Stanford Exploration Project, Report SERGEY, November 9, 2000, pages 215–??

Missing data interpolation by recursive filter preconditioning

Sergey Fomel, Robert Clapp, and Jon Claerbout[1]

ABSTRACT

Missing data interpolation problems can be conveniently preconditioned by recursive inverse filtering. A helix transform allows us to implement this idea in the multidimensional case. We show with examples that helix preconditioning can give a magnitude-order speedup in comparison with the older methods.

INTRODUCTION

A recent work (Claerbout, 1997) proposed a *helix* transform for mapping multidimensional convolution operators to their one-dimensional equivalents. The helix idea proves the feasibility of multidimensional deconvolution, an issue that has been in question for more than 15 years. By mapping discrete convolution operators to one-dimensional space, the inverse filtering problem can be conveniently recast in terms of recursive filtering, a well-known part of the digital filtering theory.

In this paper, we show how recursive deconvolution can be applied for preconditioning interpolation problems. We consider a problem of filling empty bins in a regularly gridded data volume. For a given estimate of the regularization filter, the missing data problem reduces to least-square optimization. Theoretical analysis and numerical examples show that helix preconditioning can produce a significant speed-up in the convergence of the iterative optimization schemes.

THEORY OF MISSING DATA INTERPOLATION

Claerbout (1992) formulates the basic principle of missing data interpolation as follows:

> A method for restoring missing data is to ensure that the restored data, after specified filtering, has minimum energy.

Mathematically, this principle can be expressed by the simple equation

$$\mathbf{D m} \approx \mathbf{0}\,, \tag{1}$$

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

where $\boldsymbol{m}$ is the data vector, and $\boldsymbol{D}$ is the specified filter. The approximate equality sign means that equation (1) is solved by minimizing the squared norm (the power) of its left side. Additionally, the known data values must be preserved in the optimization scheme. Introducing the mask operator $\boldsymbol{K}$, which can be considered as a diagonal matrix with zeros on the missing data locations and ones elsewhere, we can rewrite equation (1) in the more rigorous form

$$\boldsymbol{D}(\boldsymbol{I}-\boldsymbol{K})\boldsymbol{m} \approx -\boldsymbol{D}\boldsymbol{K}\boldsymbol{m} = -\boldsymbol{D}\boldsymbol{m}_k \;, \tag{2}$$

in which $\boldsymbol{I}$ is the identity operator, and $\boldsymbol{m}_k$ is the known portion of the data. It is important to note that equation (2) corresponds to the limiting case of the regularized linear system

$$\begin{cases} \boldsymbol{K}\boldsymbol{m} = \boldsymbol{m}_k \;, \\ \lambda \boldsymbol{D}\boldsymbol{m} \approx \boldsymbol{0} \end{cases} \tag{3}$$

for the scaling coefficient λ approaching zero. This means that we put far more weight on the first equation in (3) and use the second equation only to constrain the null space of the solution. Applying the general theory of data-space regularization (Fomel, 1997), one can immediately transform system (3) to the equation

$$\boldsymbol{K}\boldsymbol{P}\boldsymbol{x} \approx \boldsymbol{m}_k \;, \tag{4}$$

where $\boldsymbol{P}$ is a preconditioning operator, and $\boldsymbol{x}$ is a new variable, connected with $\boldsymbol{m}$ by the simple relationship

$$\boldsymbol{m} = \boldsymbol{P}\boldsymbol{x} \;.$$

In theory, equations (4) and (2) have exactly the same solutions if the following condition is satisfied:

$$\boldsymbol{P}\boldsymbol{P}^T = (\boldsymbol{D}^T\boldsymbol{D})^{-1} \;, \tag{5}$$

where we need to assume the self-adjoint operator $\boldsymbol{D}^T\boldsymbol{D}$ to be invertible. If $\boldsymbol{D}$ is represented by a discrete convolution, the natural choice for $\boldsymbol{P}$ is the corresponding deconvolution operator:

$$\boldsymbol{P} = \boldsymbol{D}^{-1} \;. \tag{6}$$

The helix transform provides a constructive way of implementing multidimensional deconvolution by one-dimensional recursive filtering.

EXAMPLES

The first two examples in this paper are taken directly from *Geophysical Exploration Mapping* (Claerbout, 1997). They start from a simple 1-D synthetic data test. Figure 1 shows the interpolation results of the unpreconditioned technique with three different filters. For comparison with the preconditioned scheme, we changed the boundary convolution conditions from internal to truncated transient convolution. The system was solved with a conjugate-gradient iterative optimization. As depicted on the right side of the figures, the interpolation process

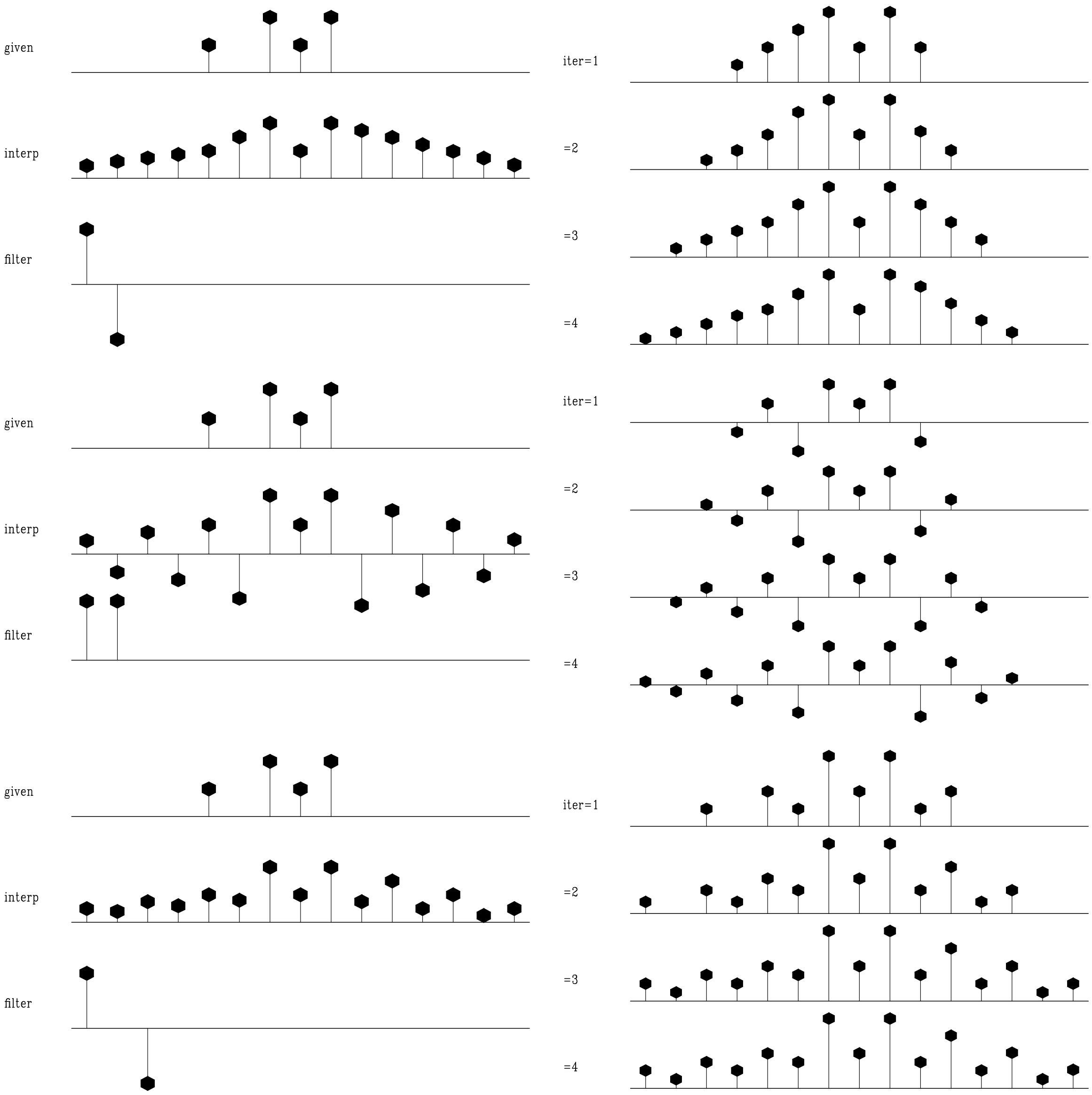

Figure 1: Unpreconditioned interpolation with three different regularization filters. On the left plot: the top shows the input data; the middle, the result of interpolation; and the bottom, the filter. The right plot shows the convergence process for the first four iterations. $\boxed{\text{mishel-mall}}$ [ER]

starts with a "complicated" model and slowly "simplifies" it until the final result is achieved.

Preconditioned interpolation (Figure 2) behaves differently. At the early iterations, the model is simple. As the iteration proceeds, new details are added into the model. After a surprisingly small number of iterations, the output closely resembles the final output. This observation is fully consistent with the general theory of regularization and preconditioning (Nichols, 1994; Harlan, 1995; Fomel, 1997). The final output of interpolation with recursive deconvolution preconditioning is exactly the same as that of the original method.

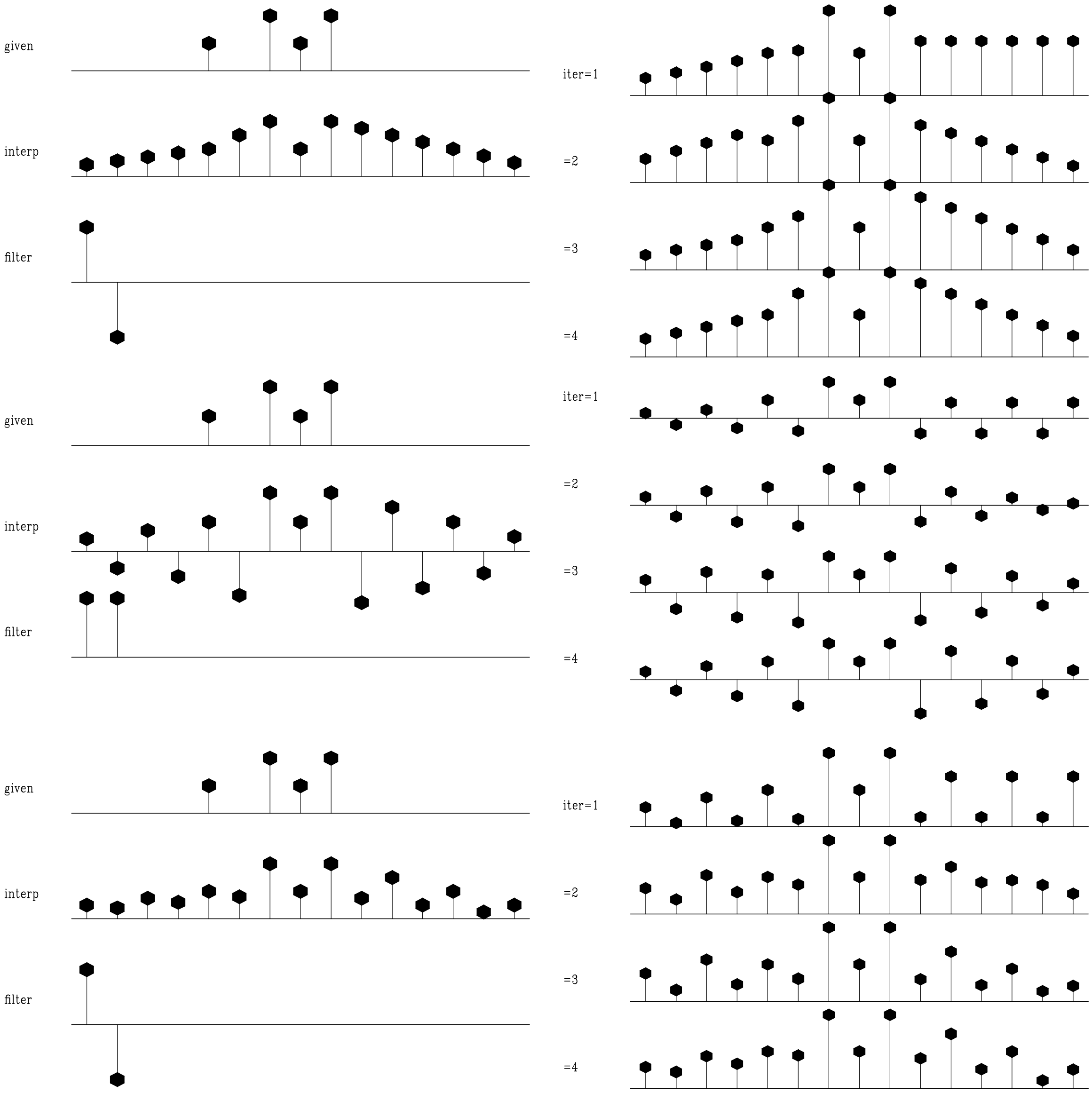

Figure 2: Interpolation with preconditioning. On the left plot: the top shows the input data; the middle, the result of interpolation; and the bottom, the filter. The right plot shows the convergence process for the first four iterations. mishel-sall [ER]

The next example is the SeaBeam dataset, a result of water bottom measurements from a single day of acquisition. This dataset has been used at SEP for benchmarking different strategies of data interpolation. The left plot in Figure 3 shows the original data. The right plot shows the result of (unpreconditioned) missing data interpolation with the Laplacian filter. The result is unsatisfactory, because the Laplacian filter doesn't absorb the spatial frequency distribution of the input dataset. We judge the quality of an interpolation scheme by its ability to hide the footprints of the acquisition geometry in the final result.

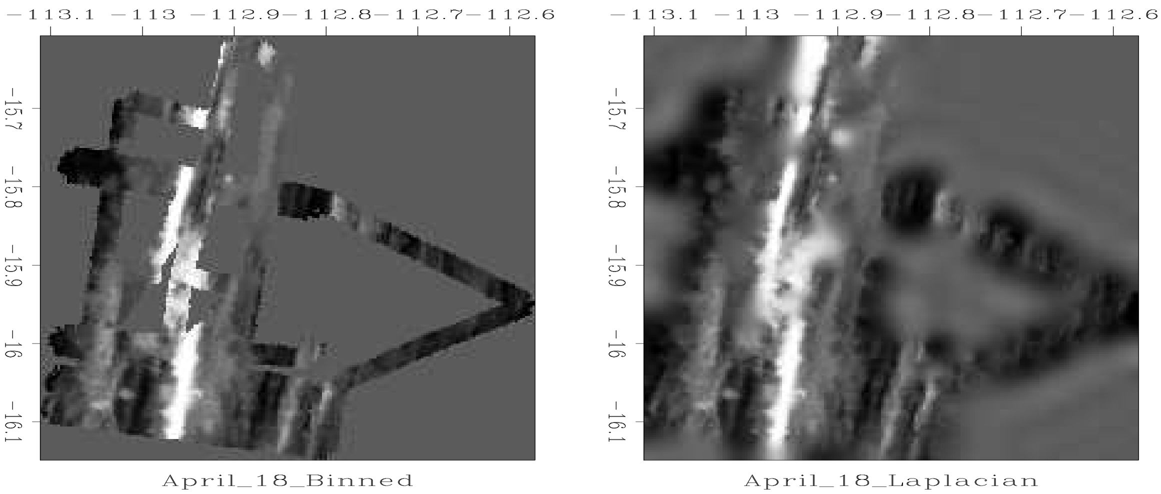

Figure 3: On the left, the SeaBeam data: the depth of the ocean under ship tracks; on the right, an interpolation with the Laplacian filter. mishel-seabdat [ER]

Claerbout (1997) obtains a significantly better result (Figure 4) by replacing the Laplacian filter with a *prediction-error filter* (PEF), estimated from the input data. The result in the left plot of Figure 4 was obtained after 200 conjugate-gradient iterations. If we stop after 20 iterations, the output (the right plot in Figure 4) shows only a small deviation from the input data. Large areas of the image remain unfilled.

Inverting the PEF convolution with the help of the helix transform, we can now apply the inverse filtering operator to precondition the interpolation problem. As expected, the result after 200 iterations (the left plot in Figure 5) is similar to the result of the corresponding unpreconditioned interpolation. However, the output after just 20 iterations (the right plot in Figure 5) is already fairly close to the solution.

For our third example we apply the preconditioning methodology to simulate interpolating well log velocities using reflector dip information as a guide. In the first two cases we used a space-invariant filter for our operator D, and the corresponding inverse P. In this example $\boldsymbol{D}$ is composed of a series of steering filters, small plane wave anihilators, oriented at some a priori angle(Clapp et al., 1997).

We started with a velocity field from a synthetic anticline model above a horizontal unconformity. To build the steering filters we make the assumption that velocity follows reflector dips. We first select four reflectors that characterize dip in the section (Figure 6, top right). The selected dips are then interpolated to all model locations and smoothed (Figure 6, top left). Using this dip field, the methodology in equation (6), and a series of well logs (Figure

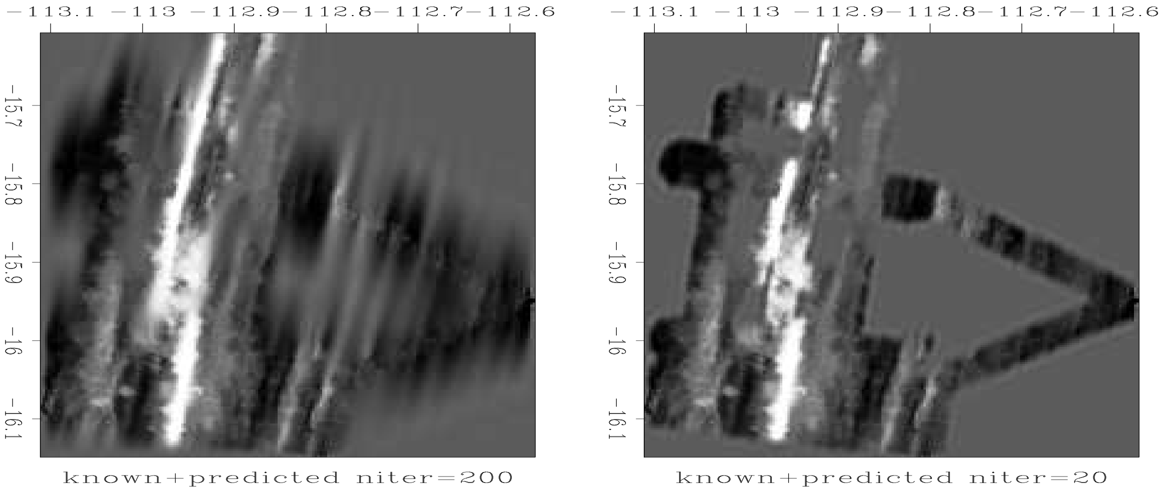

Figure 4: SeaBeam interpolation with the prediction-error filter. The left plot was taken after 200 conjugate-gradient iterations; the right, after 20 iterations. mishel-seabold [ER,M]

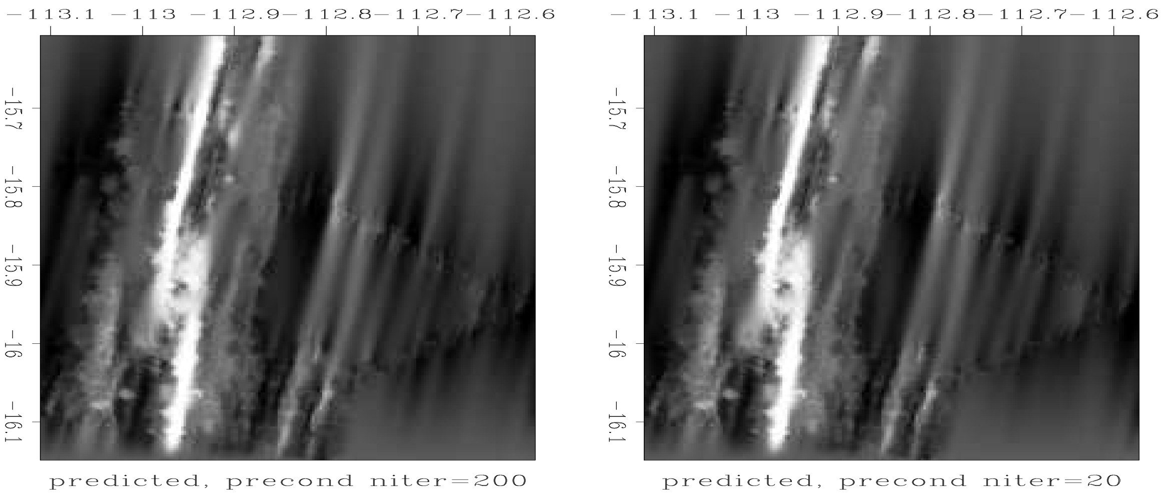

Figure 5: SeaBeam interpolation with the inverse prediction-error filter. The left plot was taken after 200 conjugate-gradient iterations; the right, after 20 iterations. mishel-seabnew [ER,M]

6, bottom left) constructed from the original velocity model, we attempted to reinterpolate the unknown velocities. The bottom right plot of Figure 6 shows that the interpolation was successful in minimal iterations (in this case only 12 iterations were required).

DISCUSSION

The result of this work can be interpreted in a broader context of geophysical estimation (often called inversion). The basic formulation of a geophysical estimation problem consists of setting up two goals, one for data fitting, and the other for model smoothing. These two goals may be written as:

$$0 \approx \mathrm{Lm} - \mathrm{d} \tag{7}$$

$$0 \approx \mathrm{Am} \tag{8}$$

which defines two residuals, a so-called "data residual" and a "model residual" that are usually minimized by conjugate-gradient, least-squares methods.

Perhaps the most straightforward application is geophysical mapping. Then d is data sprinkled randomly around in space, L is the linear interpolation operator, and m is the vector of unknown map values on a cartesian mesh. Many map pixels have no data values; and they are determined by the model-residual goal (damping) which is generally specified by a "roughening operator" A. Our experience shows that binning is often a useful approximation to interpolation L. With binning, our fitting goals look formally the same, but they are a little easier to understand

$$0 \approx \mathrm{Km} - \mathrm{b} \tag{9}$$

$$0 \approx \mathrm{Am} = \mathrm{x} \tag{10}$$

where b denotes binned data values and K is an identity matrix for nonempty bins and zero for empty ones. Also, we introduce the roughened model $\mathrm{x} = \mathrm{Am}$. Claerbout (1992, 1997) shows how to estimate the PEF A and shows that the roughened model x is indeed a "spectrally whitened" model. It is white in the multidimensional space of the model and white in the space of the unwound helix. In other words, the autocorrelation of x is an impulse function in either one-dimensional "unwound" space or in multidimensional physical space.

A good preconditioner is one that somehow allows iterative solvers to obtain their solutions in a fewer numbers of iterations. It is easy to guess a preconditioner and try it to see if it helps. Start from the fitting goals (9) and (10) for finding the model m, and any transformation B. Implicitly define a new variable y by $\mathrm{m} = \mathrm{By}$; insert it into the goals (9) and (10); iteratively solve for y; and finally convert y back to m with $\mathrm{m} = \mathrm{By}$. You have found a good preconditioner if you have solved the problem in fewer iterations.

The helix enters the picture because it offers us another guess for the operator B. As we have shown in this paper, the guess $\mathrm{B} = \mathrm{A}^{-1}$ is an outstanding choice, speeding by an order of magnitude (or more) the solution to the first problem we tried.

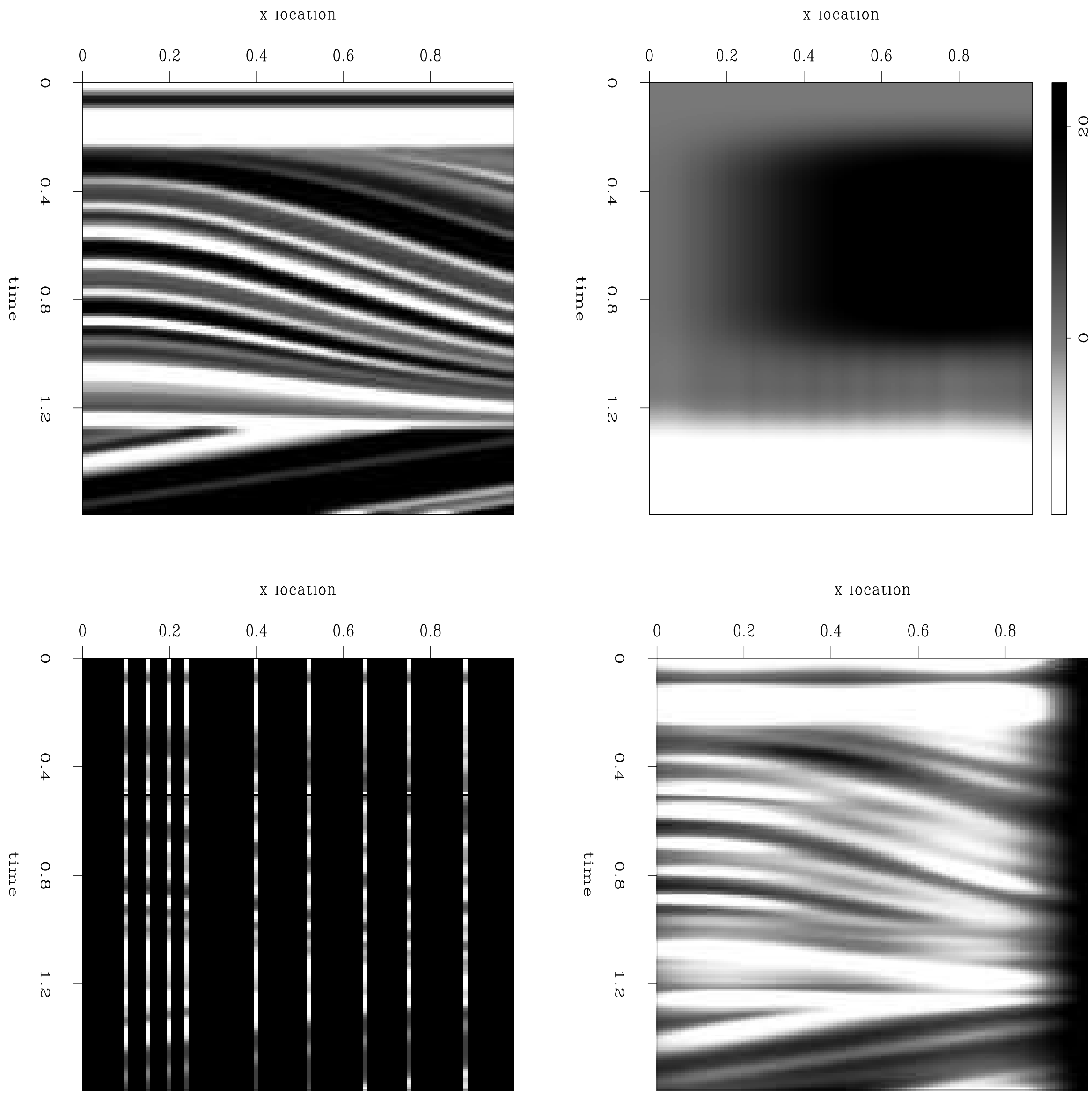

Figure 6: Interpolating a synthetic velocity field from the well data. The left top plot shows the synthetic model; top right, the dip field for calculating steering filters; bottom left, the input data; bottom right, the interpolation result. Only 12 conjugate-gradient interpolations were required. mishel-qdome-combo1 [ER]

The spectacularly successful guess is this: Instead of iteratively fitting the goals (9) and (10) for the model m we recast those goals for the whitened model x. Substituting $\mathrm{m} = \mathrm{A}^{-1}\mathrm{x}$ into the fitting goals we get

$$0 \approx \mathrm{KA}^{-1}\mathrm{x} - \mathrm{b} \tag{11}$$

$$0 \approx \mathrm{x} \tag{12}$$

When a fitting task is large and the iterations cannot go to completion then it is often suggested that we simply omit the damping (12) and regard the results of each iteration as the result of decreasing the amount of model damping. We find this idea to have merit when the model goal is cast as $0 \approx \mathrm{x}$ but not when the model smoothing goal is cast in the equivalent form $0 \approx \mathrm{Am}$.

To move towards the fitting goal (11), we start at $\mathrm{x} = \mathrm{x}_0$ where often $\mathrm{x}_0 = 0$. For each iteration, we apply polynomial division by the PEF on the helix, A^{-1}. It is very quick. At the end, we can plot the deconvolved map x, the map itself $\mathrm{m} = \mathrm{Ax}$, or the known bin values with the empties replaced by their prediction, $\mathrm{Km} + (\mathrm{I} - \mathrm{K})\mathrm{Ax}$. Our first results are exciting because they solve the problem so rapidly that we anticipate success with problems of industrial scale.

This example suggests that the philosophy of image creation by optimization has a dual orthonormality: First, Gauss (and common sense) tells us that the data residuals should be roughly equal in size. Likewise in Fourier space they should be roughly equal in size, which means they should be roughly white, i.e. orthonormal. (I use the word "orthonormal" because white means the autocorrelation is an impulse, which means the signal is statistically orthogonal to shifted versions of itself.) Second, to speed convergence of iterative methods, we need a whiteness, another othonormality, in the solution. The map image, the physical function that we seek, might not be itself white, so we should solve first for another variable, the whitened map image, and as a final step, transform it to the "natural colored" map.

Often geophysicists create a preconditioning matrix B by inventing columns that "look like" the solutions that they seek. Then the space x has many fewer components than the space of m. This approach is touted as a way of introducing geological and geophysical prior information into the solution. Indeed, it strongly imposes the form of the solution. Perhaps this approach deserves the diminutive term "curve fitting" instead of the grandiloquent "geophysical inverse theory." Our preferred approach is not to invent the columns of the preconditioning matrix, but to estimate the prediction-error filter of the model and use its inverse.

CONCLUSIONS

Applying inverse filtering operators that we can construct with the helix transform to precondition interpolation problems, we observe a significant (order of magnitude) speed-up in the optimization convergence. Since inverse recursive filtering takes almost the same time as forward convolution, the acceleration translates straightforwardly into computational time savings.

For simple test problems, these savings are hardly noticeable. On the other hand, for large-scale (seismic-exploration-size) problems, the achieved acceleration can have a direct impact on the mere feasibility of iterative least-square inversion.

ACKNOWLEDGMENTS

We would like to thank Dave Nichols for helpful discussions.

REFERENCES

Claerbout, J. F., 1992, Earth Soundings Analysis: Processing Versus Inversion: Blackwell Scientific Publications.

Claerbout, J. F., 1997, Geophysical exploration mapping: Environmental soundings image enhancement: Stanford Exploration Project.

Clapp, R. G., Fomel, S., and Claerbout, J., 1997, Solution steering with space-variant filters: SEP–**95**, 27–42.

Fomel, S., 1997, On model-space and data-space regularization: A tutorial: SEP–**94**, 141–164.

Harlan, W. S., 1995, Regularization by model redefinition: `http://sepwww.stanford.edu/oldsep/harlan/papers/regularization.ps.gz`.

Nichols, D., 1994, A simple example of a null space and how to modify it: SEP–**82**, 177–182.