Multidimensional recursive filter preconditioning in geophysical estimation problems[a]

[a] Published in Geophysics, 68, 577-588 (2003)

Sergey Fomel[1] *and Jon F. Claerbout*[2]

ABSTRACT

Constraining ill-posed inverse problems often requires regularized optimization. We consider two alternative approaches to regularization. The first approach involves a column operator and an extension of the data space. It requires a regularization operator which enhances the undesirable features of the model. The second approach constructs a row operator and expands the model space. It employs a preconditioning operator, which enforces a desirable behavior, such as smoothness, of the model. In large-scale problems, when iterative optimization is incomplete, the second method is preferable, because it often leads to faster convergence. We propose a method for constructing preconditioning operators by multidimensional recursive filtering. The recursive filters are constructed by imposing helical boundary conditions. Several examples with synthetic and real data demonstrate an order of magnitude efficiency gain achieved by applying the proposed technique to data interpolation problems.

INTRODUCTION

Regularization is a method of imposing additional conditions for solving inverse problems with optimization methods. When model parameters are not fully constrained by the problem (i.e. the inverse problem is mathematically ill-posed), regularization restricts the variability of the model and guides iterative optimization to the desired solution by using additional assumptions about the model power, smoothness, predictability, etc. In other words, it constrains the model null space to an *a priori* chosen pattern. A thorough mathematical theory of regularization has been introduced by works of Tikhonov's school (Tikhonov and Arsenin, 1977).

In this paper, we discuss two alternative formulations of regularized least-squares inversion problems. The first formulation, which we call *model-space*, extends the data space and constructs a composite column operator. The second, *data-space*,

[1] *Bureau of Economic Geology, Jackson School of Geosciences, The University of Texas at Austin, University Station, Box X, Austin, TX 78713-8972, USA*

[2] *Stanford Exploration Project, Mitchell Bldg., Department of Geophysics, Stanford University, Stanford, California 94305-2215, USA*

formulation extends the model space and constructs a composite row operator. This second formulation is intrinsically related to the concept of model preconditioning (Vandecar and Snieder, 1994). We illustrate the general regularization theory with simple synthetic examples.

Though the final results of the model-space and data-space regularization are theoretically identical, the behavior of iterative gradient-based methods, such as the method of conjugate gradients, is different for the two cases. The obvious difference is in the case where the number of model parameters is significantly larger than the number of data measurements. In this case, the dimensions of the inverted matrix in the case of the data-space regularization are smaller than those of the model-space matrix, and the convergence of the iterative conjugate-gradient iteration requires correspondingly smaller number of iterations. But even in the case where the number of model and data parameters are comparable, preconditioning changes the iteration behavior. This follows from the fact that the objective function gradients with respect to the model parameters are different in the two cases. Since iteration to the exact solution is rarely achieved in large-scale geophysical applications, the results of iterative optimization may turn out quite differently. Harlan (1995) points out that the goals of the model-space regularization conflict with each other: the first one emphasizes "details" in the model, while the second one tries to smooth them out. He describes the advantage of preconditioning as follows:

> The two objective functions produce different results when optimization is incomplete. A descent optimization of the original (*model-space*) objective function will begin with complex perturbations of the model and slowly converge toward an increasingly simple model at the global minimum. A descent optimization of the revised (*data-space*) objective function will begin with simple perturbations of the model and slowly converge toward an increasingly complex model at the global minimum. . . . A more economical implementation can use fewer iterations. Insufficient iterations result in an insufficiently complex model, not in an insufficiently simplified model.

In this paper, we illustrate the two approaches on synthetic and real data examples from simple environmental data sets. All examples show that when we solve the optimization problem iteratively and take the output only after a limited number of iterations, it is preferable to use the preconditioning approach. When the regularization operator is convolution with a filter, a natural choice for preconditioning is inverse filtering (recursive deconvolution). We show how to extend the method of preconditioning by recursive filtering to multiple dimensions. The extention is based on modifying the boundary conditions with the helix transform (Claerbout, 1998).

REVIEW OF REGULARIZATION IN ESTIMATION PROBLEMS

The basic linear system of equations for least-squares optimization can be written in the form

$$\mathbf{d} - \mathbf{L}\mathbf{m} \approx \mathbf{0}\ , \tag{1}$$

where $\mathbf{m}$ is the vector of model parameters, $\mathbf{d}$ is the vector of experimental data values, $\mathbf{L}$ is the modeling operator, and the approximate equality implies finding the solution by minimizing the power of the left-hand side. The goal of linear estimation is to estimate an optimal $\mathbf{m}$ for a given $\mathbf{d}$.

Model-space regularization

Model-space regularization implies adding equations to system (1) to obtain a fully constrained (well-posed) inverse problem. The additional equations take the form

$$\epsilon \mathbf{D}\mathbf{m} \approx \mathbf{0}\ , \tag{2}$$

where $\mathbf{D}$ is a linear operator that represents additional requirements for the model, and ϵ is the scaling parameter. In many applications, $\mathbf{D}$ can be thought of as a filter, enhancing undesirable components in the model, or as the operator of a differential equation that we assume the model should satisfy.

The full system of equations (1-2) can be written in a short notation as

$$\mathbf{G_m}\mathbf{m} = \left[\begin{array}{c} \mathbf{L} \\ \epsilon \mathbf{D} \end{array}\right] \mathbf{m} \approx \left[\begin{array}{c} \mathbf{d} \\ \mathbf{0} \end{array}\right] = \hat{\mathbf{d}}\ , \tag{3}$$

where $\hat{\mathbf{d}}$ is the augmented data vector:

$$\hat{\mathbf{d}} = \left[\begin{array}{c} \mathbf{d} \\ \mathbf{0} \end{array}\right]\ , \tag{4}$$

and $\mathbf{G_m}$ is a *column* operator:

$$\mathbf{G_m} = \left[\begin{array}{c} \mathbf{L} \\ \epsilon \mathbf{D} \end{array}\right]\ . \tag{5}$$

The estimation problem (3) is fully constrained. We can solve it by means of unconstrained least-squares optimization, minimizing the least-squares norm of the compound residual vector

$$\hat{\mathbf{r}} = \hat{\mathbf{d}} - \mathbf{G_m}\mathbf{m} = \left[\begin{array}{c} \mathbf{d} - \mathbf{L}\mathbf{m} \\ -\epsilon \mathbf{D}\mathbf{m} \end{array}\right]\ . \tag{6}$$

The formal solution of the regularized optimization problem has the known form (Parker, 1994)

$$<\mathbf{m}>= \left(\mathbf{L}^T\,\mathbf{L} + \epsilon^2\,\mathbf{D}^T\,\mathbf{D}\right)^{-1}\,\mathbf{L}^T\,\mathbf{d}\;, \tag{7}$$

where $<\mathbf{m}>$ denotes the least-squares estimate of $\mathbf{m}$, and $\mathbf{L}^T$ denotes the adjoint operator. One can carry out the optimization iteratively with the help of the conjugate-gradient method (Hestenes and Steifel, 1952) or its analogs (Paige and Saunders, 1982).

In the next subsection, we describe an alternative formulation of the optimization problem.

Data-space regularization (model preconditioning)

The data-space regularization approach is closely connected with the concept of *model preconditioning*. We can introduce a new model $\mathbf{p}$ with the equality

$$\mathbf{m} = \mathbf{P}\mathbf{p}\;, \tag{8}$$

where $\mathbf{P}$ is a preconditioning operator. The residual vector $\mathbf{r}$ for the data-fitting goal (1) can be defined by the relationship

$$\epsilon\mathbf{r} = \mathbf{d} - \mathbf{L}\mathbf{m} = \mathbf{d} - \mathbf{L}\mathbf{P}\mathbf{p}\;, \tag{9}$$

where ϵ is the same scaling parameter as in (2) – the reason for this choice will be clear from the analysis that follows. Let us consider a compound model $\hat{\mathbf{p}}$, composed of the preconditioned model vector $\mathbf{p}$ and the residual $\mathbf{r}$:

$$\hat{\mathbf{p}} = \left[\begin{array}{c}\mathbf{p}\\ \mathbf{r}\end{array}\right]\;. \tag{10}$$

With respect to the compound model, we can rewrite equation (9) as

$$\left[\begin{array}{cc}\mathbf{L}\mathbf{P} & \epsilon\mathbf{I}\end{array}\right]\left[\begin{array}{c}\mathbf{p}\\ \mathbf{r}\end{array}\right] = \mathbf{G_d}\hat{\mathbf{p}} = \mathbf{d}\;, \tag{11}$$

where $\mathbf{G_d}$ is a *row* operator:

$$\mathbf{G_d} = \left[\begin{array}{cc}\mathbf{L}\mathbf{P} & \epsilon\mathbf{I}\end{array}\right]\;, \tag{12}$$

and $\mathbf{I}$ represents the data-space identity operator.

Equation (11) is clearly underdetermined with respect to the compound model $\hat{\mathbf{p}}$. If from all possible solutions of this system we seek the one with the minimal power $\hat{\mathbf{p}}^T\hat{\mathbf{p}}$, the formal result takes the well-known form

$$<\hat{\mathbf{p}}>= \left[\begin{array}{c}<\mathbf{p}>\\ <\mathbf{r}>\end{array}\right] = \mathbf{G_d}^T\left(\mathbf{G_d}\,\mathbf{G_d}^T\right)^{-1}\mathbf{d} = \left[\begin{array}{c}\mathbf{P}^T\mathbf{L}^T\left(\mathbf{L}\mathbf{P}\mathbf{P}^T\mathbf{L}^T + \epsilon^2\mathbf{I}\right)^{-1}\mathbf{d}\\ \epsilon\left(\mathbf{L}\mathbf{P}\mathbf{P}^T\mathbf{L}^T + \epsilon^2\mathbf{I}\right)^{-1}\mathbf{d}\end{array}\right]\;. \tag{13}$$

Applying equation (8), we obtain the corresponding estimate $<\mathbf{m}>$ for the initial model $\mathbf{m}$, as follows:

$$<\mathbf{m}>= \mathbf{P}\,\mathbf{P}^T\,\mathbf{L}^T\,\left(\mathbf{L}\,\mathbf{P}\,\mathbf{P}^T\,\mathbf{L}^T + \epsilon^2\,\mathbf{I}\right)^{-1}\,\mathbf{d}\;. \tag{14}$$

It is easy to show algebraically that estimate (14) is equivalent to estimate (7) under the condition

$$\mathbf{C} = \mathbf{P}\,\mathbf{P}^T = \left(\mathbf{D}^T\,\mathbf{D}\right)^{-1}\;, \tag{15}$$

where we need to assume the self-adjoint operator $\mathbf{D}^T\,\mathbf{D}$ to be invertible.

To prove the equivalence, consider the operator

$$\mathbf{G} = \mathbf{L}^T\,\mathbf{L}\,\mathbf{C}\,\mathbf{L}^T + \epsilon^2\,\mathbf{L}^T\;, \tag{16}$$

which is a mapping from the data space to the model space. Grouping the multiplicative factors in two different ways, we can obtain the equality

$$\mathbf{G} = \mathbf{L}^T\,\left(\mathbf{L}\,\mathbf{C}\,\mathbf{L}^T + \epsilon^2\,\mathbf{I}\right) = \left(\mathbf{L}^T\,\mathbf{L} + \epsilon^2\mathbf{C}^{-1}\right)\,\mathbf{C}\,\mathbf{L}^T\;, \tag{17}$$

or, in another form,

$$\mathbf{C}\,\mathbf{L}^T\,\left(\mathbf{L}\,\mathbf{C}\,\mathbf{L}^T + \epsilon^2\,\mathbf{I}\right)^{-1} \equiv \left(\mathbf{L}^T\,\mathbf{L} + \epsilon^2\,\mathbf{C}^{-1}\right)^{-1}\,\mathbf{L}^T\;. \tag{18}$$

The left-hand side of equality (18) is exactly the projection operator from formula (14), and the right-hand side is the operator from formula (7).

This proves the legitimacy of the alternative data-space approach to data regularization: the model estimation is reduced to a least-squares minimization of the specially constructed compound model $\hat{\mathbf{p}}$ under the constraint (9).

We summarize the differences between the model-space and data-space regularization in Table 1.

Although the two approaches lead to similar theoretical results, they behave quite differently in the process of iterative optimization. In the next section, we illustrate this fact with many examples and show that in the case of incomplete optimization, the second (preconditioning) approach is generally preferable.

ONE-DIMENSIONAL SYNTHETIC EXAMPLES

In the first example, the input data $\mathbf{d}$ were randomly subsampled (with decreasing density) from a sinusoid (Figure 1). The forward operator $\mathbf{L}$ in this case is linear interpolation. In other words, we seek a regularly sampled model $\mathbf{m}$ on 200 grid points that could predict the data with a forward linear interpolation. Sparse irregular distribution of the input data makes the regularization necessary. We applied convolution with the simple $(1, -1)$ difference filter as the operator $\mathbf{D}$ that forces

	Model-space	*Data-space*
effective model	$\mathbf{m}$	$\hat{\mathbf{p}} = \left[\begin{array}{c} \mathbf{p} \\ \mathbf{r} \end{array}\right]$
effective data	$\hat{\mathbf{d}} = \left[\begin{array}{c} \mathbf{d} \\ \mathbf{0} \end{array}\right]$	$\mathbf{d}$
effective operator	$\mathbf{G_m} = \left[\begin{array}{c} \mathbf{L} \\ \epsilon\mathbf{D} \end{array}\right]$	$\mathbf{G_d} = \left[\begin{array}{cc} \mathbf{LP} & \epsilon\mathbf{I} \end{array}\right]$
optimization problem	minimize $\hat{\mathbf{r}}^T\hat{\mathbf{r}}$, where $\hat{\mathbf{r}} = \hat{\mathbf{d}} - \mathbf{G_m}\mathbf{m}$	minimize $\hat{\mathbf{p}}^T\hat{\mathbf{p}}$ under the constraint $\mathbf{G_d}\hat{\mathbf{p}} = \mathbf{d}$
formal estimate for $\mathbf{m}$	$\left(\mathbf{L}^T\mathbf{L} + \epsilon^2\mathbf{C}^{-1}\right)\mathbf{L}^T\mathbf{d}$, where $\mathbf{C}^{-1} = \mathbf{D}^T\mathbf{D}$	$\mathbf{C}\mathbf{L}^T(\mathbf{L}\mathbf{C}\mathbf{L}^T + \epsilon^2\mathbf{I})^{-1}\mathbf{d}$, where $\mathbf{C} = \mathbf{P}\mathbf{P}^T$.

Table 1: Comparison between model-space and data-space regularization

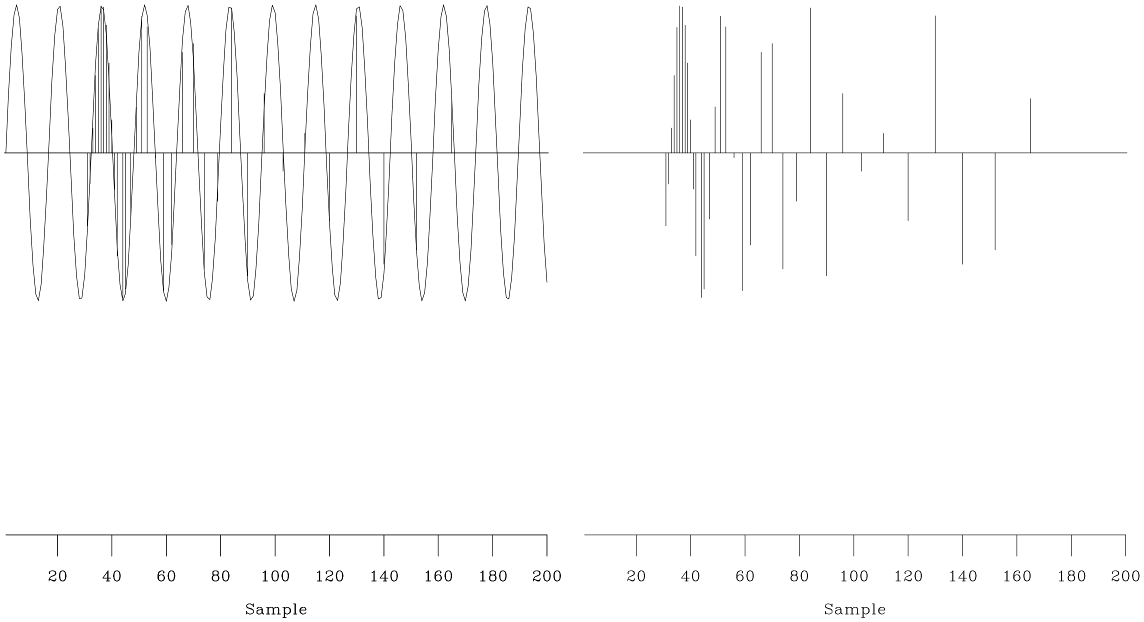


Figure 1: The input data (right) are irregularly spaced samples of a sinusoid (left).

model continuity (the first-order spline). An appropriate preconditioner **P** in this case is the inverse of **D**, which amounts to recursive causal integration (Claerbout, 1999). Figures 2 and 3 show the results of inverse interpolation after exhaustive 300 iterations of the conjugate-direction method. The results from the model-space and data-space regularization look similar except for the boundary conditions outside the data range. As a result of using the causal integration for preconditioning, the rightmost part of the model in the data-space case stays at a constant level instead of decreasing to zero. If we specifically wanted a zero-value boundary condition, we could easily implement it by adding a zero-value data point at the boundary.

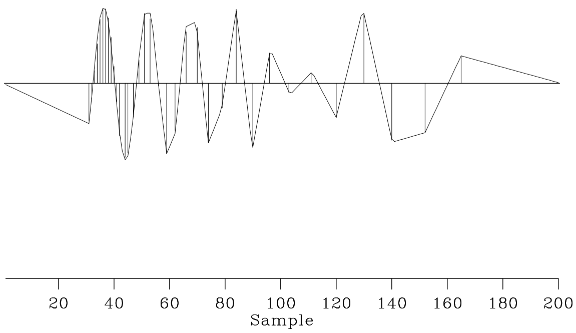


Figure 2: Estimation of a continuous function by the model-space regularization. The difference operator **D** is the derivative operator (convolution with $(1,-1)$).

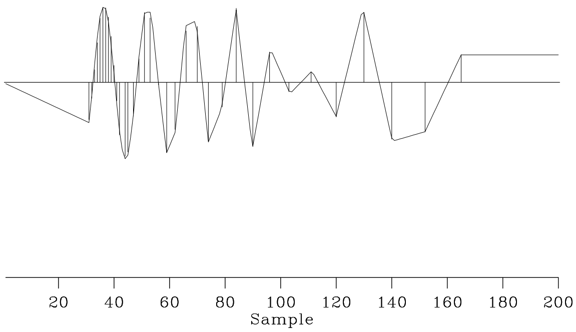


Figure 3: Estimation of a continuous function by the data-space regularization. The preconditioning operator **P** is causal integration.

The model preconditioning approach provides a much faster rate of convergence. We measured the rate of convergence using the power of the model residual, defined as the least-squares distance from the current model to the final solution. Figure 4 shows that the preconditioning (data regularization) method converged to the final solution in about 6 times fewer iterations than the model regularization. Since the cost of each iteration for both methods is roughly equal, the computational economy is evident. Figure 5 shows the final solution, and the estimates from model- and data-space regularization after only 5 iterations of conjugate directions. The data-space estimate appears much closer to the final solution.

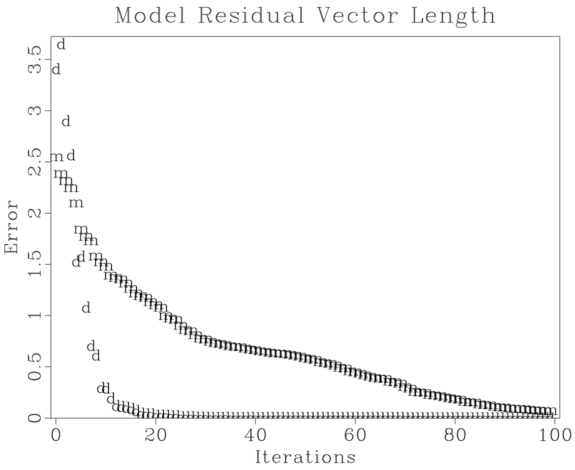


Figure 4: Convergence of the iterative optimization, measured in terms of the model residual. The "d" points stand for data-space regularization; the "m" points for model-space regularization.

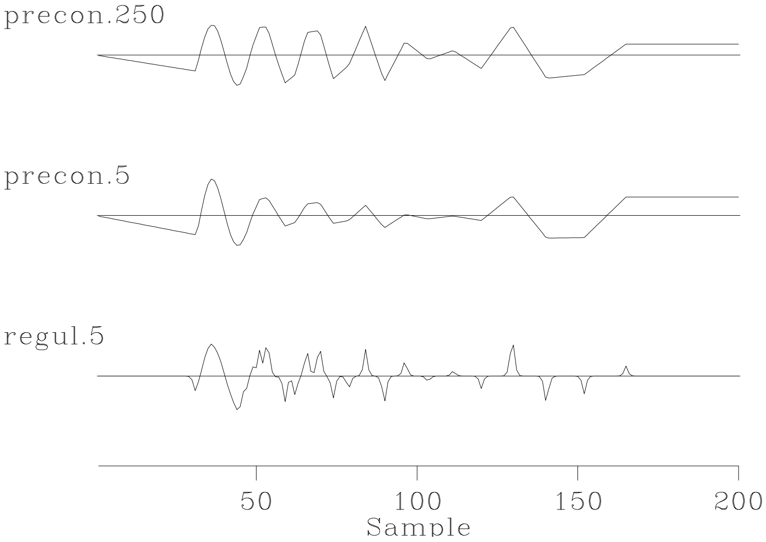
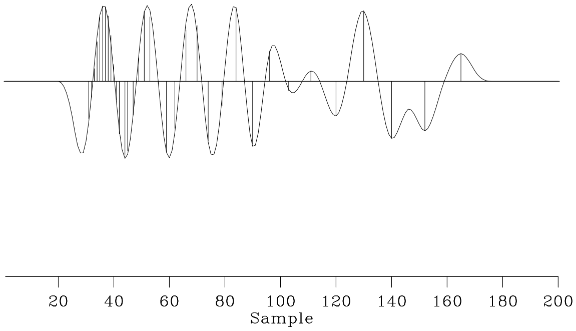


Figure 5: The top figure is the exact solution found in 250 iterations. The middle is with data-space regularization after 5 iterations. The bottom is with model-space regularization after 5 iterations.

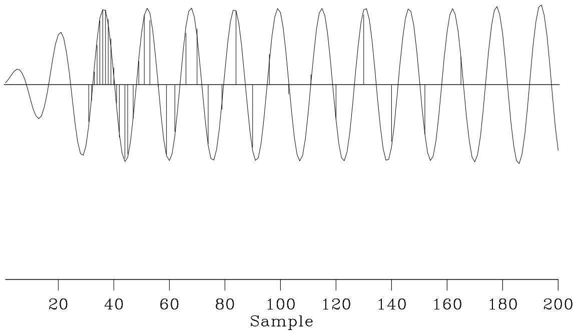
Changing the preconditioning operator changes the regularization result. Figure 6 shows the result of data-space regularization after a triangle smoother is applied as the model preconditioner. Triangle smoother is a filter with the Z-transform $\frac{\left(1-Z^N\right)\left(1-Z^{-N}\right)}{(1-Z)(1-Z^{-1})}$ (Claerbout, 1992). We chose the filter length $N = 6$.

Figure 6: Estimation of a smooth function by the data-space regularization. The preconditioning operator **P** is a triangle smoother.



If, instead of looking for a smooth interpolation, we want to limit the number of frequency components, then the optimal choice for the model-space regularization operator **D** is a prediction-error filter (PEF). To obtain a mono-frequency output, we can use a three-point PEF, which has the Z-transform representation $D(Z) = 1 + a_1 Z + a_2 Z^2$. In this case, the corresponding preconditioner **P** can be the three-point *recursive* filter $P(Z) = 1/(1 + a_1 Z + a_2 Z^2)$. To test this idea, we estimated the PEF $D(Z)$ from the output of inverse linear interpolation (Figure 3), and ran the data-space regularized estimation again, substituting the recursive filter $P(Z) = 1/D(Z)$ in place of the causal integration. We repeated this two-step procedure three times to get a better estimate for the PEF. The result, shown in Figure 7, exhibits the desired mono-frequency output. One can accommodate more frequency components in the model using a longer filter.

Figure 7: Estimation of a mono-frequency function by the data-space regularization. The preconditioning operator **P** is a recursive filter (the inverse of PEF).



Regularization after binning: missing data interpolation

One of the factors affecting the convergence of iterative data regularization is clustering of data points in the output bins. When least-squares optimization assigns equal weight to each data point, it may apply inadequate effort to fit a cluster of data points with similar values in a particular output bin. To avoid this problem, we can replace the regularized optimization with a less accurate but more efficient two-step approach: data binning followed by missing data interpolation.

Missing data interpolation is a particular case of data regularization, where the input data are already given on a regular grid, and we need to reconstruct only the missing values in empty bins. The basic principle of missing data interpolation is formulated as follows (Claerbout, 1992):

> 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

$$\mathbf{D}\mathbf{m} \approx \mathbf{0}\ , \tag{19}$$

where $\mathbf{m}$ is the model vector and $\mathbf{D}$ is the specified filter. Equation (19) is completely equivalent to equation (2). The approximate equality sign means that equation (19) 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 $\mathbf{K}$, which is a diagonal matrix with zeros at the missing data locations and ones elsewhere, we can rewrite equation (19) in the extended form

$$\mathbf{D}(\mathbf{I}-\mathbf{K})\mathbf{m} \approx -\mathbf{D}\mathbf{K}\mathbf{m} = -\mathbf{D}\mathbf{m}_k\ , \tag{20}$$

where $\mathbf{I}$ is the identity operator and $\mathbf{m}_k$ represents the known portion of the data. It is important to note that equation (20) corresponds to the regularized linear system

$$\left\{ \begin{array}{l} \mathbf{K}\mathbf{m} = \mathbf{m}_k\ , \\ \epsilon \mathbf{D}\mathbf{m} \approx \mathbf{0} \end{array} \right. \tag{21}$$

in the limit of the scaling coefficient ϵ approaching zero. System (21) is equivalent to system (1-2) with the masking operator $\mathbf{K}$ playing the role of the forward interpolation operator $\mathbf{L}$. Setting ϵ to zero implies associating more weight on the first equation in (21) and using the second equation only to constrain the null space of the solution. Applying the general theory of data-space regularization from the previous section, we can immediately transform system (21) to the equation

$$\mathbf{K}\mathbf{P}\mathbf{p} \approx \mathbf{m}_k\ , \tag{22}$$

where $\mathbf{P}$ is a preconditioning operator and $\mathbf{p}$ is the preconditioning variable, connected with $\mathbf{m}$ by the simple relationship (8). According to equations (14) and (7) from the previous section, equations (22) and (20) have exactly the same solutions if condition (15) is satisfied. If $\mathbf{D}$ is represented by a discrete convolution, the natural choice for $\mathbf{P}$ is the corresponding deconvolution (inverse recursive filtering) operator:

$$\mathbf{P} = \mathbf{D}^{-1}\ . \tag{23}$$

We illustrate the missing data problem with a simple 1-D synthetic data test taken from Claerbout (1999). Figure 8 shows the interpolation results of the unpreconditioned technique with two different filters **D**. For comparison with the preconditioned scheme, we changed the boundary convolution conditions from internal to truncated transient convolution. As in the previous example, the system was solved with a conjugate-gradient iterative optimization.

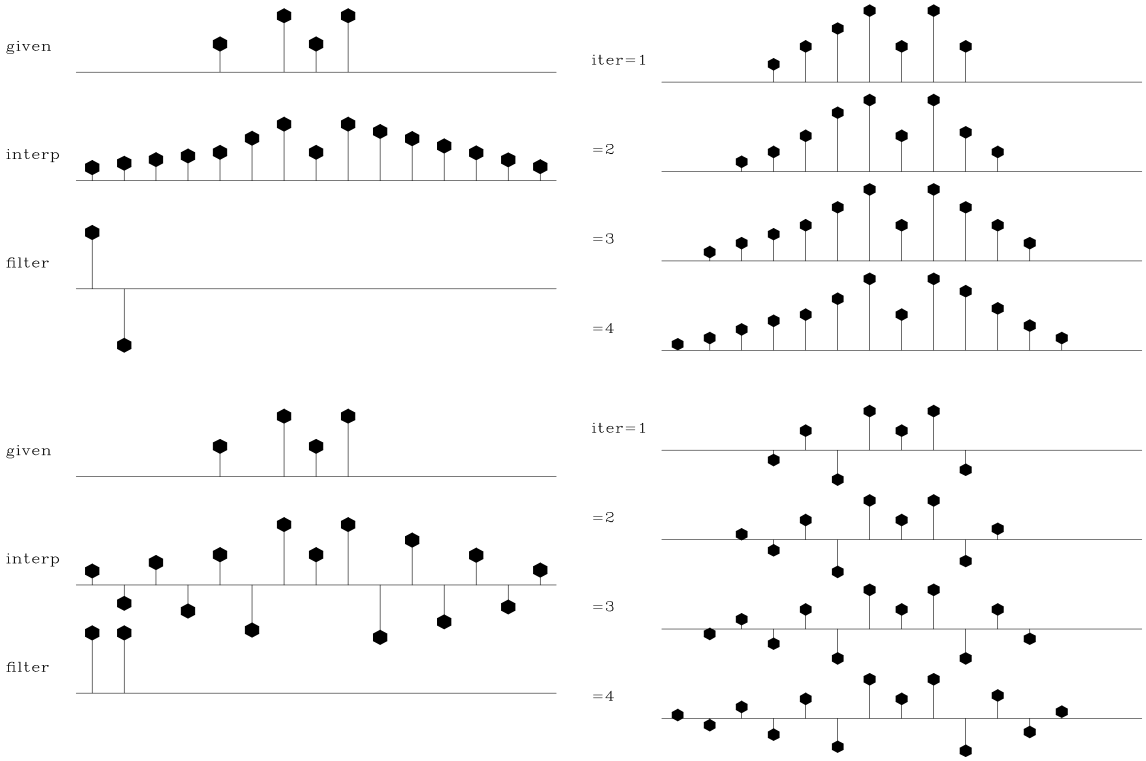


Figure 8: Unpreconditioned interpolation with two different regularization filters. Left plot: the top shows the input data; the middle, the result of interpolation; the bottom, the filter impulse response. The right plot shows the convergence process for the first four iterations.

As depicted on the right side of the figures, the interpolation process starts with a “complicated” model and slowly extends it until the final result is achieved.

Preconditioned interpolation behaves differently (Figure 9). 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. The final output of interpolation with preconditioning by recursive deconvolution is exactly the same as that of the original method.

The next section extends the idea of preconditioning by inverse recursive filtering to multiple dimensions.

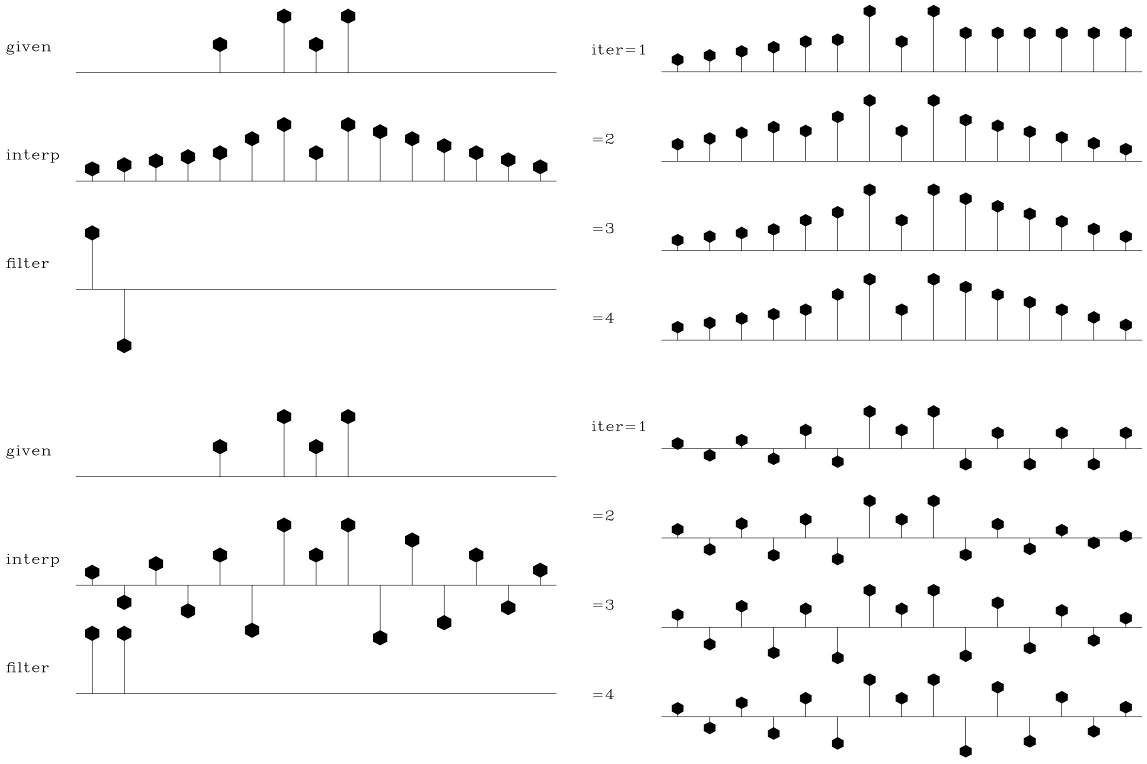


Figure 9: Interpolation with preconditioning. Left plot: the top shows the input data; the middle, the result of interpolation; the bottom, the filter impulse response. The right plot shows the convergence process for the first four iterations.

MULTIDIMENSIONAL RECURSIVE FILTER PRECONDITIONING

Claerbout (1998) proposed a *helix* transform for mapping multidimensional convolution operators to their one-dimensional equivalents. This transform proves the feasibility of multidimensional deconvolution, an issue that has been in question for more than 25 years (Claerbout, 1976). 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.

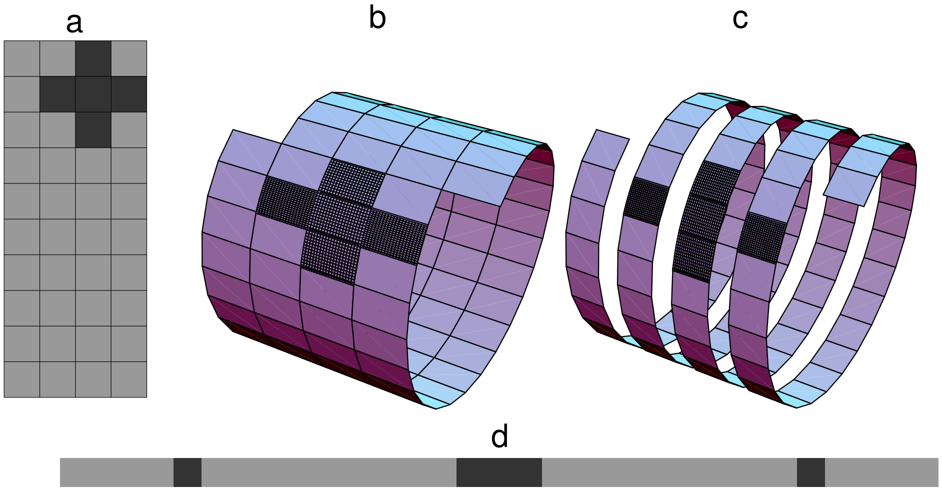


Figure 10: Helix transform of two-dimensional filters to one dimension (a scheme). The two-dimensional filter (plot `a`) is equivalent to the one-dimensional filter in (plot `d`), assuming that a shifted periodic condition is imposed on one of the axes (plots `b` and `c`.)

The helix filtering idea is schematically illustrated in Figure **??**. The left plot (labeled "`a`" in the figure) shows a two-dimensional digital filter overlayed on the computational grid. A two-dimensional convolution computes its output by sliding the filter over the plane. If we impose helical boundary conditions on one of the axes, the filter will slide to the beginning of the next trace after reaching the end of the previous one (plot "`b`"). As evident from plots "`c`" and "`d`", this is completely equivalent to one-dimensional convolution with a long 1-D filter with internal gaps. For efficiency, the gaps are simply skipped in a helical convolution algorithm. The gain is not in the convolution itself, but in the ability to perform recursive inverse filtering (deconvolution) in multiple dimensions. A multi-dimensional filter is mapped to its 1-D analog by imposing helical boundary conditions on the appropriate axes. After that, inverse filtering is applied recursively in a one-dimensional manner. Neglecting parallelization and indexing issues, the cost of inverse filtering is equivalent to the cost of convolution. It is proportional to the data size and to the number of non-zero filter coefficients.

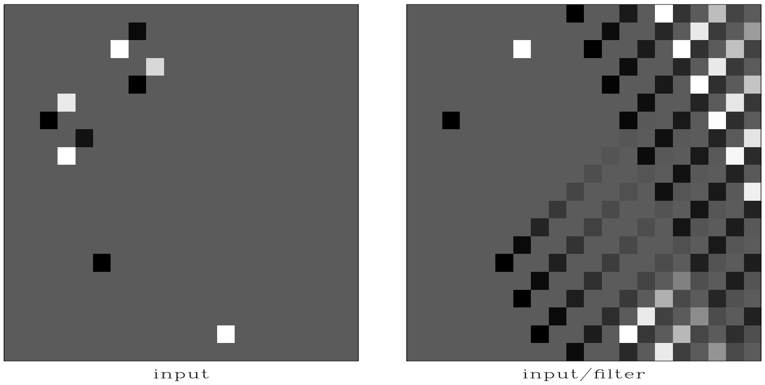


Figure 11: Illustration of 2-D deconvolution with helix transform. Left is the input: two spikes and two filters. Right is the output of deconvolution.

An example of two-dimensional recursive filtering is shown in Figure 11. The left plot contains two spikes and two filter impulse responses with different polarity. After deconvolution with the given filter, the filter responses turn into spikes, and the initial spikes turn into long-tailed inverse impulse responses (right plot in Figure 11). Helical wrap-around, visible on the top and bottom boundaries, indicates the direction of the helix. Claerbout (1999) presents more examples and discusses all the issues of multidimensional helical deconvolution in detail.

MULTIDIMENSIONAL EXAMPLES

Our first multidimensional example is the SeaBeam dataset, a result of water bottom measurements from a single day of acquisition. SeaBeam is an apparatus for measuring water depth both directly under a ship and somewhat off to the sides of the ship's track. The dataset has been used at the Stanford Exploration Project for benchmarking different strategies of data interpolation. The left plot in Figure 12 shows the original data. The right plot shows the result of (unpreconditioned) missing data interpolation with the Laplacian filter after 200 iterations. The result is unsatisfactory, because the Laplacian filter does not 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. The ship track from the original acquisition pattern is clearly visible in the Laplacian result, which is an indication of a poor interpolation method.

We can obtain a significantly better image (Figure 13) by replacing the Laplacian filter with a two-dimensional prediction-error filter estimated from the input data. The result in the left plot of Figure 13 was obtained after 200 conjugate-gradient iterations. If we stop after 20 iterations, the output (the right plot in Figure 13) shows only a small deviation from the input data. Large areas of the image remain unfilled. At each iteration, the interpolation process progresses only to the length of

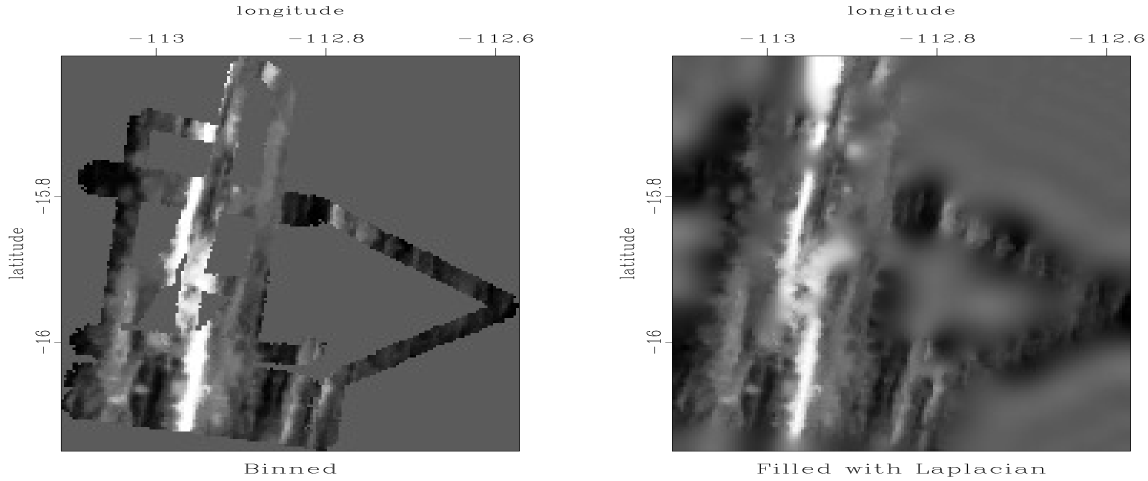


Figure 12: On the left, the SeaBeam data: the depth of the ocean under ship tracks; on the right, an interpolation with the Laplacian filter.

the filter.

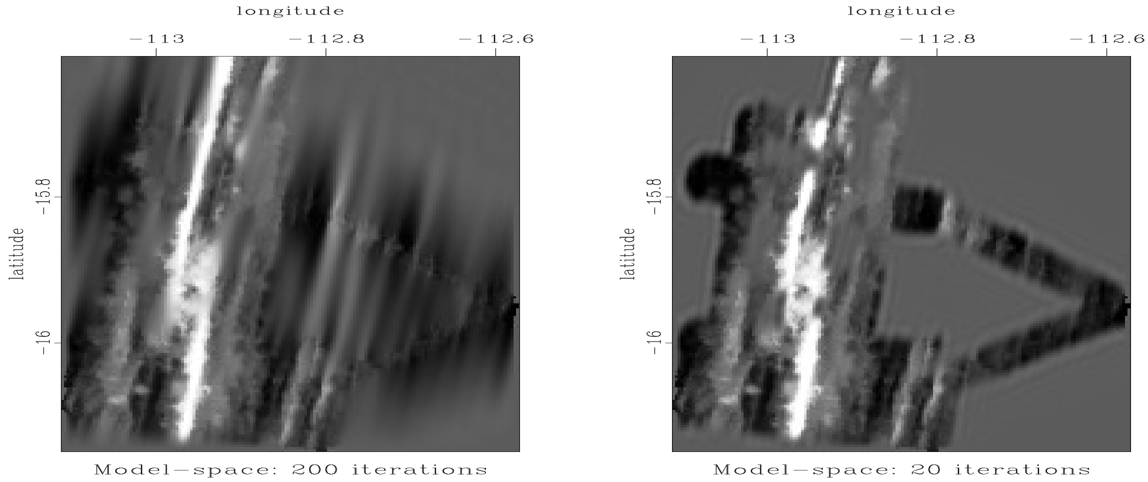
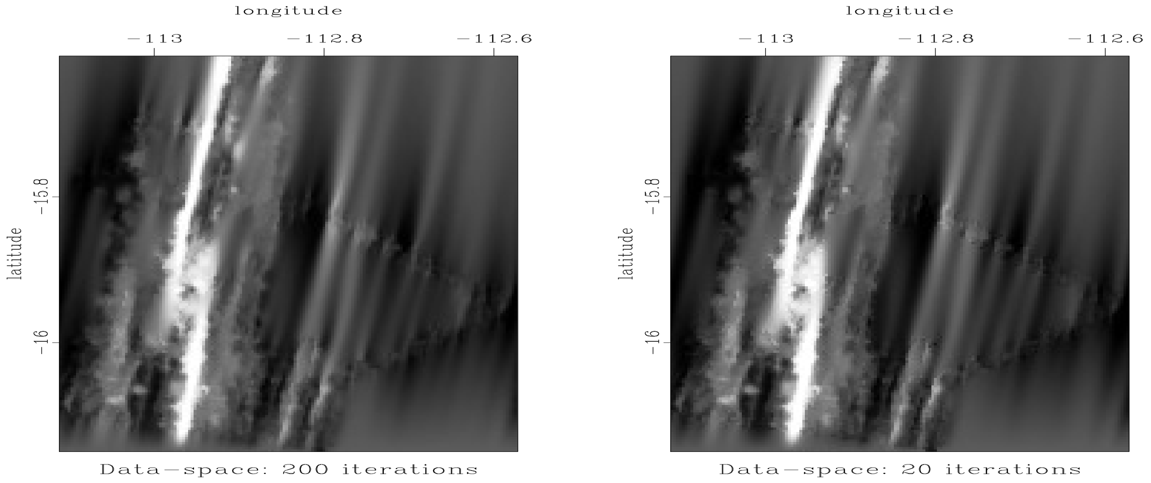


Figure 13: SeaBeam interpolation with the prediction-error filter. The left plot was taken after 200 conjugate-gradient iterations; the right, after 20 iterations.

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 14) is similar to the result of the corresponding unpreconditioned (model-space) interpolation. However, the output after just 20 iterations (the right plot in Figure 14) is already fairly close to the solution.

For a more practical test, we chose the North Sea seismic reflection dataset, previously used for testing azimuth moveout and common-azimuth migration (Biondi et al., 1998; Biondi, 1997). Figure 15 shows the highly irregular midpoint geometry



Data−space: 200 iterations

Data−space: 20 iterations

Figure 14: SeaBeam interpolation with the inverse prediction-error filter. The left plot was taken after 200 conjugate-gradient iterations; the right, after 20 iterations.

for a selected in-line and cross-line offset bin in the data. The data irregularity is also evident in the bin fold map, shown in Figure 16. The goal of data regularization is to create a regular data cube at the specified bins from the irregular input data, which have been preprocessed by normal moveout without stacking.

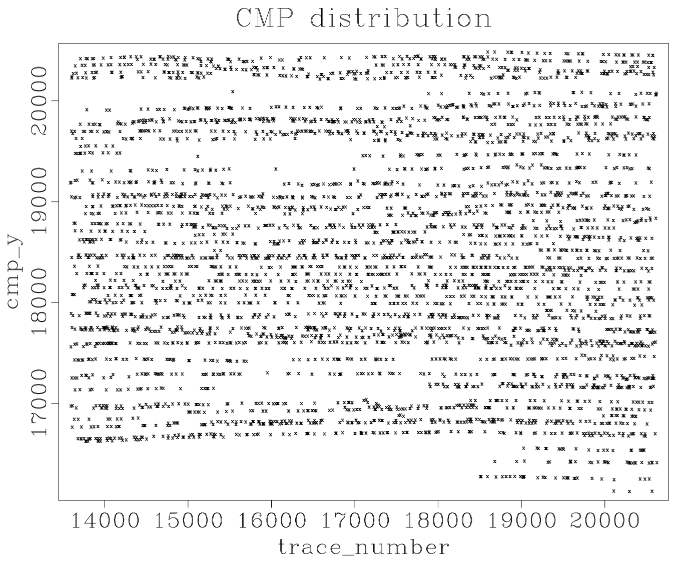

CMP distribution

Figure 15: Midpoint distribution for a 50 by 50 m offset bin in the 3-D North Sea dataset.

The data cube after normalized binning is shown in Figure 17. Binning works reasonably well in the areas of large fold but fails to fill the zero fold gaps and has an overall limited accuracy.

For efficiency, we perform regularization on individual time slices. Figure 18 shows the result of regularization using bi-linear interpolation and smoothing preconditioning (data-space regularization) with the minimum-phase Laplacian filter (Fomel et al., 2002). The empty bins are filled in a consistent manner but the data quality is dis-

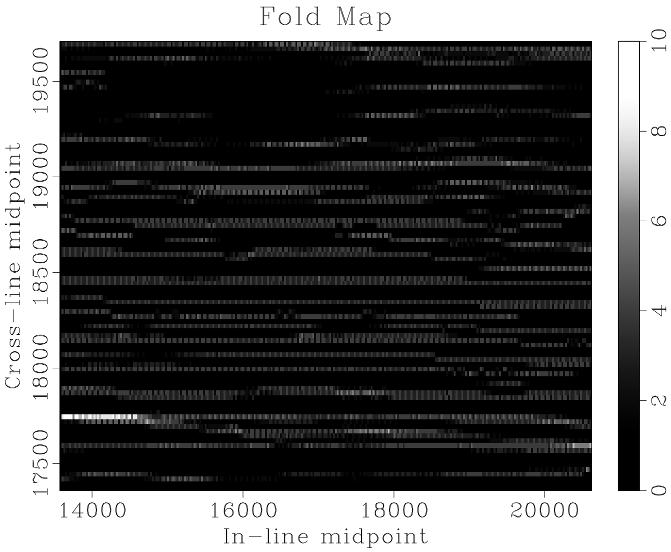


Figure 16: Map of the fold distribution for the 3-D data test.

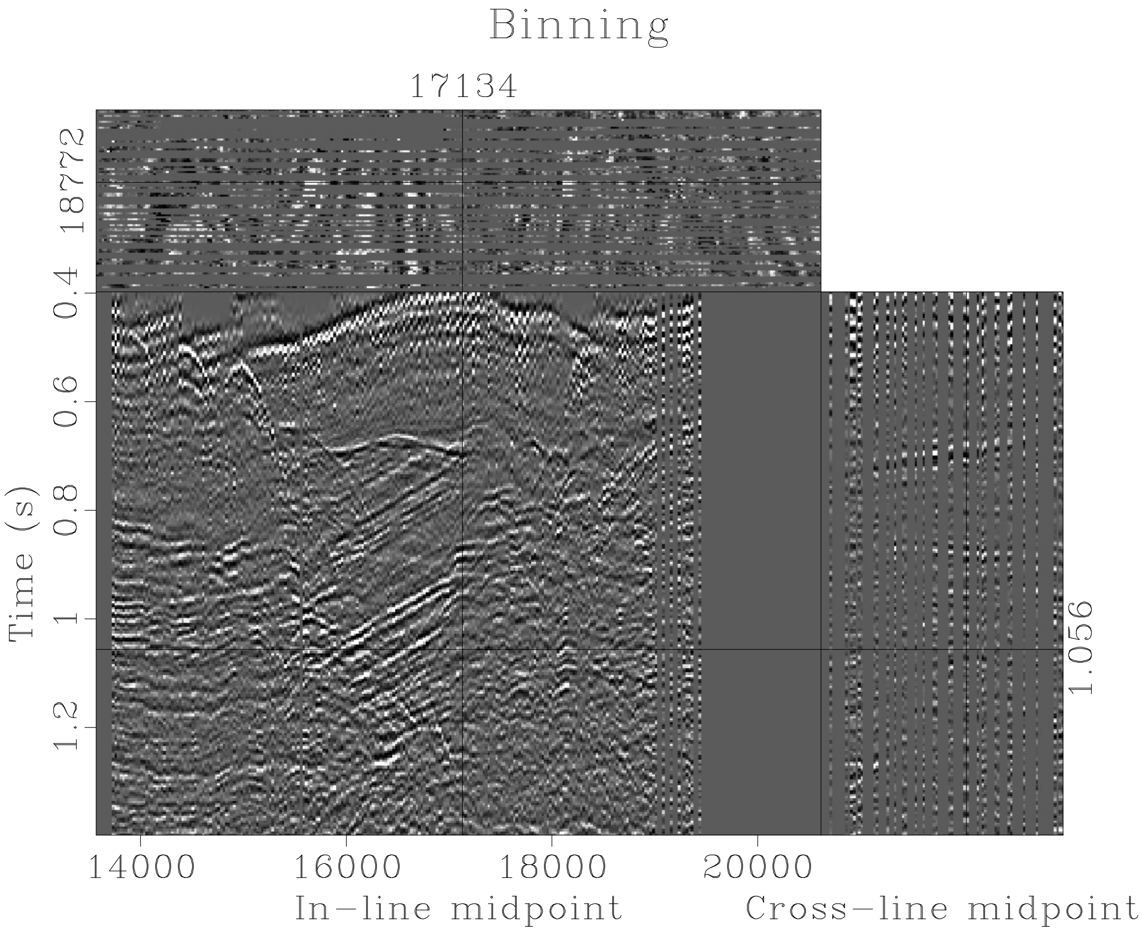


Figure 17: 3-D data after normalized binning.

torted because simple smoothing fails to characterize the complicated data structure. Instead of continuous events, we see smoothed blobs in the time slices. The events in the in-line and cross-line sections are also not clearly pronounced.

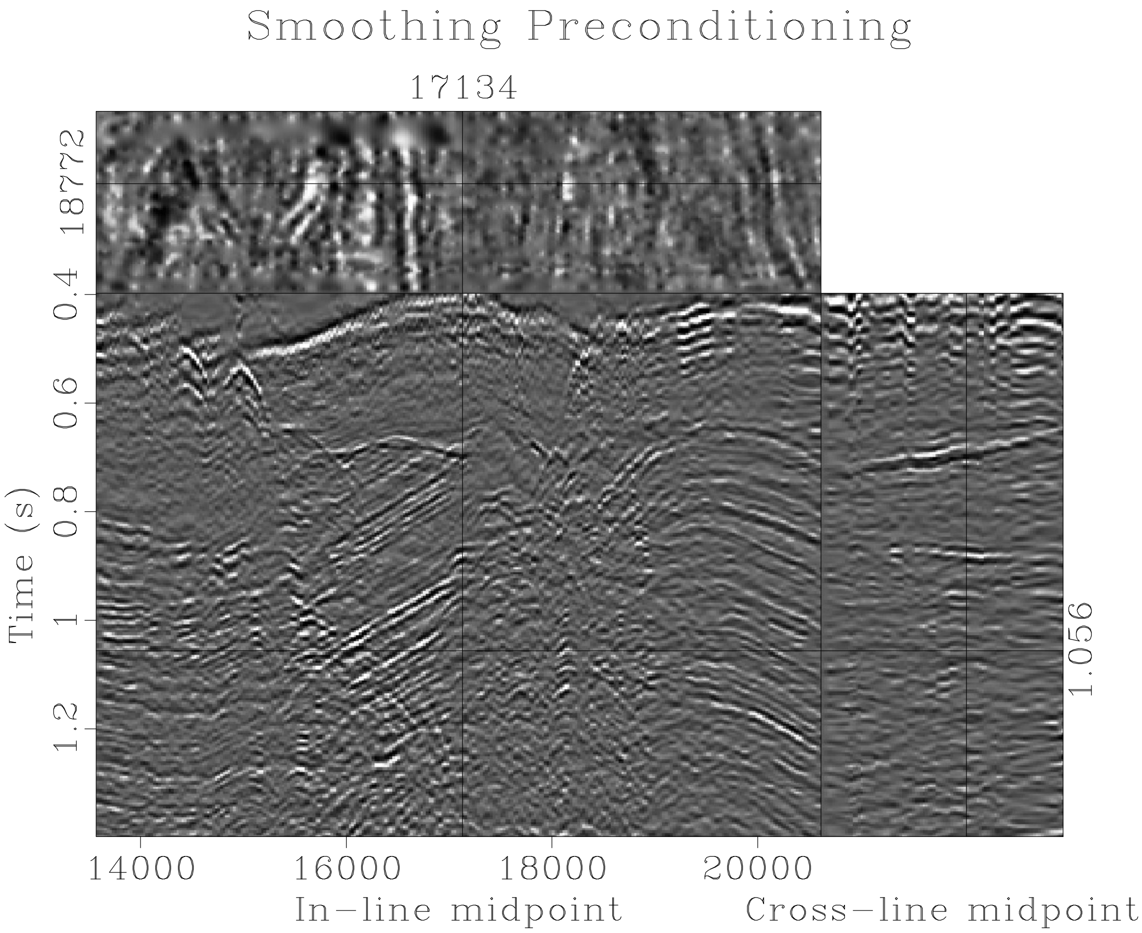


Figure 18: 3-D data regularized with bi-linear interpolation and smoothing preconditioning.

We can use the smoothing regularization result to estimate the local dips in the data, design invertible local plane-wave destruction filters (Fomel, 2001), and repeat the regularization process. Inverse interpolation with plane-wave data-space regularization is shown in Figure 19. The result is noticeably improved: the continuous reflection events become clearly visible in the time slices. Despite the irregularities in the input data, the regularization result preserves both flat reflection events and steeply-dipping diffractions. Preserving diffractions is important for correct imaging of sharp edges in the subsurface structure (Biondi et al., 1998).

For simplicity, we assumed only a single local dip component in the data. This assumption degrades the result in the areas of multiple conflicting dips, such as the intersections of plane reflections and hyperbolic diffractions in Figure 19. One could improve the image by considering multiple local dips. Fomel (2002) describes an alternative offset-continuation approach, which uses a physical connection between neighboring offsets instead of assuming local continuity in the midpoint domain.

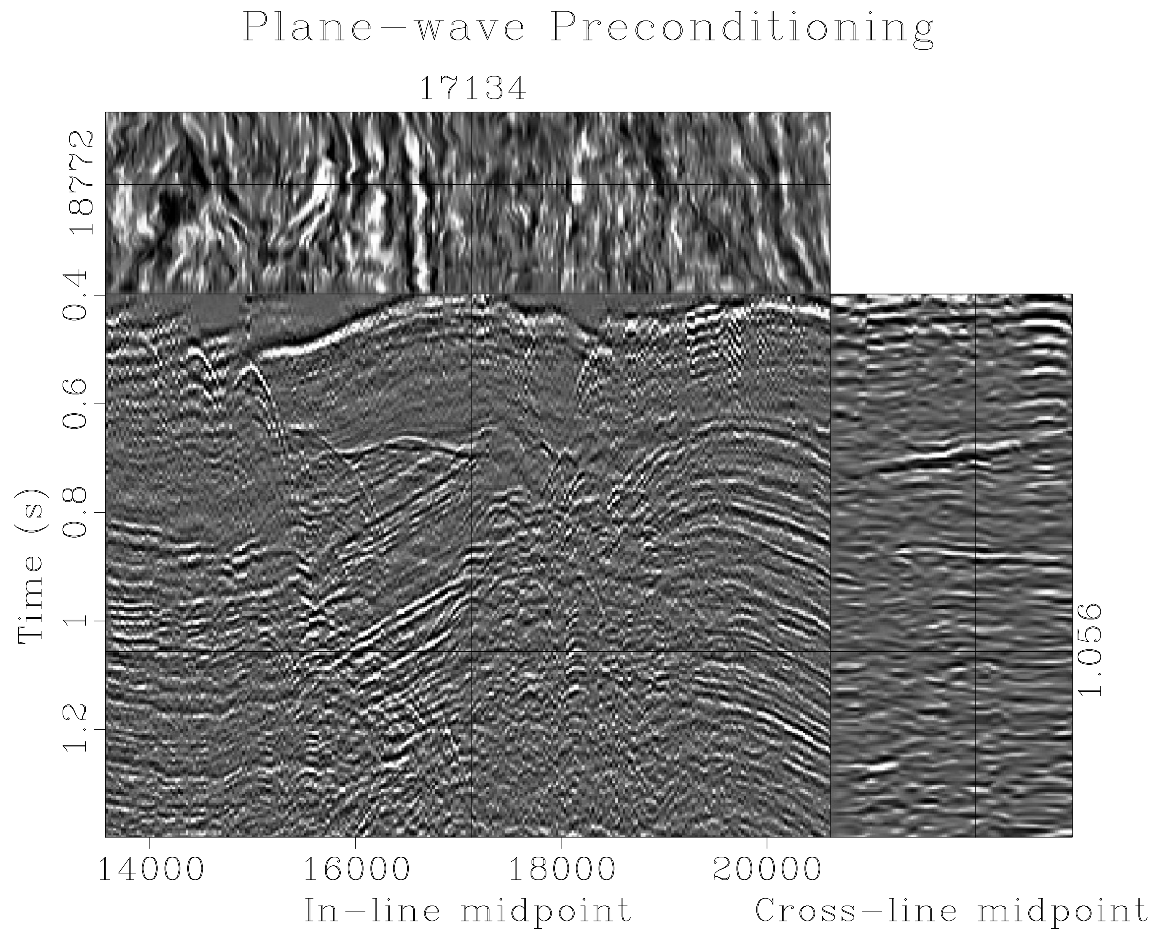


Figure 19: 3-D data regularized with cubic B-spline interpolation and local plane-wave preconditioning.

The 3-D results of this paper were obtained with an efficient 2-D regularization in time slices. This approach is computationally attractive because of its easy parallelization: different slices can be interpolated independently and in parallel. Figure 20 shows the interpolation result for four selected time slices. Local plane waves, barely identifiable after binning (left plots in Figure 20), appear clear and continuous in the interpolation result (right plots in Figure 20). The time slices are assembled together to form the 3-D cube shown in Figure 19.

CONCLUSIONS

Regularization is often a necessary part of geophysical estimation. Its goal is to impose additional constraints on the model and to guide the estimation process towards the desired solution.

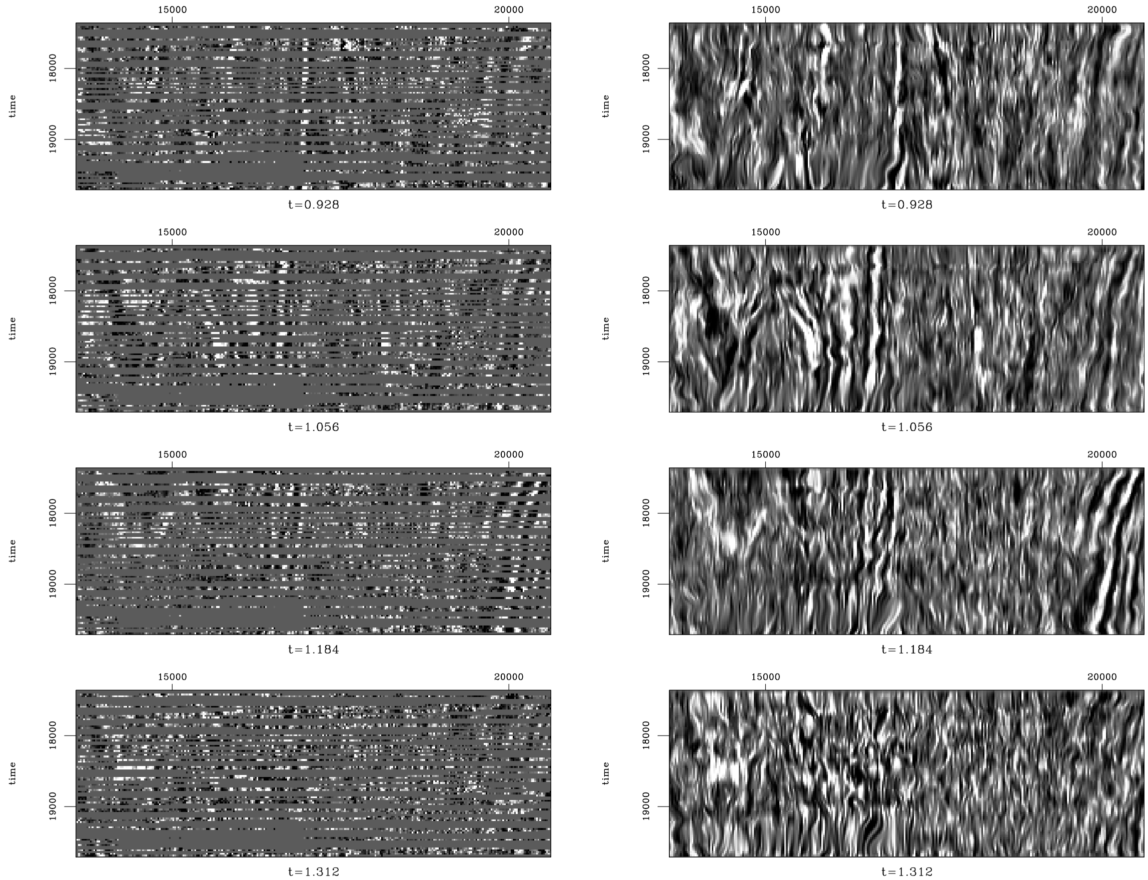


Figure 20: Selected time slices of the 3-D dataset. Left: after binning. Right: after plane-wave data regularization. The data regularization program identifies and continues local plane waves in the data.

We have considered two different regularization methods. The first, model-space approach involves a convolution operator that enhances the undesired features in the model. The second, data-space, approach involves inverse filtering (deconvolution) to precondition the model. Although the two approaches lead to the theoretically equivalent results, their behavior in iterative estimation methods is quite different. Using several synthetic and real data examples, we have demonstrated that the second, preconditioning approach is generally preferable because it shows a significantly faster convergence at early iterations.

We suggest a constructive method for preconditioning multidimensional estimation problems using the helix transform. Applying inverse filtering operators constructed this way, 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-squares estimation.

ACKNOWLEDGMENTS

We are grateful to Jim Berryman, Bill Harlan, Dave Nichols, Gennady Ryzhikov, and Bill Symes for insightful discussions about preconditioning and optimization problems.

The financial support for this work was provided by the sponsors of the Stanford Exploration Project (SEP). The 3-D North Sea dataset was released to SEP by Conoco and its partners, BP and Mobil. The SeaBeam dataset is courtesy of Alistair Harding of the Scripps Institution of Oceanography.

REFERENCES

Biondi, B., 1997, Azimuth moveout + common-azimuth migration: Cost-effective prestack depth imaging of marine data: 67th Ann. Internat. Mtg, Soc. of Expl. Geophys., 1375–1378.

Biondi, B., S. Fomel, and N. Chemingui, 1998, Azimuth moveout for 3-D prestack imaging: Geophysics, **63**, 574–588.

Claerbout, J., 1976, Fundamentals of geophysical data processing: Blackwell.

———, 1992, Earth Soundings Analysis: Processing Versus Inversion: Blackwell Scientific Publications.

———, 1998, Multidimensional recursive filters via a helix: Geophysics, **63**, 1532–1541.

———, 1999, Geophysical estimation by example: Environmental soundings image enhancement: Stanford Exploration Project.

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

———, 2002, Seismic reflection data interpolation with differential offset and shot continuation: submitted for publication in Geophysics.

Fomel, S., P. Sava, J. Rickett, and J. Claerbout, 2002, The Wilson-Burg method of spectral factorization with application to helical filtering: submitted to Geophysical Prospecting.

Harlan, W. S., 1995, Regularization by model redefinition: http://billharlan.com/pub/papers/regularization.pdf.

Hestenes, M. R., and E. Steifel, 1952, Methods of conjugate gradients for solving linear systems: J. Res. Nat. Bur. Stand., **49**, 409–436.

Paige, C. C., and M. A. Saunders, 1982, Algorithm 583, LSQR: Sparse linear equations and least squares problems: Assn. Comp. Mach. Trans. Mathematical Software, **8**, 195–209.

Parker, R. L., 1994, Geophysical inverse theory: Princeton Univ. Press.

Tikhonov, A. N., and V. Y. Arsenin, 1977, Solution of ill-posed problems: John Wiley and Sons.

Vandecar, J. C., and R. Snieder, 1994, Obtaining smooth solutions to large, linear, inverse problems: Geophysics, **59**, 818–829.