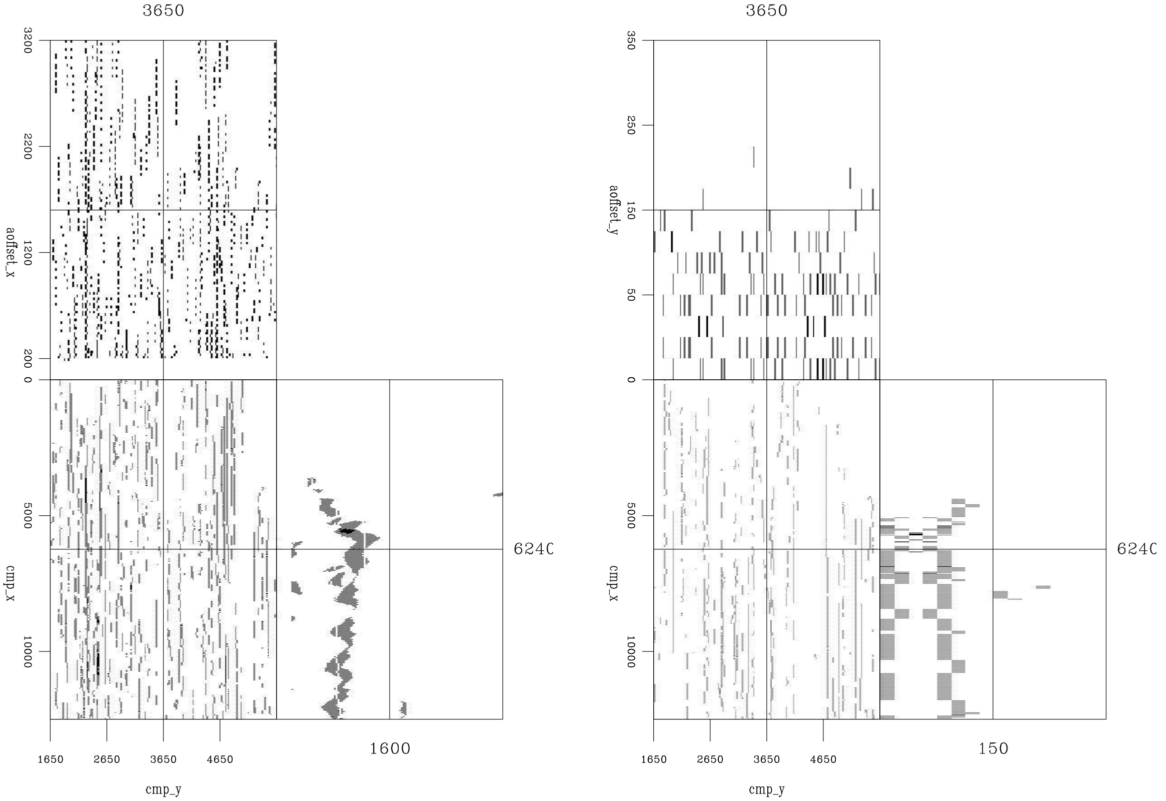

Figure 6: The fold from a portion of a 3-D marine dataset. The left panel is a subset at a constant h_y. Three panels from the subset are shown. The right panel is a subset at a constant h_x. Both panels show the same cmp_x, cmp_y locations. bob1-fold.elf [CR,M]

have 1500 time samples, 1000 cmp_x, 1000 cmp_y, $128 h_x$, and require 20 h_y. That amounts 15 TBs, exceeding the entire storage capacity of SEP. Even a small portion of the dataset (500 cmp_x, 200 cmp_y) will still consume 1.5 TB.

Parallelization and Regularization

The standard Beowulf cluster, and SEP's cluster, consists of many single or dual processor nodes with communication between nodes having fairly large latency. As a result we want limited communication and as coarse a grain scheme as possible. The AMO operation (Vlad and Biondi, 2001) consists of:

$\mathbf{S}_{\log}$ log stretch of time axis,

$\mathbf{F_t}$ FFT of the stretched time axis,

$\mathbf{F_{xy}}$ FFT the cmp_x and cmp_y axes,

C complex multiplication

$\mathbf{F_{xy}^{-1}} = \mathbf{F'_{xy}}$ FFT the cmp_x and cmp_y axes,

$\mathbf{F_{t}^{-1}} = \mathbf{F'_{t}}$ FFT of the stretched time axis, and

$\mathbf{S}_{\log}^{-1}$ inverse log stretch of time axis.

Our regularization operator is not limited to the AMO operator. Our choice of regularization also effects our ability to parallelize the problem. Ideally we would like to have a regularization operator that assessed continuity over all axes (a non-stationary Prediction Error Filter for example), but that would either eliminate our ability to parallelize the problem or require massive communication between the nodes (to pass the boundary areas over the axes we parallelized over).

If we sacrifice regularizing along the time axis the problem becomes more manageable. We can redefine our data as

$$\mathbf{d}_{\mathrm{new}} = \mathbf{F_t}\mathbf{S}_{\log}\mathbf{d}, \tag{8}$$

along the first axis.

We can now split the data along the time axis and regularize along any of the remaining axes. For this paper I chose to only regularize along the offset axes. A 4-D prediction error filter would be preferable, but would require simultaneously infilling and estimating the filter or some ad-hoc scheme that is beyond the scope of this paper.

Regularizing only over offset also allows additional cost savings. We can pull out the $\mathbf{F_{xy}}$ operator outside our filtering operation. For the cascaded derivative operation used in Biondi and Vlad (2001) we save the cost of six $\mathbf{F_{xy}}$ per iteration step (approximately 67% reduction in cost).

Our expanded model space (h_y axis) requires a new regularization scheme. Two obvious choices come to mind. The first is to cascade derivative regularization along the h_y axis. Our new regularization operator becomes

$$\mathbf{A} = \mathbf{F_{xy}}\mathbf{D_{hy,r}}\mathbf{D_{hy,l}}\mathbf{D_{hx,l}}\mathbf{D_{hx,r}}, \tag{9}$$

where $\mathbf{D_{ha,b}}$ is taking the derivative (after transforming the (t, cmp_x, cmp_y) cube) in the b direction along the a axis. The other approach is to use some arbitrary filter, such as a factored Laplacion (Claerbout, 1999). I tested both methodologies. The first approach does not have a completely symmetric impulse response (Fomel, 2001), but proved to converge in fewer iterations.

On even a small problem the current formulation is still problematic on SEP's current architecture. Having to store six copies of the model exceed our node's disk capacity even after splitting the data along the time axes. The final simplification is instead of solving a single global inversion problem to solve for each frequency independently. This final simplification makes the problem manageable, but at a price. We are doing a low number of conjugate gradient iterations, therefore our solution step size (and direction after the first iteration) is going to be different for the global and the individual local problems.

CONCLUSIONS

Data regularization is an important problem when using migration methods that rely on the data being on a regular mesh. Traditional methods that apply the adjoint of a continuation operation such as AMO can lead to poor amplitude information in the regularized (and later migrated) cube. By setting up the regularization problem as inverse problem the amplitudes in the regular model space are significantly improved. The problem can be made computationally acceptable by intelligent parallelization and regularization choices.

REFERENCES

Biondi, B., and Vlad, I., 2001, Amplitude preserving prestack imaging of irregularly sampled 3-D data: SEP–**110**, 1–18.

Biondi, B., Fomel, S., and Chemingui, N., 1998, Azimuth moveout for 3-D prestack imaging: Geophysics, **63**, no. 2, 574–588.

Claerbout, J., and Nichols, D., 1994, Spectral preconditioning: SEP–**82**, 183–186.

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

Clapp, R. G., and Biondi, B., 1995, Multi-azimuth velocity estimation: SEP–**84**, 75–88.

Clapp, M. L., 2003, Illumination compensation: Model space weighting vs. regularized inversion: SEP–**113**, 369–378.

Duquet, B., and Marfurt, K. J., 1999, Filtering coherent noise during prestack depth migration: Geophysics, **64**, no. 4, 1054–1066.

Fomel, S., Clapp, R., and Claerbout, J., 1997, Missing data interpolation by recursive filter preconditioning: SEP–**95**, 15–25.

Fomel, S., 2001, Three-dimensional seismic data regularization: Ph.D. thesis, Stanford University.

Guitton, A., 2003, Amplitude and kinematic corrections of migrated images for non-unitary imaging operators: SEP–**113**, 349–362.

Prucha, M. L., Clapp, R. G., and Biondi, B., 2000, Seismic image regularization in the reflection angle domain: SEP–**103**, 109–119.

Rickett, J., 2001, Model-space vs data-space normalization for finite-frequency depth migration: SEP–**108**, 81–90.

Ronen, S., and Liner, C. L., 2000, Least-squares DMO and migration: Geophysics, **65**, no. 5, 1364–1371.

Vlad, I., and Biondi, B., 2001, Effective AMO implementation in the log-stretch, frequency-wavenumber domain: SEP–**110**, 63–70.

Zhou, B., Mason, I. M., and Greenhalgh, S. A., 1996, An accurate formulation of log-stretch dip moveout in the frequency-wavenumber domain: Geophysics, **61**, no. 3, 17–23.