Stanford Exploration Project, Report 120, May 3, 2005, pages 187–196

Data regularization: inversion with azimuth move-out

Robert G. Clapp[1]

ABSTRACT

Data regularization is cast as a least-squares inversion problem. The model space is a five-dimensional (t,cmpx, cmpy, hx, hy) hypercube. The regularization minimizes the difference between various (t, cmpx, cmpy) cubes by applying a filter that acts in (hx,hy) plane. Azimuth Move-out is used transform the cubes to the same (hx,hy) before applying the filter. The methodology is made efficient by a Fourier-domain implementation, and preconditioning the problem. The methodology, along with two approximations is demonstrated on 3-D dataset from the North Sea. The inversion result proves superior at a reasonable cost.

INTRODUCTION

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) cause imaging problems. The most effectively family of multiple removal methods, SRME (Verschuur et al., 1992) rely on data regularity. The standard marine acquisition technology results in empty bins in the cross-line direction.

Migration methods also desire a greater level of regularity than is often present in seismic surveys. There are two general approaches to deal with this problem. One approach is to treat the imaging problem as inverse problem. 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, and rely on an accurate subsurface velocity model.

Another approach is to try to regularize the data. AMO provides an effective regularization tool (Biondi et al., 1998). AMO is generally applied as an adjoint to create a more regularized volume. These regularized volumes still often contain in 'acquisition footprint' or more subtle amplitude effects. Chemingui (1999) used a logstretch transform to make the AMO operator stationary in time. He then cast the regularization problem as a frequency by frequency inversion problem using a Kirchoff-style AMO operator. He showed that acquisition footprint could be significantly reduced. The downside of this approach is relatively high cost of Kirchoff implementation and the difficulty with a frequency-by-frequency approach to a global inversion problem.

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

Biondi and Vlad (2001) built on the work of Fomel (2001). 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 (t,cmpx,cmpy) cubes. The consistency took two forms. In the first a simple difference between two adjacent in-line offset cubes was minimized. In the second the difference was taken after transforming the cubes to the same offset AMO. 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, 2003a).

In this paper I examine and extend the work in Biondi and Vlad (2001) and Clapp (2003b). I compare the result of using the AMO operator as an adjoint, using a diagonal operate to approximate the Hessian, and doing a full inverse. I show that applying the inverse proves to be significantly better. In the paper I begin the paper by going the general methodology, I then discuss how to implement it effectively on a Beowulf cluster.

ADJOINT IMPLEMENTATION

To map the irregular recorded seismic data onto the regular mesh is a far from trivial. 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 can be 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 irregular data space $\mathbf{d}$, our estimation procedure becomes,

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

The wavenumber domain AMO operator works on a regular sampled cube, so the problem is more complicated. We first must map the data to a regular sampled space by applying the interpolation operator $\mathbf{L}'$. The regular sampled cube $\mathbf{s}$ is now a full five dimensional volume $(t, \mathrm{cmpx}, \mathrm{cmpy}, \mathrm{hx}, \mathrm{hy})$. We can produce the model $\mathbf{m}$ at a given (hx,hy) by summing nearby cubes (t, cmpx, cmpy) that have transformed to our desired (hx,hy) through AMO. To write this in a mathematical form we need to make some definitions. We will define ihx and ihy as the offset indicies of the expanded space $\mathbf{s}$. These indicies correspond to the half-offset hx and hy. The output space, $\mathbf{m}$, is defined as a coralary ihx' and ihy‘ which also correspond to hx and hy. The notation $\mathbf{m}(\mathbf{ihx}', \mathbf{ihy}')$ correspond to the 3-D subcube (t,cmpx,cmpy) at the given ihx' and ihy'. Finally $\mathbf{T}_{\mathbf{x_1} \Rightarrow \mathbf{x_2}}$ refers to transforming the cube through AMO from the offset vector $\mathbf{x_1}$ to $\mathbf{x_2}$, nx and ny is the region in sampling of $\mathbf{s}$ that we wish to sum over; and dhx and dhy is the sampling of the cube in hx and hy respectively. We obtain

$$\mathbf{m}(\mathbf{ihx}', \mathbf{ihy}') = \sum_{\mathbf{iy}=-\mathbf{ny}}^{\mathbf{ny}} \sum_{\mathbf{ix}=-\mathbf{nx}}^{\mathbf{nx}} \mathbf{T}_{(\mathrm{hx}+\mathrm{ixdhx},\mathrm{hy}+\mathrm{iydhy}) \Rightarrow (\mathrm{hx},\mathrm{hy})} \mathbf{s}(\mathbf{ix}+\mathbf{ihx}, \mathbf{iy}+\mathbf{ihy}). \tag{2}$$

In we to write our regularization problem in the form of equation (2), **S** is a spraying operation where the columns of the matrix are defined by equation (1). We then obtain or model by applying

$$\mathbf{m} = \mathbf{S}'\mathbf{L}'\mathbf{d}. \tag{3}$$

The formulation suffers from all of the usual problems associated with applying an adjoint operation. We are spraying into a regular mesh, but the data is not regular. Areas with higher concentration of data traces will tend to map to artificially higher amplitudes in the model space. In the Kirchoff formulation we can do some division by hit count to help minimize this effect. Because we are operating in the wave number domain we can't normalize by something as simple as hit count. We can accomplish something similar by following the approach of Claerbout and Nichols (1994) and Rickett (2001). We approximate the Hessian of the least squares solution,

$$\mathbf{m} = \left(\mathbf{S}\mathbf{L}'\mathbf{L}\mathbf{S}\right)^{-\mathbf{1}}\mathbf{S}'\mathbf{L}'\mathbf{d}, \tag{4}$$

by the diagonal operator **W**. We form **W** by

$$\mathbf{W}^{-\mathbf{1}} = \text{diag}\left[\left(\mathbf{S}'\mathbf{L}'\mathbf{L}\mathbf{S}\mathbf{1} + \alpha\right)\right], \tag{5}$$

where **1** is a vector of 1s, α is a stabilization term, and diag[] map the vector to the diagonal of the matrix. We scale our adjoint solution by **W** obtaining

$$\mathbf{m} = \mathbf{W}\mathbf{S}'\mathbf{L}'\mathbf{d}. \tag{6}$$

Implementation

There are several issues that must be considered when implementing AMO in this form. The large volume of data that we are dealing with means that the problem must be parallelized. The problem can be parallelized in several different ways. While it is possible to split in the (cmpx, cmpy) plane, boundary effect are a concern because the operator is applied in the wave-number domain. Because the operator is applied in the frequency domain parallelizing over frequency seems a natural choice. The problem with dividing the problem along the frequency axis is that the intermediate space **s** can become enormous, even for fairly small datasets, which would require some level of patching along other axes. In addition it requires a troubling transpose. The input data has its inner axis as time, while we want the outer axis to be frequency. For multi-gigabyte this can be quite time consuming. For this reason I chose to parallelize offset. Each process is assigned an output (hx,hy) range. It takes the input that range plus the additional summation range implied by equation (2).

The parallel job is controlled by the library described in Clapp (2005). Each node receives a SEP3D (Biondi et al., 1996) volume corresponding to its output space and the summation region implied by (2). The serial code first NMOs, log-stretches, and converts to frequency its data volume. The data volume is transposed and equation (6) is applied. The regularized frequency slices are transposed, inverse Fourier transformed, and has inverse NMO applied to it. Finally the data is recombined to form the regularized volume.

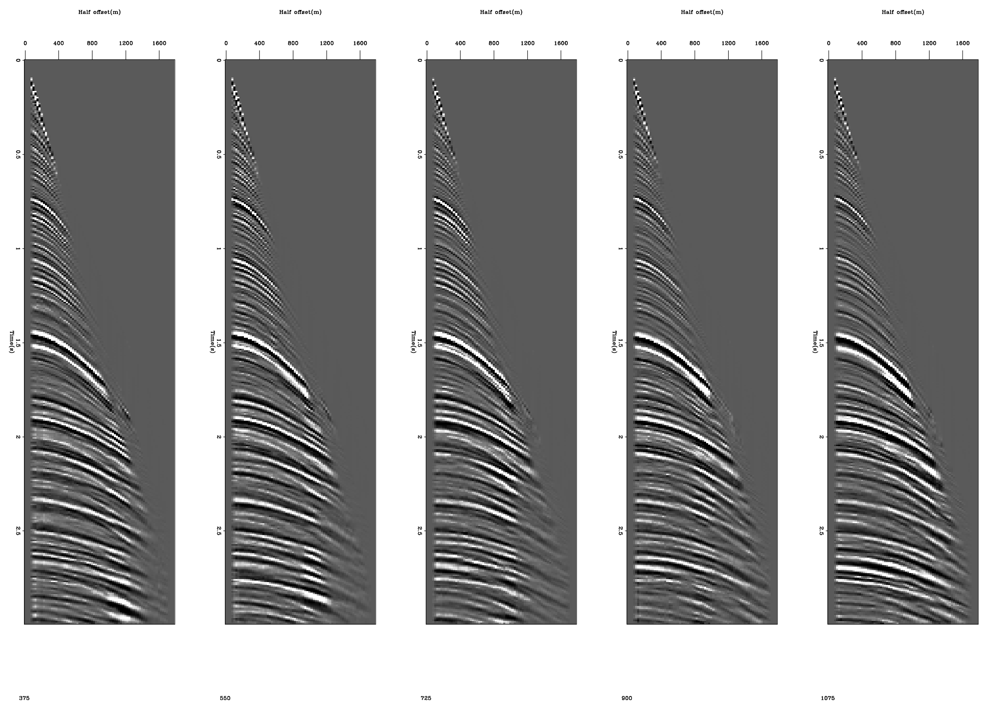

Figure 1: Five CMP gathers from the regularized dataset. Note the overall smoothness of the result. bob2-adj.cmp [CR]

Results

The methodology was applied on a 3-D marine dataset from the North Sea. The output model space was limited to hx = 0, *ny* of equation 2 was set to the entire hy range, and *nx* was set to 1. Figure 1 shows five CMP gathers from a relatively simple geologic portion of the dataset. The overall result is satisfactory and event continuity is quite good. Note the brightening and dimming as a function of offset in the lower portion of the second and third panels. The correction factor, equation (5) does not sufficiently emulate the inverse Hessian to remove all of acquisition artifacts.

REGULARIZING OVER OFFSET

There is a notable drawback from the approach described above. The operator **S** can be quite costly, We are doing nx*ny AMO transforms for every output (hx,hy). If we are only interested in a common azimuth dataset hy = 0, the cost is acceptable as long as *ny* is fairly small. If we want any cross line offset output offset the cost isn't acceptable. In addition **S** is a modified version (because of the AMO transform) of a small 2-D box car filter. If you desire additional smoothness in the in-line offset direction (to suppress amplitude variations) we must try a different approach.

Biondi and Vlad (2001) proposed reducing the dimensionality of the problem by ignoring the azimuth direction. They added a smoothness constraint to the problem by applying a Leaky derivative **D** operator between AMO transformed (t,cmpx,cmpy) cubes setting up the minimization,

$$Q(\mathbf{m}) = ||\mathbf{d} - \mathbf{L}\mathbf{m}||^2 + \epsilon^2 ||\mathbf{D}\mathbf{m}||^2, \tag{7}$$

where ϵ controls the weighting between the two goals. They preconditioned the problem with the inverse of **D**, leaky integration between AMO transformed cubes **C** and applied the same Hessian approximation to obtain the approximation

$$\mathbf{m} = \mathbf{C}\mathbf{W}\mathbf{C}'\mathbf{L}'\mathbf{d}, \tag{8}$$

where **C** is the inverse of **D** and

$$\mathbf{W}^{-1} = \mathrm{diag}\left[\mathbf{C}'\mathbf{L}'\mathbf{L}\mathbf{C}\mathbf{1} + \epsilon^2\right]. \tag{9}$$

Clapp (2003b) noted that ignoring the azimuth removed some of the advantage of using the AMO operator and suggested that **C** should be applying polynomial division with a 2-D filter operating in the the (hx,hy) plane. For this exercise I chose a small helical derivative for my 2-D filter (Claerbout, 1999).

Implementation

Implementing this method proves to be more problematic then the adjoint case. The added difficulty is caused by the 2-D filter. In order to parallelize over offset we would have to have significant inter-processor communication. This is problematic from both stability, we must rely all of the nodes remaining up, and efficiency, both the cost of sending the data and the delay caused by machine A needing data from B which needs data from C. As a result I decided to parallelize over frequency. As mentioned before this also has drawbacks. An entire (cmpx,cmpy,hx,hy) hypercube can not be held in memory for large problems, so we must do patching along some other axes.

The necessary transposes (to move from an inner time axis, to an outer frequency and back) complicate matters. The data are initially broken along the trace axis. The local datasets are NMOed, FFTed, and transposed. The transposed data is then recombined with frequency the outer axis. The procedure is significantly faster then performing the transpose on a single machine where disk IO dominates. By using multiple nodes which can each do the transpose in-core, or nearly core, the entire processing drops to minimally more than distributing and collecting the data.

Frequency blocks are then distributed to the nodes and equation (8) is applied. The data is then collected and re-split along the cmpx axis. The new regularized frequency slices are transposed, inverse FFTed, inverse NMOed, and recombined to form the output volume.

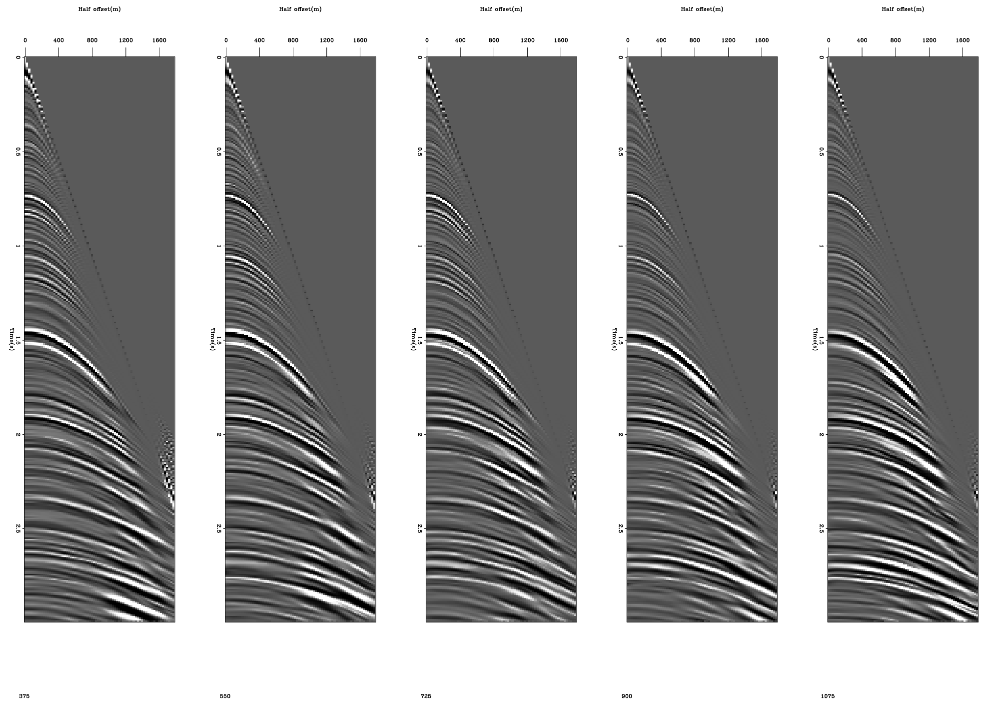

Figure 2: The same five CMP gathers from Figure 1 estimated by applying equation (8). Note the reduced frequency content in the image. bob2-approx0.cmp [CR]

Results

Equation 8 was applied to the 3-D marine dataset. Figure 2 shows the same five CMP gathers as Figure 1. The CMP gathers were created by AMO stacking to zero hy,

$$\mathbf{m}(\mathrm{hy}=0,\mathrm{ihx}) = \sum_{\mathrm{hy}} \mathbf{T}_{(\mathrm{hx},\mathrm{hy}+\mathrm{iydhy})\Rightarrow(\mathrm{hx},0)} \mathbf{m}(\mathbf{ihy},\mathbf{ihx}). \tag{10}$$

The resulting CMP gathers show a reduction in the dimming and brightening as function of offset but also show notable reduction in frequency content. The approximation is more economical in forming a full 5-D space then the adjoint formulation and produces a greater continuity in the offset plane. On the other hand, noticeable brightening and dimming can still be seen. In addition transfer times, due to the transposes, dominate the processing time.

INVERSION

Instead of approximating the Hessian with a diagonal matrix we can attempt to estimate the least squares inverse using a conjugate gradient solver. The model is preconditioned by using polynomial division to apply the helical derivative and the new preconditioned variable $\mathbf{p}$ is estimated through

$$Q(\mathbf{p}) = ||\mathbf{d} - \mathbf{LCp}||^2 + \epsilon^2 ||\mathbf{p}||^2, \tag{11}$$

where $\mathbf{m} = \mathbf{Cp}$.

Implementation

The implementation follows the same form as the approximate solution. The data is converted to frequency and distributed to the nodes. The inversion is done on the distributed files. Vector operations (scale, add, dot product) are calculated with MPI based routines. After the model has been estimated the data is redistributed for conversion back to time.

Results

Ten conjugate gradient iterations with $\epsilon = 0$ were applied. Figure 3 shows the same five CMP gathers show in the previous two sections. The frequency content is restored compared to the result seen in Figure 2. In addition the brightening and dimming seen in the result of the previous two methods are almost completely removed.

CONCLUSION

Three different methods to regularize seismic data with a wavenumber based AMO operator are described. The adjoint implementation is the most efficient for creating a Common Azimuth but inefficient for creating multi-azimuth data. The adjoint approach also shows noticeable amplitude dimming and brightening due to acquisition geometry. Formulating the regularization problem as an inverse problem and then approximating the Hessian with a diagonal operator provides better results. The cost and memory requirements are significantly increased but multi-azimuth data comes for free. Estimating the model with a conjugate gradient solver produces the best results. Amplitude artifacts are virtually eliminated and the frequency content is noticeably improved.

ACKNOWLEDGMENTS

I would like than TotalFinaElf for providing the data used in this paper.

REFERENCES

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

Biondi, B., Clapp, R., and Crawley, S., 1996, Seplib90: Seplib for 3-D prestack data: SEP–**92**, 343–364.

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

Chemingui, N., 1999, Imaging irregularly sampled 3D prestacked data: SEP–**101**.

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

Clapp, R. G., 2003b, AMO regularization: Effective approximate inverses for amplitude preservation: SEP–**114**, 159–170.

Clapp, R. G., 2005, Inversion and fault tolerant parallelization using Python: SEP–**120**, 41–62.

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.

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.

Verschuur, D. J., Berkhout, A. J., and Wapenaar, C. P. A., 1992, Adaptive surface-related multiple elimination: Adaptive surface-related multiple elimination:, Soc. of Expl. Geophys., Geophysics, 1166–1177.

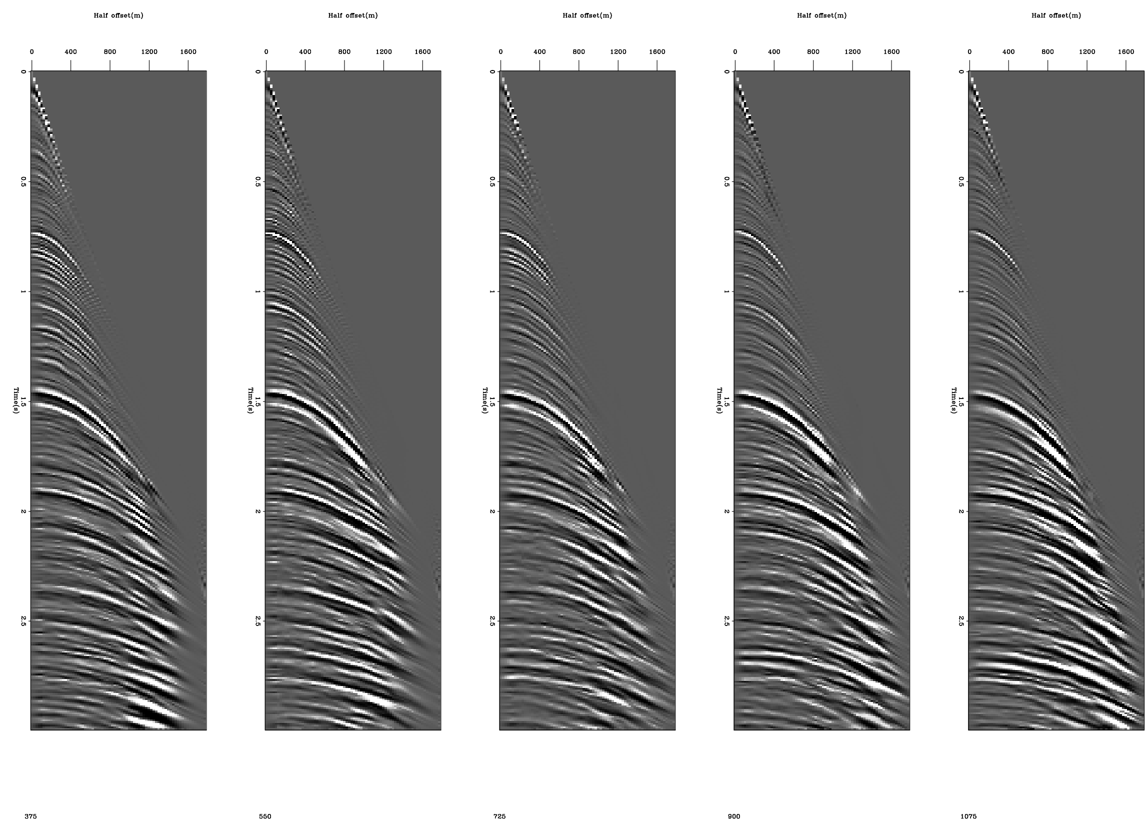

Figure 3: The result of ten iterations minimizing equation (11). Note the improved frequency content compared to Figure 2 and the decreased dimming and brightening of both Figure 1 and Figure 2. bob2-inv0.cmp [CR]