Chapter 1

Iterative data regularization

According to the theoretical conclusions of Chapter **??**, data regularization can be formulated as an optimization problem. In fact, there are two theoretically equivalent formulations: *model-space* regularization and *data-space* regularization. The former is closely related to Tikhonov's regularization for ill-posed inverse problems (Tikhonov and Arsenin, 1977). Mathematically, it extends the data space and constructs a composite column operator. Data-space regularization extends the model space and constructs a composite row operator. It leads to the concept of model preconditioning (Nichols, 1994).

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 that those of the model-space matrix, and the convergence of the iterative conjugate-gradient iteration is correspondingly faster. 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. The first iteration of the model-space regularization yields $\mathbf{L}^T\mathbf{d}$ as the model estimate regardless of the regularization operator $\mathbf{D}$, while the first iteration of the data-space regularization yields $\mathbf{C}\mathbf{L}^T\mathbf{d}$, which is an already "simplified" version of the model. Since iteration to the exact solution is never achieved in the large-scale problems, the results of iterative optimization may turn out quite differently. Harlan (1995) points out that the two components of the model-space regularization [Equations (**??**) and (**??**)] 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:

> 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 chapter, I 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. A particularly convenient method is preconditioning by recursive filtering, which is extended to the multidimensional case with the help of Claerbout's helix transform (Claerbout, 1998a). Invertible multidimensional filters can be created by helical spectral factorization.

ONE-DIMENSIONAL SYNTHETIC EXAMPLES

The simple test examples in this section are borrowed from Claerbout (1999).

In the first example, the input data were randomly subsampled (with decreasing density) from a sinusoid (Figure 1.1). The forward operator $\mathbf{L}$ in this case is linear interpolation. In other words, we seek a regularly sampled model on 200 grid points that could predict the data

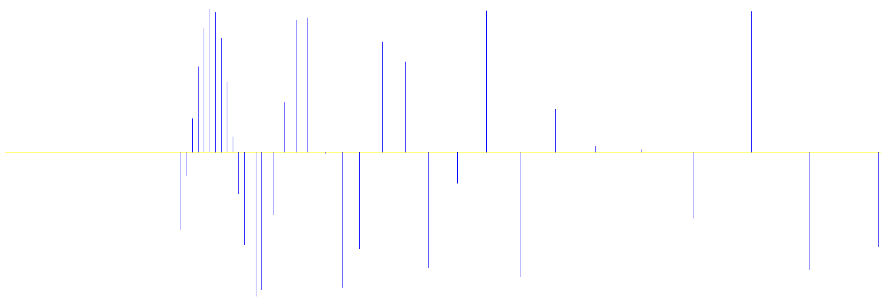

Figure 1.1: The input data (right) are irregularly spaced samples of a sinusoid (left). $\boxed{\text{optim-data}}$ [ER]

with a forward linear interpolation. Sparse irregular distribution of the input data makes the regularization enforcement a necessity. Following Claerbout (1999), I applied convolution with the simple $(1,-1)$ difference filter as the operator $\mathbf{D}$ that forces model continuity (the first-order spline). An appropriate preconditioner $\mathbf{P}$ in this case is recursive causal integration. Figures 1.2 and 1.3 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.

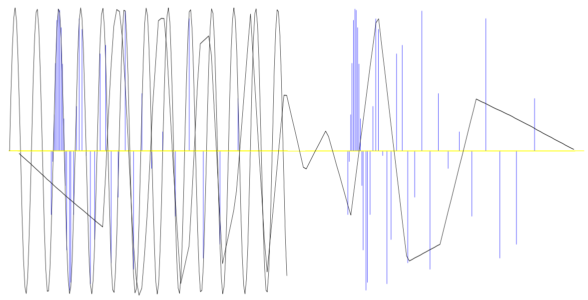

Figure 1.2: Estimation of a continuous function by the model-space regularization. The difference operator $\mathbf{D}$ is the derivative operator (convolution with $(1,-1)$). $\boxed{\text{optim-im1}}$ [ER,M]

As expected from the general theory, the model preconditioning provides a much faster rate of convergence. I measured the rate of convergence using the model residual, which is a distance from the current model to the final solution. Figure 1.5 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 each method is roughly equal, the computational economy is evident. Figure 1.4 shows the final solution, and

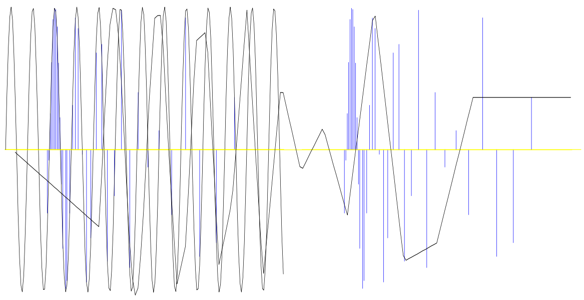

Figure 1.3: Estimation of a continuous function by the data-space regularization. The preconditioning operator **P** is causal integration. optim-fm1 [ER,M]

the estimates from model- and data-space regularization after only 5 iterations of conjugate directions. The data-space estimate looks much closer to the final solution than its competitor.

Figure 1.4: 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. optim-early1 [ER]

Changing the preconditioning operator changes the regularization result. Figure 1.6 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 $\frac{(1-Z^N)(1-Z^{-N})}{(1-Z)(1-Z^{-1})}$ (Claerbout, 1992). I chose the filter length $N = 6$.

If, instead of looking for a smooth interpolation, we want to limit the number of frequency components, then the best 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 could be the three-point *recursive* filter $P(Z) = 1/(1 + a_1 Z + a_2 Z^2)$. To test

Figure 1.5: 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. optim-schwab1 [ER]

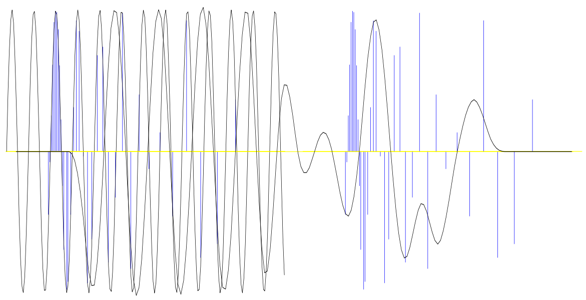

Figure 1.6: Estimation of a smooth function by the data-space regularization. The preconditioning operator **P** is a triangle smoother. optim-fm6 [ER,M]

this idea, I estimated the PEF $D(Z)$ from the output of inverse linear interpolation (Figure 1.3), and ran the data-space regularized estimation again, substituting the recursive filter $P(Z) = 1/D(Z)$ in place of the causal integration. I repeated this two-step procedure three times to get a better estimate for the PEF. The result, shown in Figure 1.7, exhibits the desired mono-frequency output.

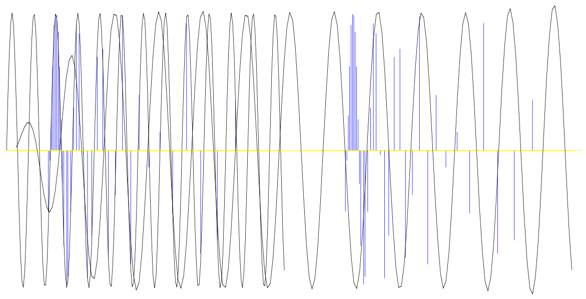

Figure 1.7: Estimation of a mono-frequency function by the data-space regularization. The preconditioning operator **P** is a recursive filter (the inverse of PEF). optim-pm1 [ER,M]

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. Since 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. 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

$$\mathbf{D}\mathbf{m} \approx \mathbf{0}\,, \tag{1.1}$$

where $\mathbf{m}$ is the data vector and $\mathbf{D}$ is the specified filter. Equation (1.1) is completely equivalent to equation (**??**). The approximate equality sign means that equation (1.1) 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 $\mathbf{K}$, which can be considered as a diagonal matrix with zeros at the missing data locations and ones elsewhere, we can rewrite equation (1.1) in the extended form

$$\mathbf{D}(\mathbf{I}-\mathbf{K})\mathbf{m} \approx -\mathbf{D}\mathbf{K}\mathbf{m} = -\mathbf{D}\mathbf{m}_k\,, \tag{1.2}$$

in which $\mathbf{I}$ is the identity operator, and $\mathbf{m}_k$ represents the known portion of the data. It is important to note that equation (1.2) corresponds to the limiting case of the regularized linear system

$$\begin{cases} \mathbf{Km} = \mathbf{m}_k \ , \\ \epsilon \mathbf{Dm} \approx \mathbf{0} \end{cases} \tag{1.3}$$

for the scaling coefficient ϵ approaching zero. System (1.3) is equivalent to system (**??**-**??**) with the masking operator **K** playing the role of the forward interpolation operator **L**. Setting ϵ to zero implies putting far more weight on the first equation in (1.3) and using the second equation only to constrain the null space of the solution. Applying the general theory of data-space regularization from Chapter **??**, we can immediately transform system (1.3) to the equation

$$\mathbf{KPp} \approx \mathbf{m}_k \ , \tag{1.4}$$

where **P** is a preconditioning operator, and **p** is the preconditioning variable, connected with **m** by the simple relationship

$$\mathbf{m} = \mathbf{Pp} \ .$$

According to equations (**??**) and (**??**) from Chapter **??**, equations (1.4) and (1.2) have exactly the same solutions if the following condition is satisfied:

$$\mathbf{P}\mathbf{P}^T = \left(\mathbf{D}^T \mathbf{D}\right)^{-1} \ , \tag{1.5}$$

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

$$\mathbf{P} = \mathbf{D}^{-1} \ . \tag{1.6}$$

I illustrate the missing data problem with a simple 1-D synthetic data test taken from Claerbout (1999). Figure 1.8 shows the interpolation results of the unpreconditioned technique with three different filters. For comparison with the preconditioned scheme, I 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.

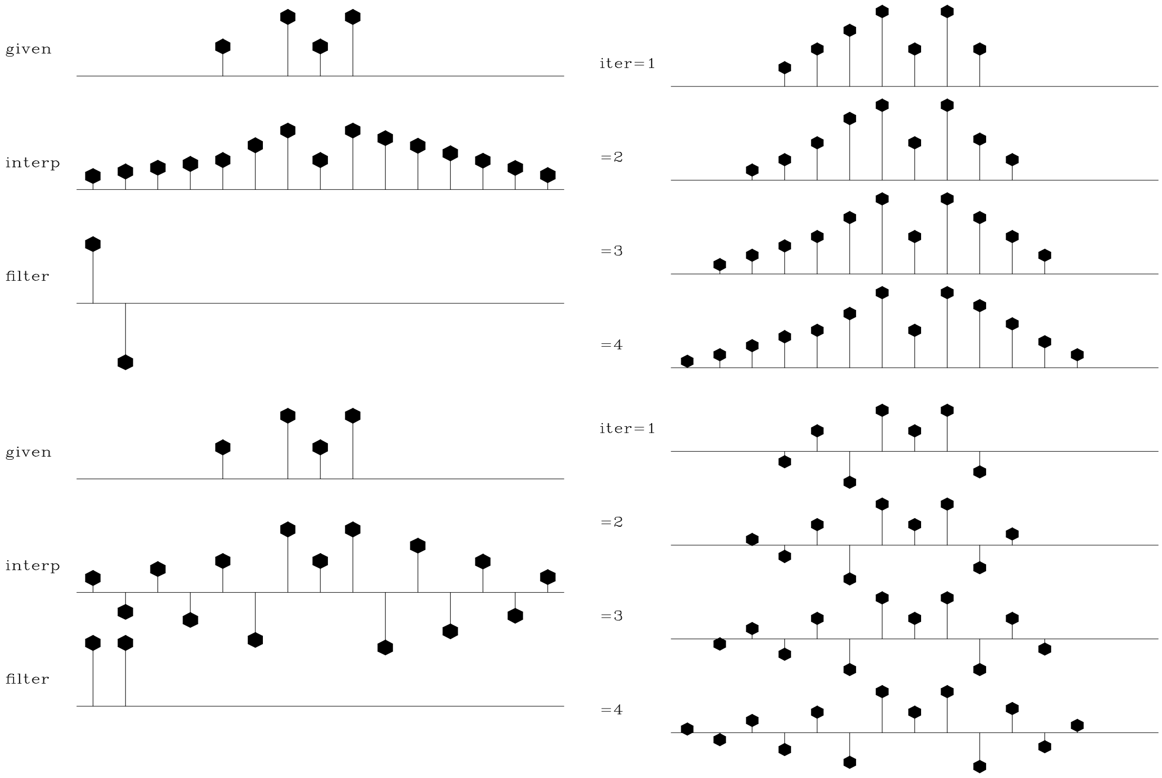

Figure 1.8: Unpreconditioned interpolation with two different regularization filters. Left plot: the top shows the input data; the middle, the result of interpolation; the bottom, the filter. The right plot shows the convergence process for the first four iterations. optim-mall [ER]

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

Preconditioned interpolation (Figure 1.9) 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. The final output of interpolation with recursive deconvolution preconditioning 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.

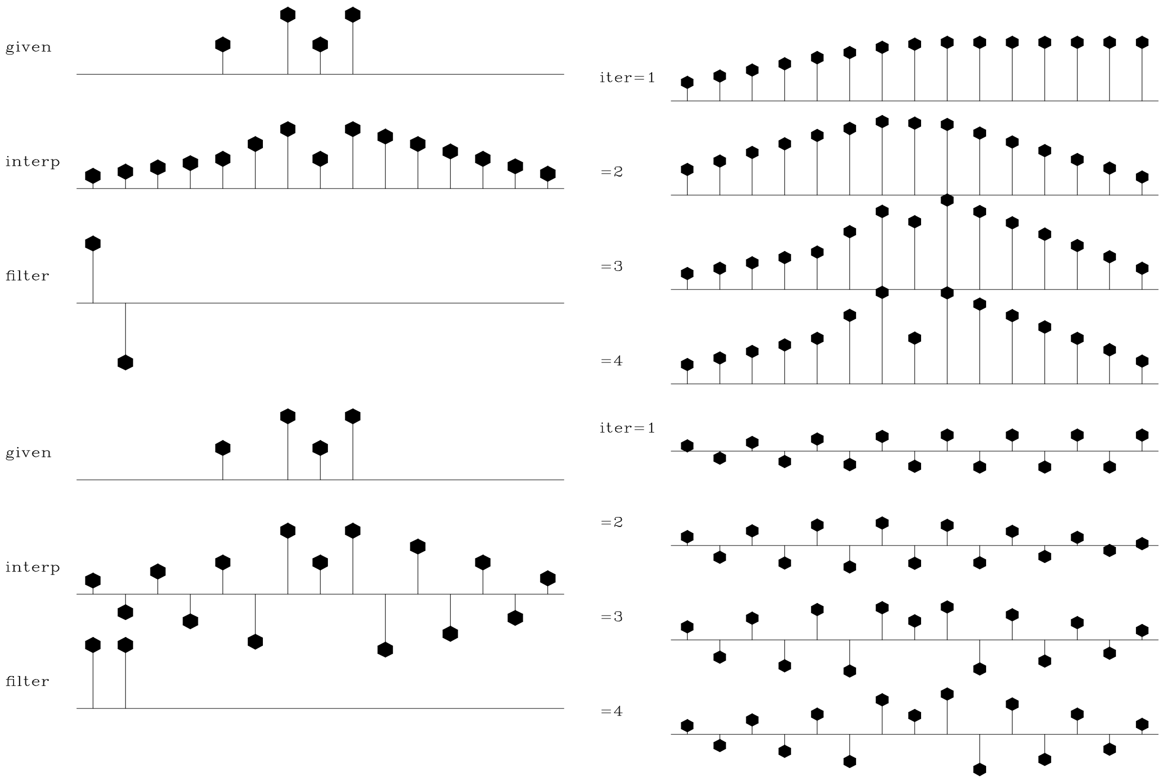

Figure 1.9: Interpolation with preconditioning. Left plot: the top shows the input data; the middle, the result of interpolation; the bottom, the filter. The right plot shows the convergence process for the first four iterations. optim-sall [ER]

MULTIDIMENSIONAL RECURSIVE FILTER PRECONDITIONING WITH HELIX TRANSFORM

Claerbout (1997, 1998a,c) 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 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.

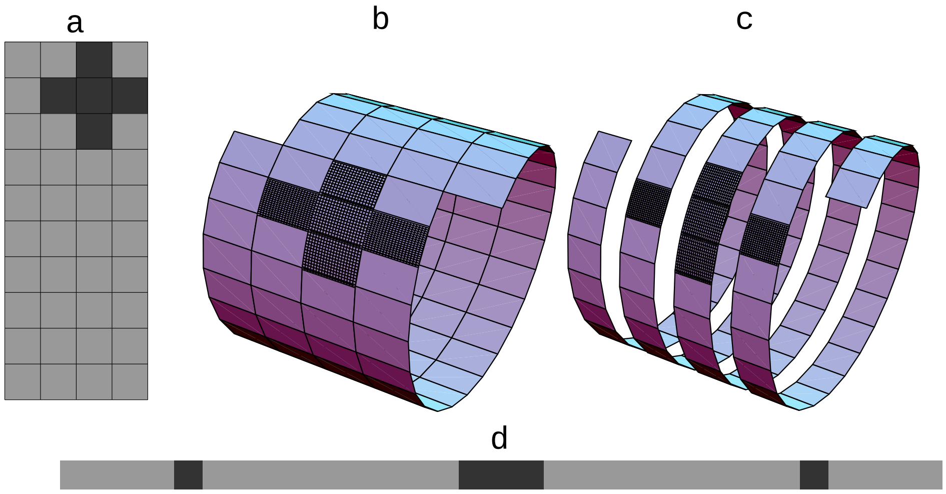

Figure 1.10: The helix transform of two-dimensional filters to one dimension. The two-dimensional filter in the left plot is equivalent to the one-dimensional filter in the right plot, assuming that a shifted periodic condition is imposed on one of the axes. optim-helix1 [CR]

The helix filtering idea is schematically illustrated in Figure 1.10. 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 ignored in a helical convolution program. The computational gain is not, however, in the convolution itself, but in the ability to perform recursive inverse filtering (deconvolution) in multiple dimensions. A multidimensional 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.

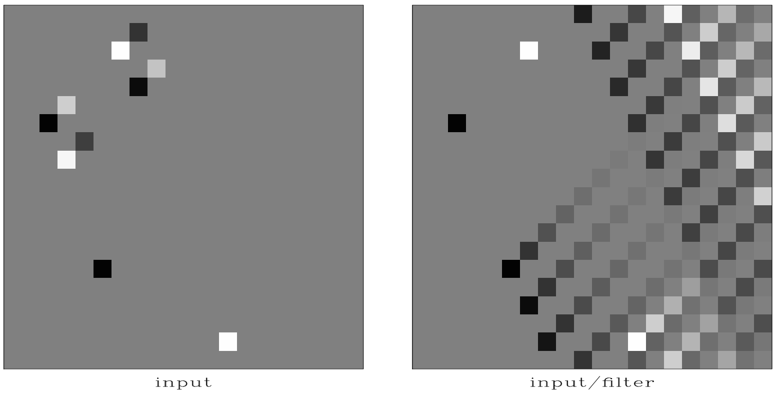

Figure 1.11: Illustration of 2-D deconvolution with helix transform. Left is the input: two spikes and two filters. Right is the output of deconvolution. optim-waves [ER]

An example of two-dimensional recursive filtering is shown in Figure 1.11. 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 1.11). Helical wrap-around, visible on the horizontal boundaries, indicates the direction of the helix. Claerbout (1999) presents more examples and discusses all the issues of multidimensional helical deconvolution in detail.

As is known from the one-dimensional theory (Claerbout, 1976), a stable recursive filtering requires a minimum-phase filter, which can be constructed with a spectral factorization algorithm. The Wilson-Burg spectral factorization method, described in the next section, is particularly convenient for helical filtering.

Wilson-Burg spectral factorization

Spectral factorization constructs a minimum-phase signal from its spectrum. The algorithm, suggested by Wilson (1969), approaches this problem directly with Newton's iterative method. In a Z-transform notation, Wilson's method implies solving the equation

$$S(Z) = A(Z)\bar{A}(1/Z) \tag{1.7}$$

for a given spectrum $S(Z)$ and unknown minimum-phase signal $A(Z)$ with an iterative linearization

$$\begin{aligned} S(Z) &= A_t(Z)\bar{A}_t(1/Z) + A_t(Z)[\bar{A}_{t+1}(1/Z) - \bar{A}_t(1/Z)] + \bar{A}_t(1/Z)[A_{t+1}(Z) - A_t(Z)] \\ &= A_t(Z)\bar{A}_{t+1}(1/Z) + \bar{A}_t(1/Z)A_{t+1} - A_t(Z)\bar{A}_t(1/Z)\,, \end{aligned} \tag{1.8}$$

where $A_t(Z)$ denotes the signal estimate at iteration t. Starting from some initial estimate $A_0(Z)$, such as $A_0(Z) = 1$, one iteratively solves the linear equation (1.8) for the updated signal $A_{t+1}(Z)$. Wilson (1969) presents a rigorous proof that iteration (1.8) operates with minimum-phase signals provided that the initial estimate $A_0(Z)$ is minimum-phase.

Burg (1998, personal communication) recognized that dividing both sides of equation (1.8) by $\bar{A}_t(1/Z)A_t(Z)$ leads to a particularly convenient form, where the terms on the left are symmetric, and the two terms on the right are correspondingly strictly causal and anticausal:

$$1 + \frac{S(Z)}{\bar{A}_t(1/Z)\, A_t(Z)} = \frac{A_{t+1}(Z)}{A_t(Z)} + \frac{\bar{A}_{t+1}(1/Z)}{\bar{A}_t(1/Z)} \tag{1.9}$$

Equation (1.9) leads to the Wilson-Burg algorithm, which accomplishes spectral factorization by a recursive application of convolution (polynomial multiplication) and deconvolution (polynomial division). The algorithm proceeds as follows:

1. Compute the left side of equation (1.9) using forward and adjoint polynomial division.

2. Abandon negative lags, to keep only the causal part of the signal, and also keep half of the zero lag. This gives us $A_{t+1}(Z)/A_t(Z)$.

iter	a_0	a_1	a_2	a_3
0	1.000000	0.000000	0.000000	0.000000
1	36.523964	23.737839	6.625787	0.657103
2	26.243151	25.726116	8.471050	0.914951
3	24.162354	25.991493	8.962727	0.990802
4	24.001223	25.999662	9.000164	0.999200
5	24.000015	25.999977	9.000029	0.999944
6	23.999998	26.000002	9.000003	0.999996
7	23.999998	26.000004	9.000001	1.000000
8	23.999998	25.999998	9.000000	1.000000
9	24.000000	26.000000	9.000000	1.000000

Table 1.1: Example convergence of the Wilson-Burg iteration

3. Multiply out (convolve) the denominator $A_t(Z)$. Now we have the desired result $A_{t+1}(Z)$.

4. Iterate until convergence.

An example of the Wilson-Burg convergence is shown in Table 1.1 on a simple 1-D signal. The autocorrelation $S(Z)$ in this case is $1334 + 867(Z + 1/Z) + 242\left(Z^2 + 1/Z^2\right) + 24\left(Z^3 + 1/Z^3\right)$, and the corresponding minimum-phase signal is $A(Z) = (2+Z)(3+Z)(4+Z) = 24+26Z+9Z^2+Z^3$. A quadratic rate of convergence is visible from the table. The convergence slows down for signals whose polynomial roots are close to the unit circle (Wilson, 1969).

Comparison of Wilson-Burg and Kolmogoroff methods

The Kolmogoroff algorithm of spectral factorization (Kolmogoroff, 1939; Claerbout, 1976) is widely used because of its computationally efficiency. While this method is easily extended to the multi-dimensional case with the help of helical transform (Rickett and Claerbout, 1999a,b), there are several circumstances that make the Wilson-Burg method more attractive in multi-dimensional filtering applications.

- The Kolmogoroff method takes $O(N \log N)$ operations, where N is the length of the auto-correlation function. The cost of the Wilson-Burg method is proportional to the [number of iterations] $\times$ [filter length] $\times N$. If we keep the filter small and limit the number of iterations, the Wilson-Burg method can be cheaper (linear in N).

- The Kolmogoroff method works in the frequency domain and assumes periodic boundary conditions. Auto-correlation functions, therefore, need to be padded with zeros before they are Fourier transformed. For functions with zeros near the unit circle, the padding may need to be many orders of magnitude greater than the original filter length, N (Rickett and Claerbout, 1998). The Wilson-Burg method is implemented in the time-domain, so no padding is required.

- Newton's method (the basis of the Wilson-Burg algorithm) converges quickly when the initial guess is close to the solution. If we take advantage of this property, the method may converge in one or two iterations, reducing the cost even further. It is impossible to make use of an initial guess with the Kolmogoroff method.

- The Kolmogoroff method, when applied to helix filtering, involves the dangerous step of truncating the filter coefficients to reduce the size of the filter. If the auto-correlation function has roots close to the unit circle, truncating filter coefficients may easily lead to non-minimum-phase filters. With Wilson-Burg, we can fix the shape of the filter from the very beginning. This does not guarantee that we will find the exact solution, but at least we can obtain a reasonable minimum-phase approximation to the desired filter. The safest practical strategy in the case of an unknown initial estimate is to start with finding the longest possible filter, remove those of its coefficients that are smaller a certain threshold, and repeat the factoring process again with the shorter filter.

Factorization examples

The first simple example of helical spectral factorization is shown in Figure 1.12. A minimum-phase factor is found by spectral factorization of its autocorrelation. The result is additionally confirmed by applying inverse recursive filtering, which turns the filter into a spike (the right-most plot in Figure 1.12.)

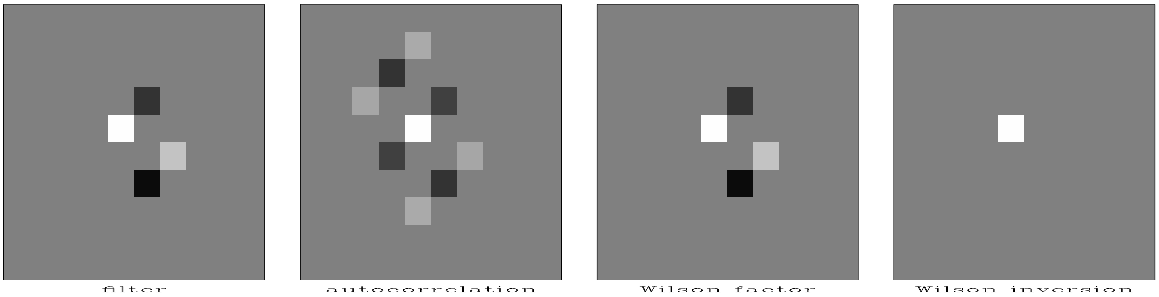

Figure 1.12: Example of 2-D Wilson-Burg factorization. From left to right: the input filter; its auto-correlation; the factor obtained by the Wilson-Burg method; the result of the deconvolution. optim-autowaves [ER]

A practically useful example is depicted in Figure 1.13. The symmetric Laplacian operator is often used in practice for regularizing smooth data (see a more detailed discussion in Chapter 5). In order to construct a corresponding recursive preconditioner, I factor the Laplacian auto-correlation (the biharmonic operator) using the Wilson-Burg algorithm. Figure 1.13 shows the resultant filter. The minimum-phase Laplacian filter has several times more coefficients that the original Laplacian. Therefore, its application would be more expensive in a convolution application. The real advantage follows from the applicability of the minimum-phase filter for inverse filtering (deconvolution). As demonstrated by 2-D examples later in this chapter, the gain in convergence from recursive filter preconditioning outweighs the loss of efficiency from the longer filter. Figure 1.14 shows a construction of the smooth inverse impulse response by application of the $\mathbf{C} = \mathbf{P}\mathbf{P}^T$ operator, where $\mathbf{P}$ is deconvolution with the minimum-phase Laplacian. The application of $\mathbf{C}$ is equivalent to a numerical solution of the biharmonic equation, discussed in Chapter 5.

TWO-DIMENSIONAL ENVIRONMENTAL DATA EXAMPLES

Using the idea of recursive filter preconditioning, which I illustrated on 1-D synthetic examples at the beginning of this chapter, and the multi-dimensional tools of the previous section, we can now proceed to multi-dimensional tests. In the first set of tests, I use simple environmental data sets. Such data are convenient for quick testing while appearing less artificial than

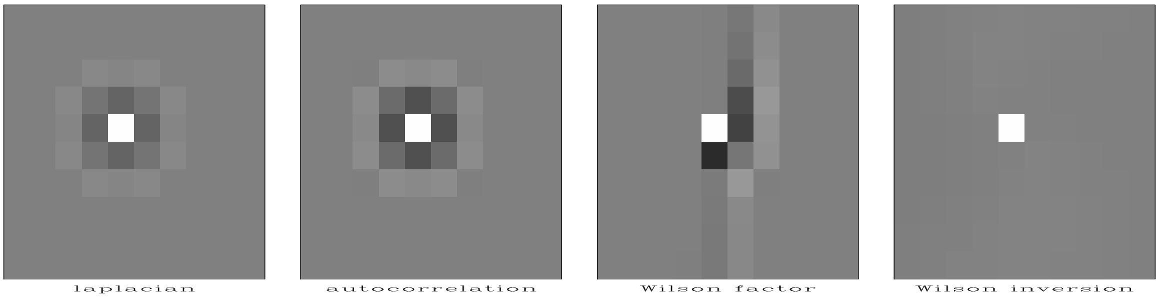

Figure 1.13: Creating a minimum-phase Laplacian filter. From left to right: Laplacian filter; its auto-correlation; the factor obtained by the Wilson-Burg method (minimum-phase Laplacian); the result of the deconvolution. optim-laplac [ER]

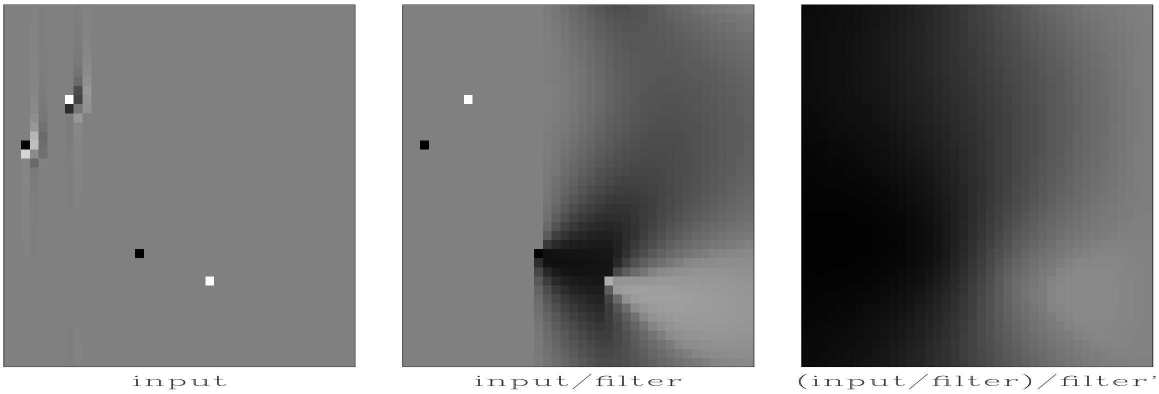

Figure 1.14: 2-D deconvolution with the minimum-phase Laplacian. Left: input. Center: output of deconvolution. Right: output of deconvolution and adjoint deconvolution (equivalent to solving the biharmonic differential equation). optim-thin42 [ER]

synthetic examples. I proceed to seismic data examples after fully developing the regularization tools in Chapter 5.

Chernobyl rainfall in Switzerland

The first test dataset contains rainfall measurements from Switzerland on the 8th of May 1986. The dataset was used in the Spatial Interpolation Comparison (Dubois, 1999) for comparing different spatial interpolation methods. Figure 1.15 shows the data area: the Digital Elevation Model of Switzerland and the country's borders[1]. A total of 467 rainfall measurements were taken. A subset of randomly selected 100 measurements was used in the 1997 Spatial Interpolation Comparison in order to compare the results with the known data. Figure 1.16 shows the spatial location of the selected data samples.

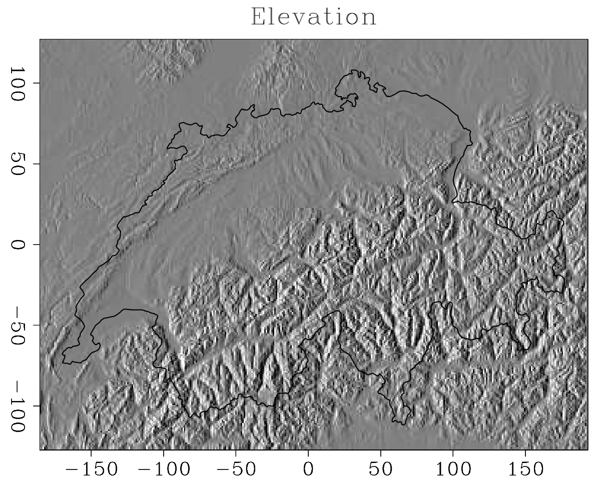

Figure 1.15: Digital Elevation Model of Switzerland and the country's borders. The country borders are extracted from the Digital Chart of the World (DCW) provided by ESRI. optim-elev [ER]

Rainfall level is generally a smoothly varying quantity. We cannot expect it to be represented a priori by a simple function. Therefore, it is reasonable to take the regularization operator **D** to be a convolution with the Laplacian filter. The corresponding preconditioning operator **P** is then a deconvolution with the minimum-phase Laplacian constructed in the previous section. The interpolation result using the model-space regularization scheme (**??**-**??**) is shown in Figure 1.17. The input irregular data were regularized on a 376 by 253 grid, which corresponds to the digital elevation model in Figure 1.15. Similarly to what happens in the

[1]I provide the elevation image only for reference. It has not been used in the interpolation experiment.

Figure 1.16: Left: data locations for all 467 measurements. Right: data locations for selected 100 measurements. optim-raindata [ER]

one-dimensional synthetic examples, the solution converges steadily but with a slow spread of information away from the known data points. It takes about $10,000$ iterations to achieve full convergence. Figure 1.18 is a correlation plot of the observed and interpolated data points for the 367 points that were not used in the interpolation experiment. If we take into account the fairly unpredictable distribution of rainfall, the correlation is relatively good in comparison with analogous results of the Spatial Interpolation Contest (Dubois, 1999).

The result of applying recursive filter preconditioning with the minimum-phase Laplacian operator is shown in Figure 1.19. Full convergence is achieved after only 100 iterations. The result after 10 iterations (the left plot in Figure 1.19) is already close to the final solution. Recursive preconditioning speeded up the iteration count by a factor of 1000. The actual gain in execution time is several times smaller because of the correspondingly longer filter, but it is still impressively large.

SeaBeam water bottom

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 the Stanford Exploration Project for benchmarking different strategies of data interpolation (Crawley, 1995a,b; Fomel, 1996a;

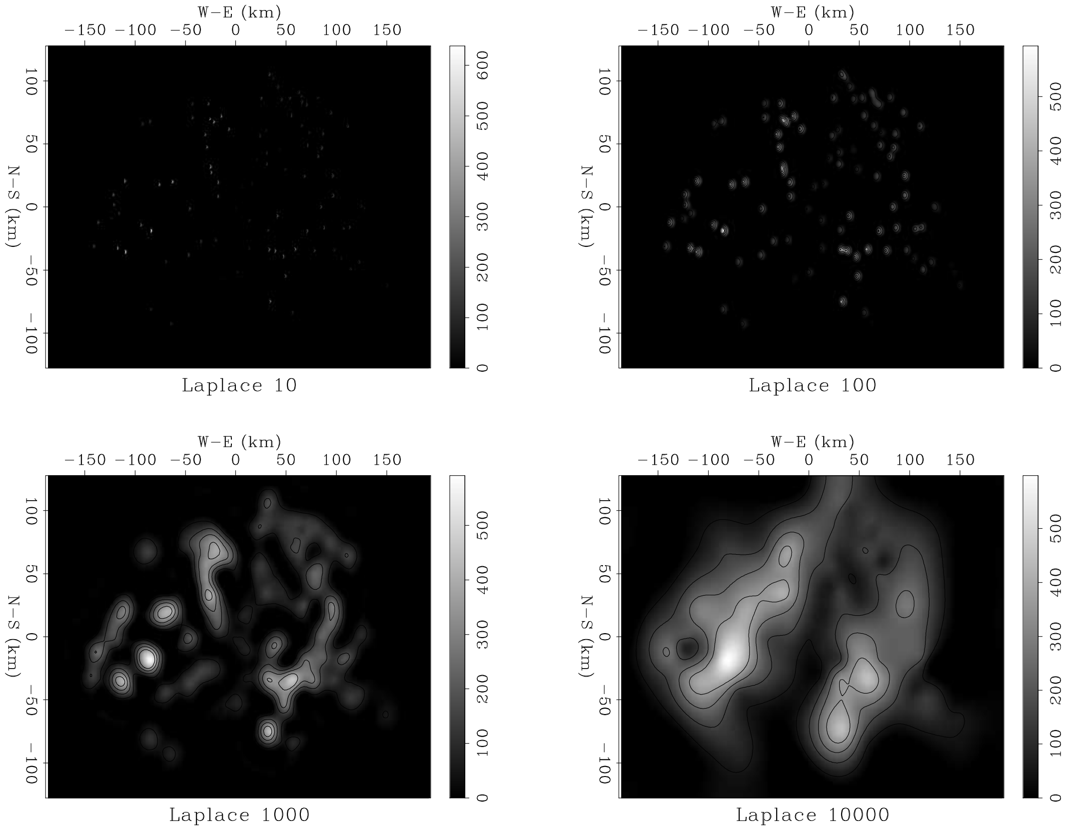

Figure 1.17: Rainfall data after model-space regularization with 10, 100, 1000, and 10000 iterations. optim-lapinter [ER]

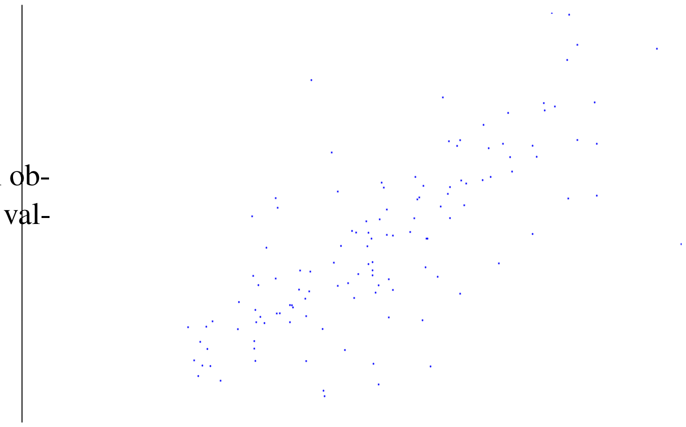

Figure 1.18: Correlation between observed and predicted rainfall data values. optim-lapstat [ER]

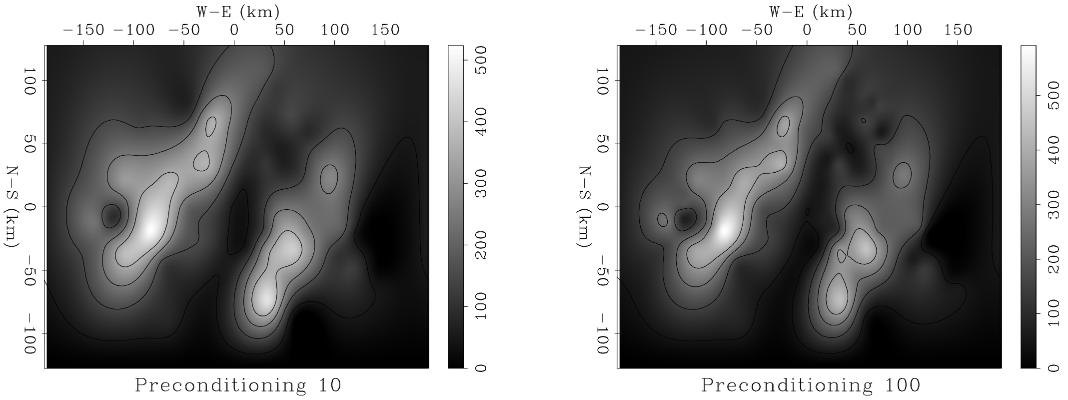

Figure 1.19: Rainfall data after data-space regularization (recursive filter preconditioning) with 10 and 100 iterations. optim-precinter [ER]

Fomel et al., 1997; Fomel, 2000a; Clapp, 2000b). The left plot in Figure 1.20 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 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.

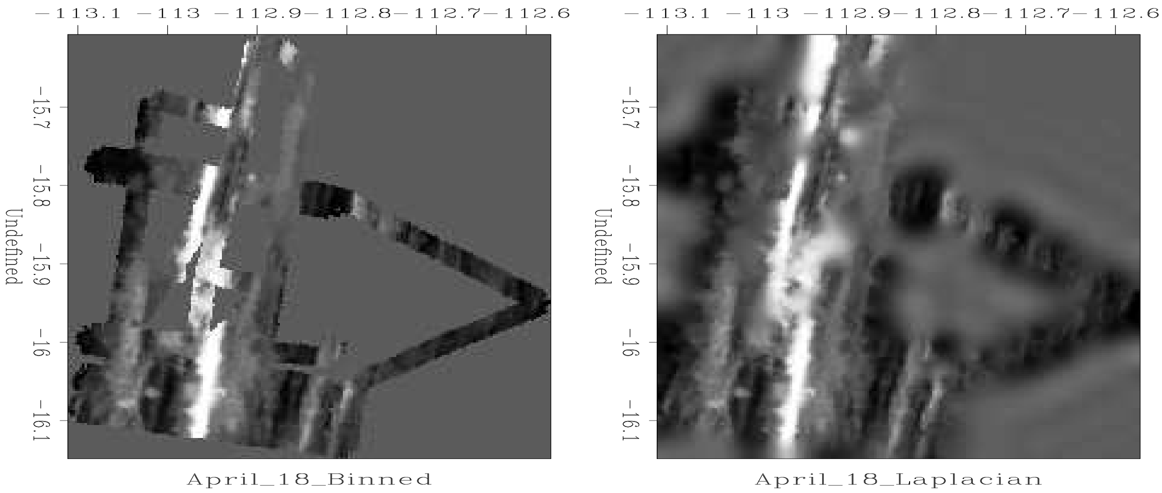

Figure 1.20: On the left, the SeaBeam data: the depth of the ocean under ship tracks; on the right, an interpolation with the Laplacian filter. optim-seabdat [ER]

We can obtain a significantly better result (Figure 1.21) by replacing the Laplacian filter with a two-dimensional prediction-error filter (PEF) estimated from the input data. The result in the left plot of Figure 1.21 was obtained after 200 conjugate-gradient iterations. If we stop after 20 iterations, the output (the right plot in Figure 1.21) 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 the filter.

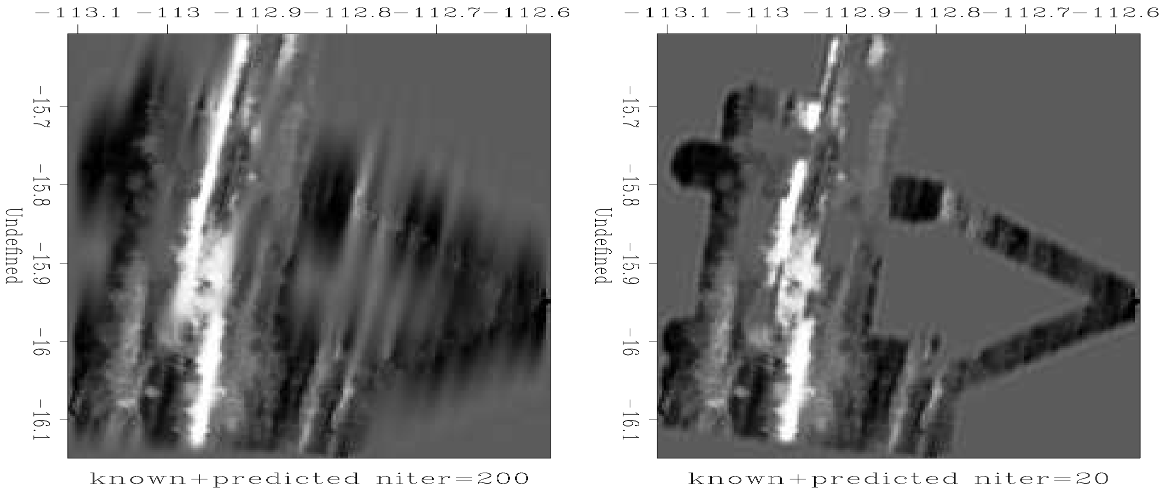

Figure 1.21: SeaBeam interpolation with the prediction-error filter. The left plot was taken after 200 conjugate-gradient iterations; the right after 20 iterations. optim-seabold [ER,M]

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 1.22) is similar to the result of the corresponding unpreconditioned interpolation. However, the output after just 20 iterations (the right plot in Figure 1.22) is already fairly close to the solution.

B-SPLINE REGULARIZATION

As demonstrated in Chapter **??**, B-splines provide an exceptionally accurate method of forward interpolation. In this section, I discuss how this choice of the forward operator affects the regularization part of the problem. In the case of B-spline interpolation, the forward operator $\mathbf{L}$ is a cascade of two operators: recursive deconvolution $\mathbf{B}^{-1}$, which converts the model

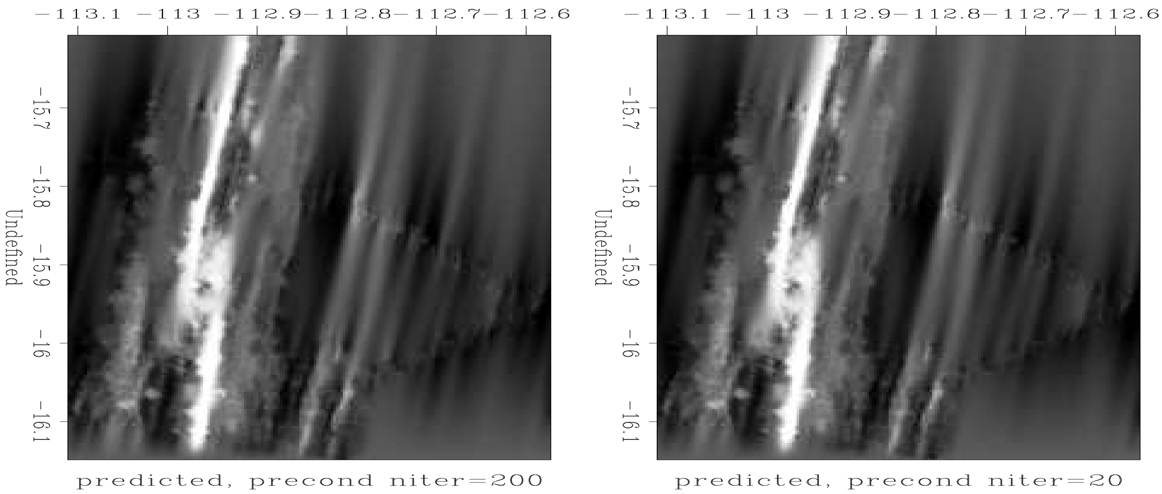

Figure 1.22: SeaBeam interpolation with the inverse prediction-error filter. The left plot was taken after 200 conjugate-gradient iterations; the right, after 20 iterations. optim-seabnew [ER,M]

vector $\mathbf{m}$ to the vector of spline coefficients $\mathbf{c}$, and a spline basis construction operator $\mathbf{F}$. System (**??**-**??**) transforms to

$$\begin{aligned} \mathbf{F}\mathbf{B}^{-1}\mathbf{m} &\approx \mathbf{d}\,; &(1.10)\\ \epsilon\mathbf{D}\mathbf{m} &\approx \mathbf{0}\,. &(1.11)\end{aligned}$$

We can rewrite (1.10-1.11) in the form that involves only spline coefficients:

$$\begin{aligned} \mathbf{F}\mathbf{c} &\approx \mathbf{d}\,; &(1.12)\\ \epsilon\mathbf{D}\mathbf{B}\mathbf{c} &\approx \mathbf{0}\,. &(1.13)\end{aligned}$$

After we find a solution of system (1.12-1.13), the model $\mathbf{m}$ will be reconstructed by the simple convolution

$$\mathbf{m} = \mathbf{B}\mathbf{c}\,. \tag{1.14}$$

This approach is clearly just another version of model preconditioning.

The inconvenient part of system (1.12-1.13) is the complex regularization operator **DB**. Is it possible to avoid the cascade of **B** and **D** and to construct a regularization operator directly applicable to the spline coefficients **c**? The answer is positive. In the following subsection, I develop a method for constructing spline regularization operators from differential equations.

Spline regularization

In many cases, the regularization condition originates in a continuous differential operator. I provide several examples of such differential operators in Chapters 5 and 6.

Let us denote the continuous regularization operator by D. Regularization implies seeking a function $f(x)$ such that the least-squares norm of $D\left[f(x)\right]$ is minimum. Using the usual expression for the least-squares norm of continuous functions and substituting the basis decomposition (**??**), we obtain the expression

$$\left\|D\left[f(x)\right]\right\| = \int \left(D\left[f(x)\right]\right)^2 dx = \int \left(\sum_{k\in K} c_k D\left[\beta(x-k)\right]\right)^2 dx \;. \tag{1.15}$$

The problem of finding function $f(x)$ reduces to the problem of finding the corresponding set of basis coefficients c_k. We can obtain the solution to the least-squares optimization by differentiating the quadratic objective function (1.15) with respect to the basis coefficients c_k. This leads to the system of linear equations

$$\sum_{k\in K} c_k \int D\left[\beta(x-k)\right] D\left[\beta(x-j)\right] dx = \sum_{k\in K} c_k d_{j-k} = 0 \;, \tag{1.16}$$

where

$$d_j = \int D\left[\beta(x)\right] D\left[\beta(x-j)\right] dx \;. \tag{1.17}$$

Equation (1.16) is clearly a discrete convolution of the spline coefficients c_k with the filter d_j defined in equation (1.17). To transform the system (1.16) to a regularization condition of the form

$$\mathbf{D}_c \mathbf{c} \approx \mathbf{0} \,, \tag{1.18}$$

we need to treat the digital filter d_j as an autocorrelation and find its minimum-phase factor by spectral factorization. The Wilson-Burg algorithm, described earlier, is an appropriate tool for the task. Equation (1.18) replaces equation (1.13) in the inverse interpolation problem setting.

We have, thus, found a constructive way of creating B-spline regularization operators from continuous differential equations.

Test example

For a simple 1-D test of B-spline regularization, I chose the function shown in Figure **??**, but sampled at irregular locations. To create two different regimes for the inverse interpolation problem, I chose 50 and 500 random locations. I interpolated these two sets of points to 500 and 50 regular samples, respectively. The first test corresponds to an under-determined situation, while the second test is clearly over-determined. Figures 1.23 and 1.24 show the input data for the two test after normalized binning to the selected regular bins.

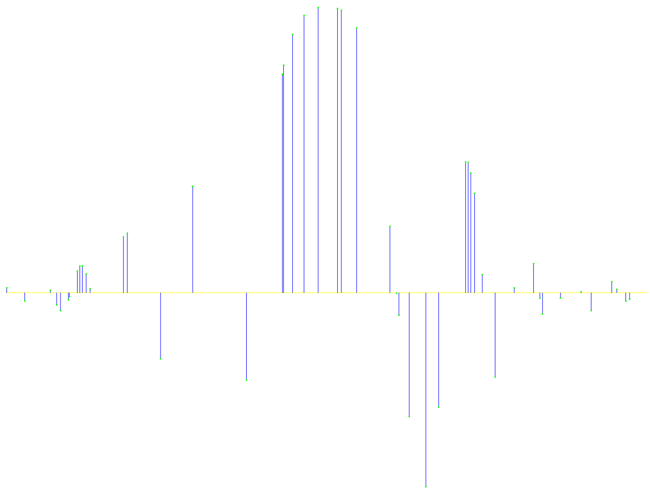

Figure 1.23: 50 random points binned to 500 regular grid points. The random data are used for testing inverse interpolation in an under-determined situation. optim-bin500 [ER]

I solved system (1.12)-(1.18) by the iterative conjugate-gradient method, utilizing a recursive filter preconditioning for faster convergence. To construct the regularization operator **D**,

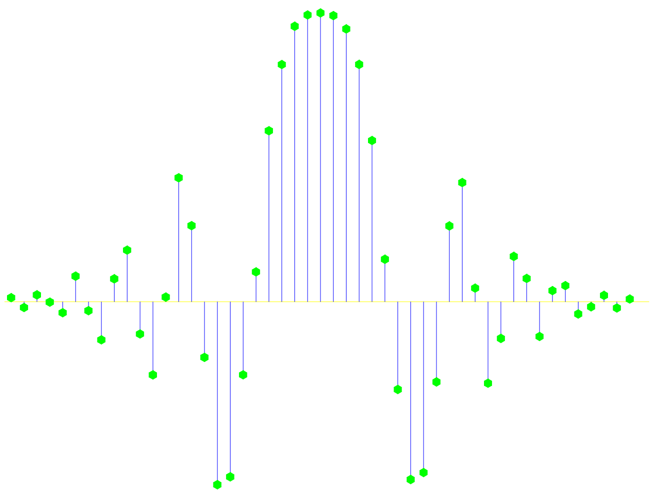

Figure 1.24: 500 random points binned to 50 regular grid points. The random data are used for testing inverse interpolation in an over-determined situation. optim-bin50 [ER]

I used the method of the previous subsection with the tension-spline differential equation that I will describe in Chapter 5.

The least-squares differences between the true and the estimated model are plotted in Figures 1.25 and 1.26. Observing the behavior of the model misfit versus the number of iterations and comparing simple linear interpolation with the third-order B-spline interpolation, we discover that

- In the under-determined case, both methods converge to the same final estimate, but B-spline inverse interpolation does it faster (with fewer iterations). However, the total computational gain is not significant because each B-spline iteration is more expensive than the corresponding linear interpolation iteration.

- In the over-determined case, both methods converge similarly at early iterations, but B-spline inverse interpolation results in a more accurate final estimate.

From the results of this simple experiment, it is apparent that the main advantage of using more accurate interpolation in the data regularization context occurs in the over-determined situation, when the estimated model is well constrained by the available data.

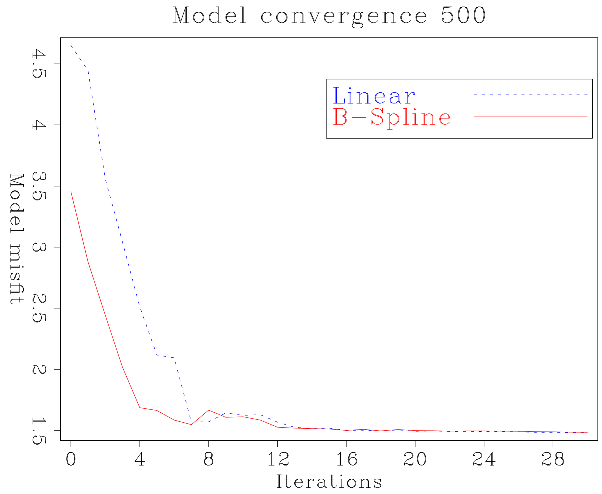

Figure 1.25: Model convergence in the under-determined case. Dashed line: using linear interpolation. Solid line: using third-order B-spline. optim-norm500 [ER]

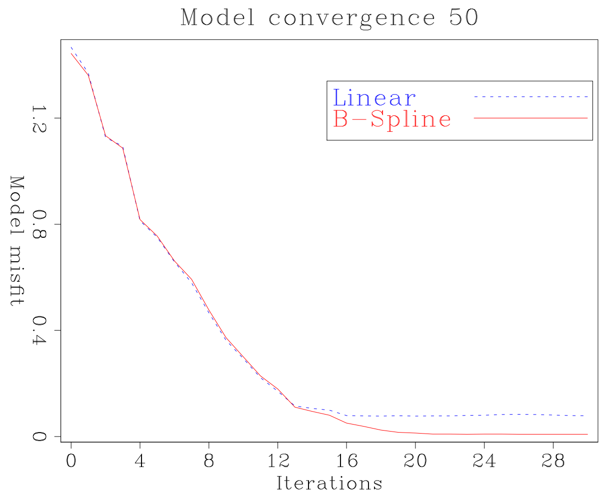

Figure 1.26: Model convergence in the over-determined case. Dashed line: using linear interpolation. Solid line: using third-order B-spline. optim-norm50 [ER]

Conclusions

We observe a significant (order-of-magnitude) speed-up in the optimization convergence when preconditionining interpolation problems with inverse recursive filtering. Since inverse filtering takes almost the same time as forward convolution, this speed-up translates straightforwardly into computational time savings.

The savings are hardly noticeable for simple test problems, but they can have a direct impact on the mere feasibility of iterative least-square inversion for large-scale (seismic-exploration-size) problems.

In the multidimensional case, recursive filtering is enabled by Claerbout's helix transform. The Wilson-Burg spectral factorization method allows us to construct stable recursive filters. By analyzing the role of B-spline interpolation in data regularization, I have introduced a method of constructing B-spline discrete regularization operators from continuous differential equations.

In the next chapter, I discuss possible choices of the regularization operator $\mathbf{D}$ and the preconditioning operator $\mathbf{P}$ in data regularization problems.

ACKNOWLEDGMENTS

I collaborated with Bob Clapp and Jon Claerbout on developing recursive preconditioning for missing data interpolation problems (Fomel et al., 1997; Clapp et al., 1997) and with Paul Sava, James Rickett, and Jon Claerbout on developing the helical version of the Wilson-Burg method (Sava et al., 1998; Sava and Fomel, 1999). For the latter, we owe a great deal to John Burg. I would like to thank him and Francis Muir for many useful and stimulating discussions.

Bibliography

Babich, V. M., 1991, Short-wavelength diffraction theory: asymptotic methods: Springer-Verlag, Berlin; New York.

Bagaini, C., and Spagnolini, U., 1993, Common shot velocity analysis by shot continuation operator: 63rd Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 673–676.

Bagaini, C., and Spagnolini, U., 1996, 2-D continuation operators and their applications: Geophysics, **61**, no. 06, 1846–1858.

Bagaini, C., Spagnolini, U., and Pazienza, V. P., 1994, Velocity analysis and missing offset restoration by prestack continuation operators: 64th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1549–1552.

Bale, R., and Jakubowicz, H., 1987, Post-stack prestack migration: 57th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Session:S14.1.

Ben-Avraham, Z., Amit, G., Golan, A., and Begin, Z. B., 1990, The bathymetry of Lake Kinneret and its structural significance: Israel Journal of Earth Sciences, **39**, 77–84.

Beylkin, G., 1985, Imaging of discontinuities in the inverse scattering problem by inversion of a causal generalized Radon transform: Journal of Mathematical Physics, **26**, 99–108.

Biondi, B., and Chemingui, N., 1994a, Transformation of 3-D prestack data by Azimuth Moveout: SEP–**80**, 125–143.

Biondi, B., and Chemingui, N., 1994b, Transformation of 3-D prestack data by azimuth moveout (AMO): 64th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1541–1544.

Biondi, B., and Palacharla, G., 1996, 3-D prestack migration of common-azimuth data: Geophysics, **61**, no. 6, 1822–1832.

Biondi, B., Fomel, S., and Chemingui, N., 1998, Azimuth moveout for 3-D prestack imaging: Geophysics, **63**, no. 02, 574–588.

Biondi, B., 1996, Common-azimuth prestack depth migration of a North Sea data set: SEP–**93**, 1–14.

Black, J. L., Schleicher, K. L., and Zhang, L., 1993, True-amplitude imaging and dip moveout: Geophysics, **58**, no. 1, 47–66.

Bleistein, N., and Cohen, J. K., 1995, The effect of curvature on true amplitude DMO: Proof of concept: ACTI, 4731U0015-2F; CWP-193, Colorado School of Mines.

Bleistein, N., 1984, Mathematical methods for wave phenomena: Academic Press Inc. (Harcourt Brace Jovanovich Publishers), New York.

Bleistein, N., 1990, Born DMO revisited: 60th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1366–1369.

Bolondi, G., Loinger, E., and Rocca, F., 1982, Offset continuation of seismic sections: Geophys. Prosp., **30**, no. 6, 813–828.

Briggs, I. C., 1974, Machine contouring using minimum curvature: Geophysics, **39**, no. 1, 39–48.

Brown, M., Clapp, R. G., and Marfurt, K., 1999, Predictive signal/noise separation of groundroll-contaminated data: SEP–**102**, 111–128.

Burg, J. P., 1972, The relationship between maximum entropy spectra and maximum likelihood spectra (short note): Geophysics, **37**, no. 2, 375–376.

Burg, J. P., 1975, Maximum entropy spectral analysis: Ph.D. thesis, Stanford University.

Canales, L. L., 1984, Random noise reduction: 54th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Session:S10.1.

Canning, A., and Gardner, G. H. F., 1996, Regularizing 3-D data sets with DMO: Geophysics, **61**, no. 04, 1103–1114.

Chemingui, N., and Biondi, B., 1994, Coherent partial stacking by offset continuation of 2-D prestack data: SEP–**82**, 117–126.

Chemingui, N., and Biondi, B., 1996, Handling the irregular geometry in wide azimuth surveys: 66th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 32–35.

Chemingui, N., 1999, Imaging irregularly sampled 3D prestacked data: Ph.D. thesis, Stanford University.

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

Claerbout, J. F., 1985, Imaging the Earth's Interior: Blackwell Scientific Publications.

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

Claerbout, J. F., 1993, 3-D local monoplane annihilator: SEP–**77**, 19–25.

Claerbout, J., 1997, Multidimensional recursive filters via a helix: SEP–**95**, 1–13.

Claerbout, J., 1998a, Multidimensional recursive filters via a helix: Geophysics, **63**, no. 5, 1532–1541.

Claerbout, J., 1998b, Multidimensional recursive filters via a helix: SEP–**97**, 319–335.

Claerbout, J. F., 1998c, Multidimensional recursive filters via a helix with application to velocity estimation and 3-D migration: 68th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1995–1998.

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. L., 1998, Regularizing time tomography with steering filters: SEP–**97**, 137–146.

Clapp, R. G., and Biondi, B. L., 2000, Tau tomography with steering filters: 2-D field data example: SEP–**103**, 1–19.

Clapp, R. G., and Brown, M., 2000, $(t-x)$ domain, pattern-based multiple separation: SEP–**103**, 201–210.

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

Clapp, R., Biondi, B., Fomel, S., and Claerbout, J., 1998, Regularizing velocity estimation using geologic dip information: 68th Ann. Internat. Meeting, Soc. Expl. Geophys., Expanded Abstracts, 1851–1854.

Clapp, R. G., Fomel, S., Crawley, S., and Claerbout, J. F., 1999, Directional smoothing of non-stationary filters: SEP–**100**, 197–209.

Clapp, R., 2000a, 3-D steering filters: SEP–**105**, 109–116.

Clapp, R., 2000b, Multiple realizations using standard inversion techniques: SEP–**105**, 67–78.

Cole, S. P., 1995, Passive seismic and drill-bit experiments using 2-D arrays: Ph.D. thesis, Stanford University.

Courant, R., 1962, Methods of mathematical physics: Interscience Publishers, New York.

Crawley, S., Clapp, R., and Claerbout, J., 1998, Decon and interpolation with nonstationary filters: SEP–**97**, 183–192.

Crawley, S., Clapp, R., and Claerbout, J., 1999, Interpolation with smoothly nonstationary prediction-error filters: 69th Ann. Internat. Meeting, Soc. Expl. Geophys., Expanded Abstracts, 1154–1157.

Crawley, S., 1995a, An example of inverse interpolation accelerated by preconditioning: SEP–**84**, 289–294.

Crawley, S., 1995b, Multigrid nonlinear SeaBeam interpolation: SEP–**84**, 279–288.

Crawley, S., 1999, Interpolation with smoothly nonstationary prediction-error filters: SEP–**100**, 181–196.

Crawley, S., 2000, Seismic trace interpolation with nonstationary prediction-error filters: Ph.D. thesis, Stanford University.

de Boor, C., 1978, A practical guide to splines: Springer-Verlag.

Deregowski, S. M., and Rocca, F., 1981, Geometrical optics and wave theory of constant offset sections in layered media: Geophys. Prosp., **29**, no. 3, 374–406.

Deregowski, S. M., 1986, What is DMO: First Break, **4**, no. 7, 7–24.

Dubois, G., 1999, Spatial interpolation comparison 97: Foreword and introduction: Journal of Geographic Information and Decision Analysis, **2**, 1–10.

Fomel, S., and Biondi, B., 1995, The time and space formulation of azimuth moveout: 65th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1449–1452.

Fomel, S., and Bleistein, N., 1996, Amplitude preservation for offset continuation: Confirmation for Kirchhoff data: SEP–**92**, 219–227.

Fomel, S., and Claerbout, J., 1995, Searching the Sea of Galilee: The splendors and miseries of iteratively reweighted least squares: SEP–**84**, 259–270.

Fomel, S., Bleistein, N., Jaramillo, H., and Cohen, J. K., 1996, True amplitude DMO, offset continuation and AVA/AVO for curved reflectors: 66th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1731–1734.

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

Fomel, S. B., 1994, Kinematically equivalent differential operator for offset continuation of seismic sections: Russian Geology and Geophysics, **35**, no. 9, 122–134.

Fomel, S., 1995a, Amplitude preserving offset continuation in theory Part 1: The offset continuation equation: SEP–**84**, 179–198.

Fomel, S., 1995b, Amplitude preserving offset continuation in theory Part 2: Solving the equation: SEP–**89**, 109–132.

Fomel, S., 1996a, Least-square inversion with inexact adjoints. Method of conjugate directions: A tutorial: SEP–**92**, 253–265.

Fomel, S., 1996b, Stacking operators: Adjoint versus asymptotic inverse: SEP–**92**, 267–292.

Fomel, S., 2000a, Applications of plane-wave destructor filters: SEP–**105**, 1–26.

Fomel, S., 2000b, Seismic data interpolation with the offset continuation equation: SEP–**103**, 237–254.

Fung, Y. C., 1965, Foundations of solid mechanics: Prentice-Hall.

Gazdag, J., 1978, Wave equation migration with the phase shift method: Geophysics, **43**, 1342–1351.

Goldin, S. V., and Fomel, S. B., 1995, Estimation of reflection coefficient in DMO: Russian Geology and Geophysics, **36**, no. 4, 103–115.

Goldin, S. V., 1988, Transformation and recovery of discontinuities in problems of tomographic type: Institute of Geology and Geophysics, Novosibirsk (in Russian).

Goldin, S., 1990, A geometric approach to seismic processing: the method of discontinuities: SEP–**67**, 171–210.

Goldin, S. V., 1994, Superposition and continuation of tranformations used in seismic migration: Russian Geology and Geophysics, **35**, no. 9, 131–145.

Golub, G. H., and Van Loan, C. F., 1996, Matrix computations: The John Hopkins University Press.

Gradshtein, I. S., and Ryzhik, I. M., 1994, Table of integrals, series, and products: Boston: Academic Press.

Haddon, R. A. W., and Buchen, P. W., 1981, Use of Kirchhoff's formula for body wave calculations in the earth: Geophys. J. Roy. Astr. Soc., **67**, 587–598.

Hale, I. D., 1983, Dip moveout by Fourier transform: Ph.D. thesis, Stanford University.

Hale, D., 1984, Dip-moveout by Fourier transform: Geophysics, **49**, no. 6, 741–757.

Hale, D., 1991, Course notes: Dip moveout processing: Soc. Expl. Geophys.

Hale, D., Ed. **DMO processing**. Society Of Exploration Geophysicists, 1995.

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

Kolmogoroff, A. N., 1939, Sur l'interpolation et extrapolation des suites stationnaires: C.R. Acad.Sci., **208**, 2043–2045.

Liner, C. L., and Cohen, J. K., 1988, An amplitude-preserving inverse of Hale's DMO: 58th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1117–1120.

Liner, C., 1990, General theory and comparative anatomy of dip moveout: Geophysics, **55**, no. 5, 595–607.

Liner, C. L., 1991, Born theory of wave-equation dip moveout: Geophysics, **56**, no. 2, 182–189.

Mazzucchelli, P., and Rocca, F., 1999, Regularizing land acquisitions using shot continuation operators: effects on amplitudes: 69th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1995–1998.

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

Nolet, G., Ed., 1987, Seismic tomography: with applications in global seismology and exploration geophysics D. Reidel.

Notfors, C. D., and Godfrey, R. J., 1987, Dip moveout in the frequency-wavenumber domain (short note): Geophysics, **52**, no. 12, 1718–1721.

Petkovsek, M., Wilf, H. S., and Zeilberger, D., 1996, $A = B$: A K Peters Ltd., Wellesley, MA.

Rickett, J., and Claerbout, J., 1998, Helical factorization of the Helmholtz equation: SEP–**97**, 353–362.

Rickett, J., and Claerbout, J., 1999a, Acoustic daylight imaging via spectral factorization: Helioseismology and reservoir monitoring: SEP–**100**, 171–180.

Rickett, J., and Claerbout, J., 1999b, Acoustic daylight imaging via spectral factorization: Helioseismology and reservoir monitoring: 69th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1675–1678.

Ronen, S., Sorin, V., and Bale, R., 1991, Spatial dealiasing of 3-D seismic reflection data: Geophysical Journal International, pages 503–511.

Ronen, J., 1987, Wave equation trace interpolation: Geophysics, **52**, no. 7, 973–984.

Ronen, S., 1994, Handling irregular geometry: Equalized DMO and beyond: 64th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1545–1548.

Salvador, L., and Savelli, S., 1982, Offset continuation for seismic stacking: Geophys. Prosp., **30**, no. 6, 829–849.

Sandwell, D. T., 1987, Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data: Geophys. Res. Letters, **14**, no. 2, 139–142.

Santos, L. T., Schleicher, J., and Tygel, M., 1997, 2.5-D true-amplitude offset continuation: Journal of Seismic Exploration, **6**, no. 2-3, 103–116.

Sava, P., and Fomel, S., 1999, Spectral factorization revisited: SEP–**100**, 227–234.

Sava, P., Rickett, J., Fomel, S., and Claerbout, J., 1998, Wilson-Burg spectral factorization with application to helix filtering: SEP–**97**, 343–351.

Schwab, M., and Claerbout, J., 1995, The interpolation of a 3-D data set by a pair of 2-D filters: SEP–**84**, 271–278.

Schwab, M., 1993, Shot gather continuation: SEP–**77**, 117–130.

Schwab, M., 1998, Enhancement of discontinuities in seismic 3-D images using a Java estimation library: Ph.D. thesis, Stanford University.

Schweikert, D. G., 1966, An interpolation curve using a spline in tension: Journal of Mathematics and Physics, **45**, 312–313.

Smith, W. H. F., and Wessel, P., 1990, Gridding with continuous curvature splines in tension: Geophysics, **55**, no. 3, 293–305.

Spagnolini, U., and Opreni, S., 1996, 3-D shot continuation operator: 66th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 439–442.

Spitz, S., 1991, Seismic trace interpolation in the f-x domain: Geophysics, **56**, no. 6, 785–794.

Stolt, R. H., 1978, Migration by Fourier transform: Geophysics, **43**, no. 1, 23–48.

Stovas, A. M., and Fomel, S. B., 1993, Kinematically equivalent DMO operators: Presented at the SEG-Moscow, SEG-Moscow.

Stovas, A. M., and Fomel, S. B., 1996, Kinematically equivalent integral DMO operators: Russian Geology and Geophysics, **37**, no. 2, 102–113.

Swain, C. J., 1976, A FORTRAN IV program for interpolating irregularly spaced data using the difference equations for minimum curvature: Computers and Geosciences, **1**, 231–240.

Symes, W. W., and Carazzone, J. J., 1991, Velocity inversion by differential semblance optimization: Geophysics, **56**, no. 5, 654–663.

Symes, W. W., 1999, All stationary points of differential semblance are asymptotic global minimizers: Layered acoustics: SEP–**100**, 71–92.

Tenenbaum, M., and Pollard, H., 1985, Ordinary differential equations : an elementary textbook for students of mathematics, engineering, and the sciences: Dover Publications.

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

Timoshenko, S., and Woinowsky-Krieger, S., 1968, Theory of plates and shells: McGraw-Hill.

Červený, V., Molotkov, I. A., and Pšenčik, I., 1977, Ray method in seismology: Univerzita Karlova, Praha.

Watson, G. N., 1952, A treatise on the theory of Bessel functions: Cambridge University Press, 2nd edition.

Wilson, G., 1969, Factorization of the covariance generating function of a pure moving average process: SIAM J. Numer. Anal., **6**, no. 1, 1–7.

Woodward, M. J., Farmer, P., Nichols, D., and Charles, S., 1998, Automated 3-D tomographic velocity analysis of residual moveout in prestack depth migrated common image point gathers: 68th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1218–1221.

Zhao, Y., 1999, Helix derivative and low-cut filters' spectral feature and application: SEP–**100**, 235–250.

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. 03, 815–820.