Geophysical Prospecting, 2003, **51**, 409–420

The Wilson–Burg method of spectral factorization with application to helical filtering

Sergey Fomel,[1]* Paul Sava,[2] James Rickett[3] and Jon F. Claerbout[2]

[1]*Bureau of Economic Geology, The University of Texas at Austin, University Station, Box X, Austin, TX 78713-8972,* [2]*Stanford Exploration Project, Department of Geophysics, Stanford University, Stanford, CA 94305, and* [3]*ChevronTexaco Exploration and Production Technology Company, 6001 Bollinger Canyon Road, San Ramon, CA 94583-2324, USA*

Received April 2001, revision accepted May 2003

ABSTRACT

Spectral factorization is a computational procedure for constructing minimum-phase (stable inverse) filters required for recursive inverse filtering. We present a novel method of spectral factorization. The method iteratively constructs an approximation of the minimum-phase filter with the given autocorrelation by repeated forward and inverse filtering and rearranging of the terms. This procedure is especially efficient in the multidimensional case, where the inverse recursive filtering is enabled by the helix transform.

To exemplify a practical application of the proposed method, we consider the problem of smooth two-dimensional data regularization. Splines in tension are smooth interpolation surfaces whose behaviour in unconstrained regions is controlled by the tension parameter. We show that such surfaces can be efficiently constructed with recursive filter preconditioning and we introduce a family of corresponding two-dimensional minimum-phase filters. The filters are created by spectral factorization on a helix.

INTRODUCTION

Spectral factorization is the task of estimating a minimum-phase signal from a given power spectrum. The advent of the helical coordinate system (Mersereau and Dudgeon 1974; Claerbout 1998) has led to renewed interest in spectral factorization algorithms, since they now apply to multidimensional problems. Specifically, spectral factorization algorithms provide the key to rapid multidimensional recursive filtering with arbitrary functions, which in turn has geophysical applications in preconditioning inverse problems (Clapp *et al.* 1998; Fomel and Claerbout 2003), wavefield extrapolation (Rickett, Claerbout and Fomel 1998; Rickett 2000; Zhang , Zhang and Zhou 2000; Zhang and Shan 2001) and three-dimensional noise attenuation (Ozdemir *et al.* 1999a,b; Rickett, Guitton and Gratwick 2001).

*E-mail: sergey.fomel@beg.utexas.edu

The Kolmogoroff (cepstral or Hilbert transform) method of spectral factorization (Kolmogoroff 1939; Claerbout 1976; Oppenheim and Shafer 1989) is often used by the geophysical community because of its computational efficiency. However, as a frequency-domain method, it has certain limitations. For example, the assumption of periodic boundary conditions often requires extreme amounts of zero-padding for a stable factorization. This is one of the limitations which make this method inconvenient for multidimensional applications.

The Wilson–Burg method, introduced in this paper, is an iterative algorithm for spectral factorization based on Newton's iterations. The algorithm exhibits quadratic convergence. It provides a time-domain approach that is potentially more efficient than the Kolmogoroff method. We include a detailed comparison of the two methods.

Recent surveys (Goodman *et al.* 1997; Sayed and Kailith 2001) have discussed some other methods for spectral factorization, such as the Schur method (Schur 1917), the Bauer method (Bauer 1955) and Wilson's original method (Wilson 1969). The latter is noted for its superb numerical

properties. We introduce Burg's modification to this algorithm, which raises the computational attractiveness of this method to a new level. The Wilson–Burg method avoids the need for matrix inversion, essential for the original Wilson's algorithm, and it reduces the computational effort from $O(N^3)$ operations to $O(N^2)$ operations per iteration. An alternative way of accelerating Wilson's iteration was suggested by Laurie (1980). We have found the Wilson–Burg algorithm to be especially suitable for applications of multidimensional helical filtering, where the number of filter coefficients can be small, and the cost effectively reduces to $O(N)$ operations.

The second part of the paper contains a practical example of the introduced spectral factorization method. The method is applied to the problem of two-dimensional smooth data regularization. This problem often occurs in mapping potential fields data and in other geophysical problems. Applying the Wilson–Burg spectral factorization method, we construct a family of 2D recursive filters, which correspond to different values of tension in the tension-spline approach to data regularization (Smith and Wessel 1990). We then use the constructed filters for an efficient preconditioning of the data regularization problem. The combination of an efficient spectral factorization and an efficient preconditioning technique provides an attractive practical method for multidimensional data interpolation. The technique is illustrated with bathymetry data from the Sea of Galilee (Lake Kinneret) in Israel.

DESCRIPTION OF THE METHOD

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,

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

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

$$\begin{aligned} S(Z) &= A_t(Z)\bar{A}_t(1/Z) + A_t(Z)[\bar{A}_{t+1}(1/Z) - \bar{A}_t(1/Z)] \\ &\quad + \bar{A}_t(1/Z)[A_{t+1}(Z) - A_t(Z)] \\ &= A_t(Z)\bar{A}_{t+1}(1/Z) + \bar{A}_t(1/Z)A_{t+1}(Z) - A_t(Z)\bar{A}_t(1/Z), \end{aligned} \tag{2}$$

where $A_t(Z)$ denotes the signal estimate at iteration t. Starting from some initial estimate $A_0(Z)$, such as $A_0(Z) = 1$, the linear system (2) is solved iteratively for the updated signal $A_{t+1}(Z)$. Wilson (1969) presented a rigorous proof that iteration (2) operates with minimum phase signals provided that the initial estimate $A_0(Z)$ is minimum phase. The original algorithm estimates the new approximation $A_{t+1}(Z)$ by matrix inversion implied in the solution of the system.

Burg (1998, pers. comm.) recognized that dividing both sides of (2) 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:

$$1 + \frac{S(Z)}{\bar{A}_t(1/Z)A_t(Z)} = \frac{A_{t+1}(Z)}{A_t(Z)} + \frac{\bar{A}_{t+1}(1/Z)}{\bar{A}_t(1/Z)}. \tag{3}$$

Equation (3) 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 (3) 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 $A_{t+1}(Z)/A_t(Z)$.

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 on a simple 1D signal. The autocorrelation $S(Z)$ in this case is $1334 + 867(Z + 1/Z) + 242(Z^2 + 1/Z^2) + 24(Z^3 + 1/Z^3)$ 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).

The clear advantage of the Wilson–Burg algorithm in comparison with the original Wilson algorithm is in the elimination of the expensive matrix inversion step. Only convolution and deconvolution operations are used at each iteration step.

Table 1 Example of convergence of the Wilson–Burg iteration

Iteration	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

Comparison of Wilson–Burg and Kolmogoroff methods

The Kolmogoroff (cepstral or Hilbert transform) spectral factorization algorithm (Kolmogoroff 1939; Claerbout 1976; Oppenheim and Shafer 1989) is widely used because of its computational efficiency. While this method is easily extended to the multidimensional case with the help of a helical transform (Rickett and Claerbout 1999), there are several circumstances that make the Wilson–Burg method more attractive in multidimensional filtering applications.

- The Kolmogoroff method takes O(N log N) operations, where N is the length of the autocorrelation function. The cost of the Wilson–Burg method is proportional to [number of iterations] × [filter length] × N. If we keep the filter small and limit the number of iterations, the Wilson–Burg method can be cheaper (linear in N). In comparison, the cost of the original Wilson's method is [number of iterations] × O(N^3).
- The Kolmogoroff method works in the frequency domain and assumes periodic boundary conditions. Autocorrelation 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. The Wilson–Burg method is implemented in the time domain, where no extra 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 helical filtering, involves the dangerous step of truncating the filter coefficients to reduce the size of the filter. If the autocorrelation function has roots close to the unit circle, truncating filter coefficients may easily lead to non-minimum-phase filters. The Wilson–Burg method allows us to 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 than a

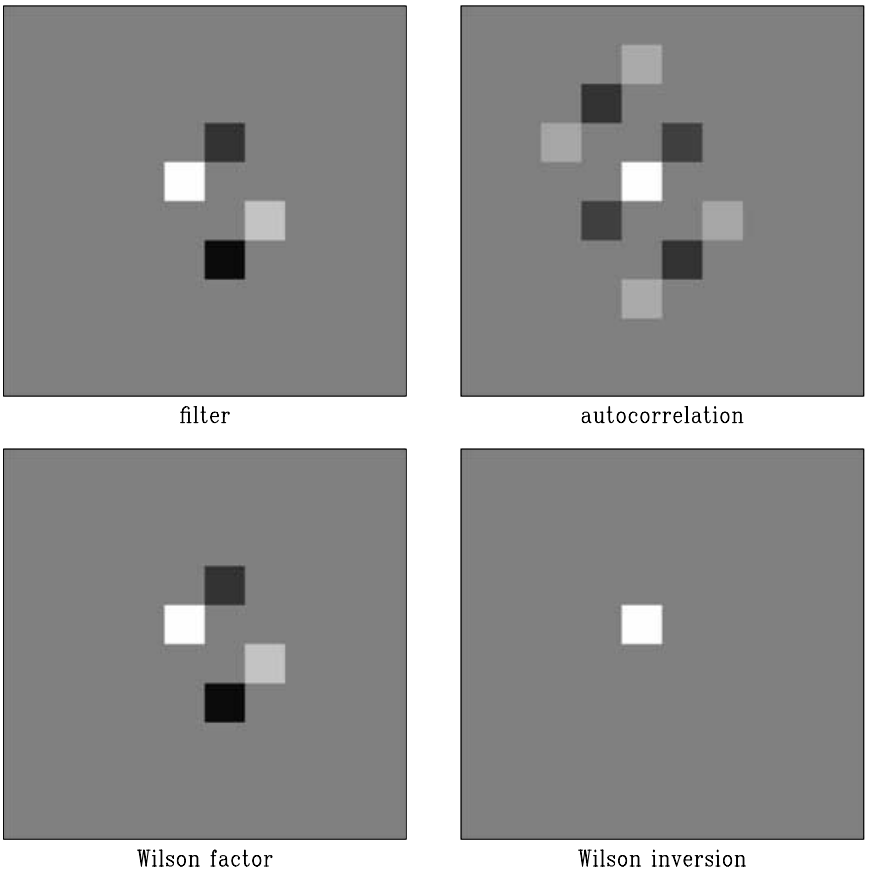

Figure 1 Example of 2D Wilson–Burg factorization. Top left: the input filter; top right: its autocorrelation; bottom left: the factor obtained by the Wilson–Burg method; bottom right: the result of deconvolution.

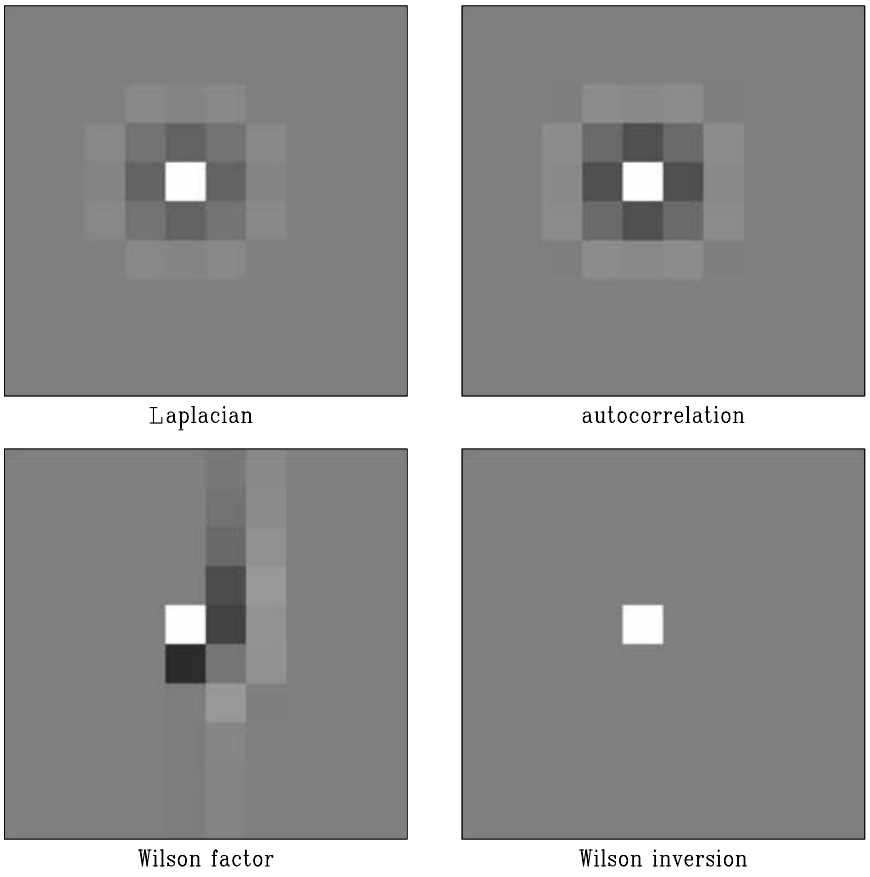

Figure 2 Creating a minimum-phase Laplacian filter. Top left: Laplacian filter; top right: its autocorrelation (biharmonic filter); bottom left: factor obtained by the Wilson–Burg method (minimum-phase Laplacian); bottom right: the result of deconvolution.

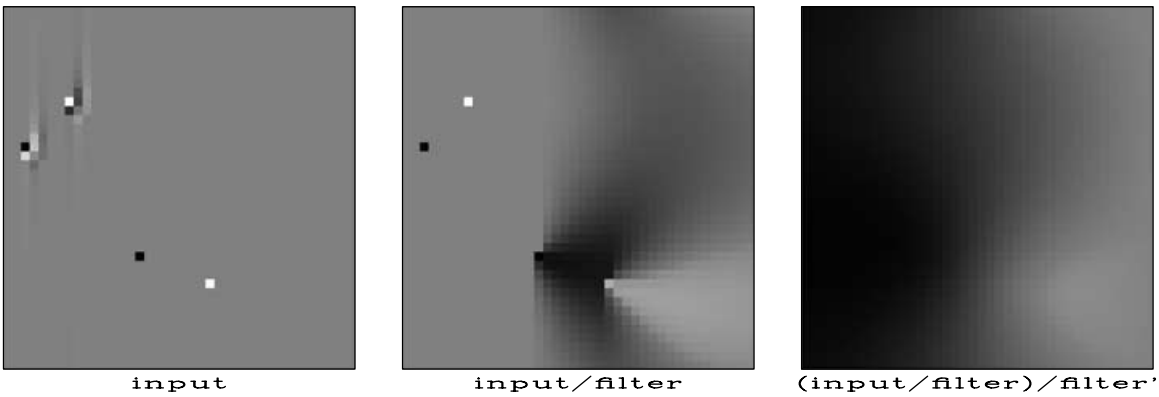

Figure 3 2D deconvolution with the minimum-phase Laplacian. Left: input; centre: output of deconvolution; right: output of deconvolution and adjoint deconvolution (equivalent to solving the biharmonic differential equation).

certain threshold, and repeat the factorizing process again with the shorter filter.

Factorization examples

The first simple example of helical spectral factorization is shown in Fig. 1. 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 rightmost plot in Fig. 1.)

A practical example is depicted in Fig. 2. The symmetric Laplacian operator is often used in practice for regularizing smooth data. In order to construct a corresponding recursive

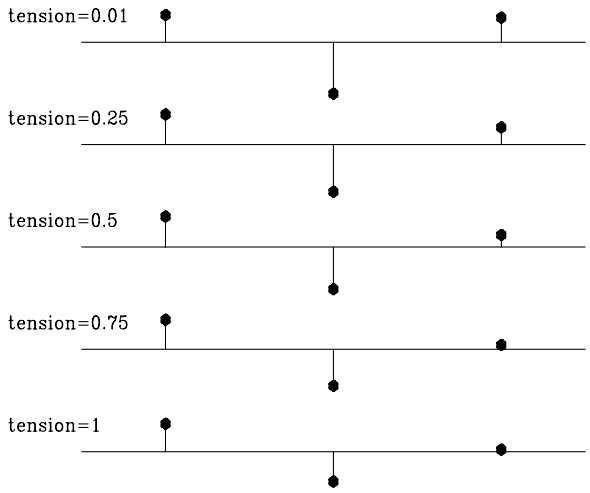

Figure 4 1D minimum-phase filters for different values of the tension parameter λ. The filters range from the second derivative for $\lambda = 0$ to the first derivative for $\lambda = 1$.

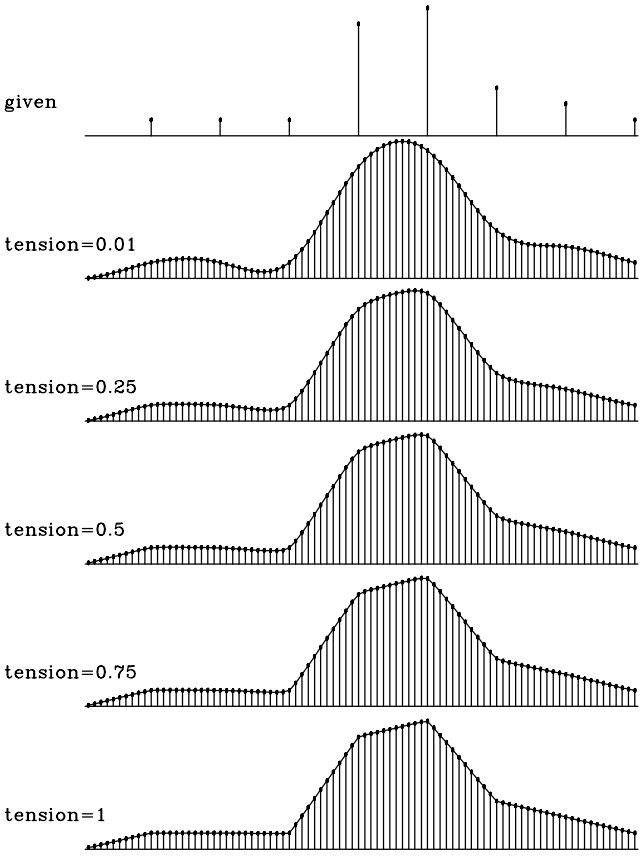

Figure 5 Interpolating a simple 1D synthetic with recursive filter preconditioning for different values of the tension parameter λ. The input data are shown at the top. The interpolation results range from a natural cubic spline interpolation for $\lambda = 0$ to linear interpolation for $\lambda = 1$.

preconditioner, we factorize the Laplacian autocorrelation (the biharmonic operator) using the Wilson–Burg algorithm. Figure 2 shows the resultant filter. The minimum-phase Laplacian filter has several times more coefficients than 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). The gain in convergence from recursive filter preconditioning outweighs the loss of efficiency from the longer filter. Figure 3 shows a construction of the smooth inverse impulse response by application of the $\mathbf{C} = \mathbf{P}\,\mathbf{P}^{\mathrm{T}}$ 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 the next section.

APPLICATION OF SPECTRAL FACTORIZATION: REGULARIZING SMOOTH DATA WITH SPLINES IN TENSION

The method of minimum curvature is an old and ever-popular approach for constructing smooth surfaces from irregularly spaced data (Briggs 1974). The surface of minimum curvature corresponds to the minimum of the Laplacian power or, in an alternative formulation, satisfies the biharmonic differential equation. Physically, it models the behaviour of an elastic plate. In the one-dimensional case, the minimum-curvature method leads to the natural cubic spline interpolation (de Boor 1978). In the 2D case, a surface can be interpolated with biharmonic splines (Sandwell 1987) or gridded with an iterative finite-difference scheme (Swain 1976). We approach the gridding (data regularization) problem with an iterative least-squares optimization scheme.

In most practical cases, the minimum-curvature method produces a visually pleasing smooth surface. However, in cases of large changes in the surface gradient, the method can create strong artificial oscillations in the unconstrained regions. Switching to lower-order methods, such as minimizing the power of the gradient, solves the problem of extraneous inflections, but also removes the smoothness constraint and leads to gradient discontinuities (Fomel and Claerbout 1995). A remedy, suggested by Schweikert (1966), is known as *splines in tension*. Splines in tension are constructed by minimizing a modified quadratic form that includes a tension term. Physically, the additional term corresponds to tension in elastic plates (Timoshenko and Woinowsky-Krieger 1968). Smith and Wessel (1990) developed a practical algorithm of

2D gridding with splines in tension and implemented it in the popular GMT software package.

In this section, we develop an application of helical preconditioning to gridding with splines in tension. We accelerate an iterative data regularization algorithm by recursive preconditioning with multidimensional filters defined on a helix (Fomel and Claerbout 2003). The efficient Wilson–Burg spectral factorization constructs a minimum-phase filter suitable for recursive filtering.

We introduce a family of 2D minimum-phase filters for different degrees of tension. The filters are constructed by spectral factorization of the corresponding finite-difference forms. In the case of zero tension (the original minimum-curvature formulation), we obtain a minimum-phase version of the Laplacian filter. The case of infinite tension leads to spectral factorization of the Laplacian and produces the *helical derivative* filter (Claerbout 2003).

The tension filters can be applied not only for data regularization but also for preconditioning in any estimation problems with smooth models. Tomographic velocity estimation is an obvious example of such an application (Woodward *et al.* 1998).

Mathematical theory of splines in tension

The traditional minimum-curvature criterion implies seeking a two-dimensional surface $f(x, y)$ in region D, which corresponds to the minimum of the Laplacian power:

$$\iint_D |\nabla^2 f(x, y)|^2 \, dx \, dy, \tag{4}$$

where ∇^2 denotes the Laplacian operator: $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$.

Alternatively, we can seek $f(x, y)$ as the solution of the biharmonic differential equation,

$$(\nabla^2)^2 f(x, y) = 0. \tag{5}$$

Fung (1965) and Briggs (1974) derived (5) directly from (4) with the help of the variational calculus and Gauss's theorem.

Formula (4) approximates the strain energy of a thin elastic plate (Timoshenko and Woinowsky-Kreiger 1968). Taking tension into account modifies both the energy formula (4) and the corresponding equation (5). Smith and Wessel (1990) suggested the following form of the modified equation:

$$[(1 - \lambda)(\nabla^2)^2 - \lambda(\nabla^2)] f(x, y) = 0, \tag{6}$$

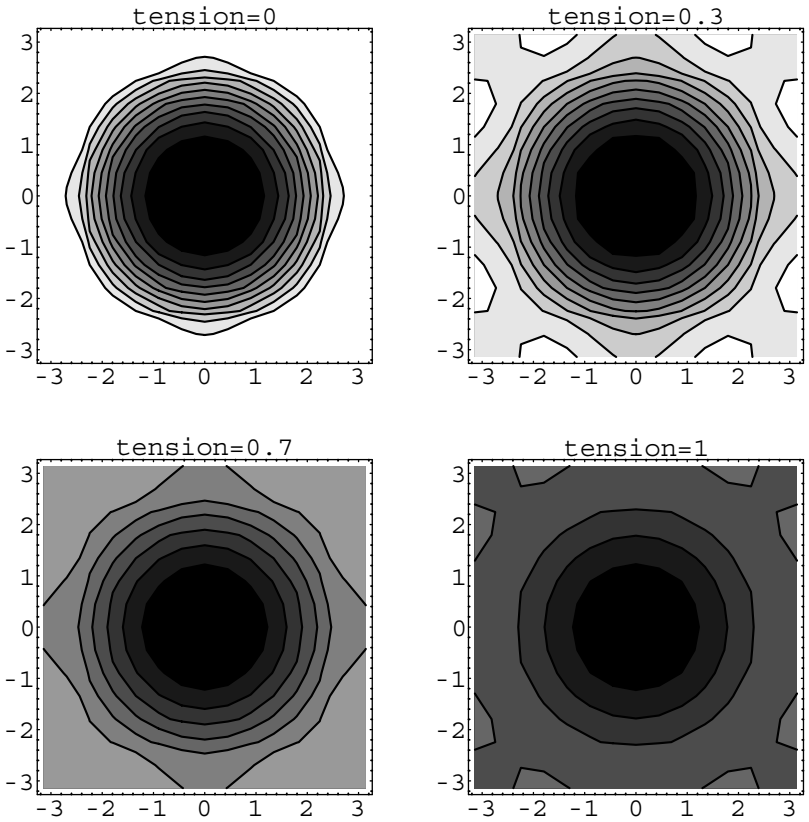

Figure 6 Spectra of the finite-difference splines-in-tension schemes for different values of the tension parameter (contour plots).

where the tension parameter λ ranges from 0 to 1. The corresponding energy functional is

$$\int_D\int [(1-\lambda)|\nabla^2 f(x, y)|^2 + \lambda|\nabla f(x, y)|^2]\, dx\, dy. \quad (7)$$

Zero tension leads to the biharmonic equation (5) and corresponds to the minimum-curvature construction. The case of $\lambda = 1$ corresponds to infinite tension. Although infinite tension is physically impossible, the resulting Laplace equation does have the physical interpretation of a steady-state temperature distribution. An important property of harmonic functions (solutions of the Laplace equation) is that they cannot have local minima and maxima in the free regions. With respect to interpolation, this means that, in the case of $\lambda = 1$, the interpolation surface will be constrained to have its local extrema only at the input data locations.

N. Sleep (2000, pers. comm.) pointed out that if the tension term $\lambda\nabla^2$ is written in the form $\nabla\cdot(\lambda\nabla)$, we can follow an analogy with heat flow and electrostatics and generalize the tension parameter λ to a local function depending on x and y. In a more general form, λ could be a tensor allowing for an anisotropic smoothing in some predefined directions, similarly to the steering-filter method (Clapp *et al.* 1998).

To interpolate an irregular set of data values f_k at points (x_k, y_k), we need to solve (6) under the constraint,

$$f(x_k, y_k) = f_k. \quad (8)$$

We can accelerate the solution by recursive filter preconditioning. If **A** is the discrete filter representation of the differential operator in (6) and we can find a minimum-phase filter **D** whose autocorrelation is equal to **A**, then an appropriate preconditioning operator is a recursive inverse filtering with the filter **D**. The preconditioned formulation of the interpolation problem takes the form of the least-squares system (Claerbout 2003),

$$\mathbf{K}\mathbf{D}^{-1}\mathbf{p} \approx \mathbf{f}_k, \quad (9)$$

where $\mathbf{f}_k$ represents the vector of known data, **K** is the operator of selecting the known data locations, and **p** is the preconditioned variable: $\mathbf{p} = \mathbf{D}\mathbf{f}$. After obtaining an iterative solution of system (9), we reconstruct the model **f** by inverse recursive filtering: $\mathbf{f} = \mathbf{D}^{-1}\mathbf{p}$. Formulating the problem in helical

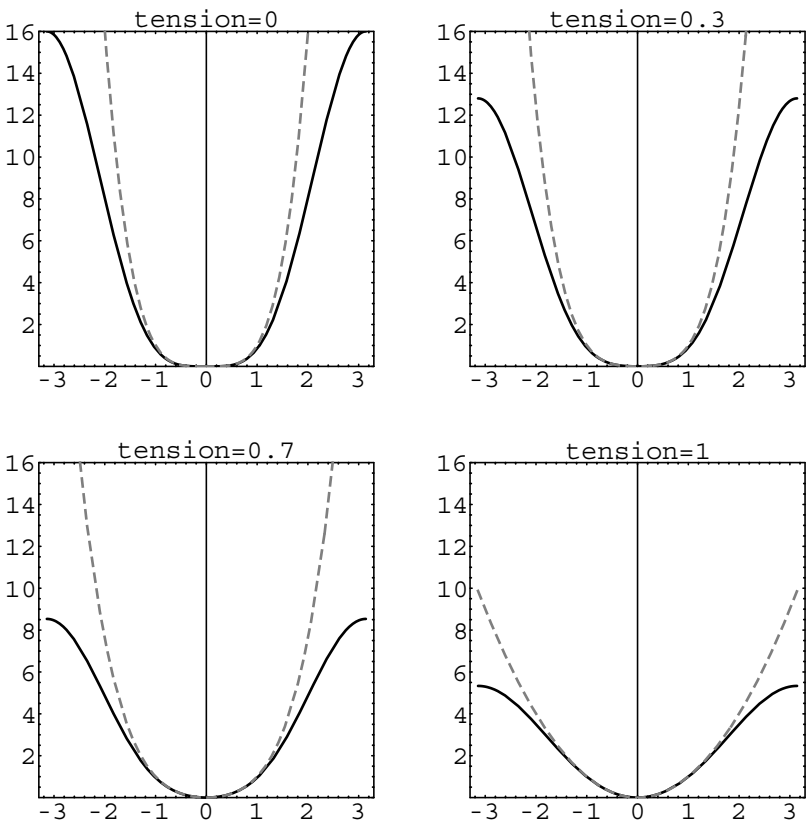

Figure 7 Spectra of the finite-difference splines-in-tension schemes for different values of the tension parameter (cross-section plots). The dashed lines show the exact spectra for continuous operators.

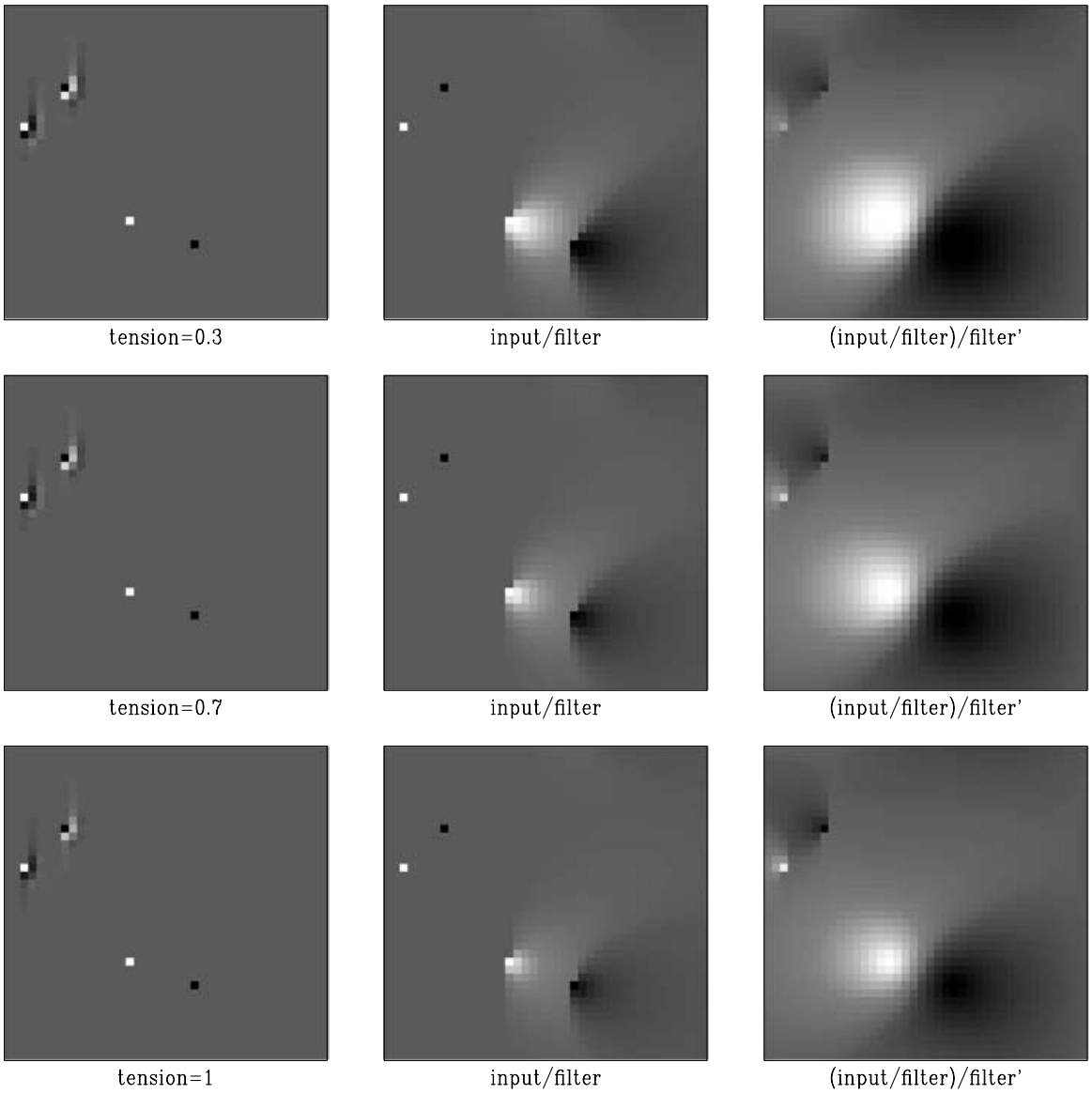

Figure 8 Inverse filtering with the tension filters. The left plots show the inputs composed of filters and spikes. Inverse filtering turns filters into impulses and spikes into inverse filter responses (middle plots). Adjoint filtering creates smooth isotropic shapes (right plots). The tension parameter takes the values 0.3, 0.7 and 1 (from top to bottom). The case of zero tension corresponds to Fig. 3.

coordinates (Mersereau and Dudgeon 1974; Claerbout 1998) enables both the spectral factorization of **A** and the inverse filtering with **D**.

Finite differences and spectral factorization

In the 1D case, one finite-difference representation of the squared Laplacian is as a centred 5-point filter with coefficients (1, −4, 6, −4, 1). On the same grid, the Laplacian operator can be approximated to the same order of accuracy with the filter (1/12, −4/3, 5/2, −4/3, 1/12). Combining the two filters in accordance with (6) and performing the spectral factorization, we can obtain a 3-point minimum-phase filter suitable for inverse filtering. Figure 4 shows a family of 1D minimum-phase filters for different values of the parameter λ. Figure 5 demonstrates the interpolation results obtained with these filters on a simple 1D synthetic. As expected, a small tension value ($\lambda = 0.01$) produces a smooth interpolation,

but creates artificial oscillations in the unconstrained regions around sharp changes in the gradient. The value of $\lambda = 1$ leads to linear interpolation with no extraneous inflections but with discontinuous derivatives. Intermediate values of λ allow us to achieve a compromise: a smooth surface with constrained oscillations.

To design the corresponding filters in two dimensions, we define the finite-difference representation of operator (6) on a 5×5 grid. The filter coefficients are chosen with the help of the Taylor expansion to match the desired spectrum of the operator around the zero spatial frequency. The matching conditions lead to the following set of coefficients for the squared Laplacian:

−1/60	2/5	7/30	2/5	−1/60
2/5	−14/15	−44/15	−14/15	2/5
7/30	−44/15	57/5	−44/15	7/30
2/5	−14/15	−44/15	−14/15	2/5
−1/60	2/5	7/30	2/5	−1/60

$$= 1/60 \begin{array}{|c|c|c|c|c|} \hline -1 & 24 & 14 & 24 & -1 \\ \hline 24 & -56 & -176 & -56 & 24 \\ \hline 14 & -176 & 684 & -176 & 14 \\ \hline 24 & -56 & -176 & -56 & 24 \\ \hline -1 & 24 & 14 & 24 & -1 \\ \hline \end{array}$$

The Laplacian representation with the same order of accuracy has the coefficients:

−1/360	2/45	0	2/45	−1/360
2/45	−14/45	−4/5	−14/45	2/45
0	−4/5	41/10	−4/5	0
2/45	−14/45	−4/5	−14/45	2/45
−1/360	2/45	0	2/45	−1/360

$$= 1/360 \begin{array}{|c|c|c|c|c|} \hline -1 & 16 & 0 & 16 & -1 \\ \hline 16 & -112 & -288 & -112 & 16 \\ \hline 0 & -288 & 1476 & -288 & 0 \\ \hline 16 & -112 & -288 & -112 & 16 \\ \hline -1 & 16 & 0 & 16 & -1 \\ \hline \end{array}$$

For the sake of simplicity, we assumed equal spacing in the x- and y-directions. The coefficients can be easily adjusted for anisotropic spacing. Figures 6 and 7 show the spectra of the finite-difference representations of operator (6) for different values of the tension parameter. The finite-difference spectra appear to be fairly isotropic (independent of angle in polar coordinates). They match the exact expressions at low frequencies.

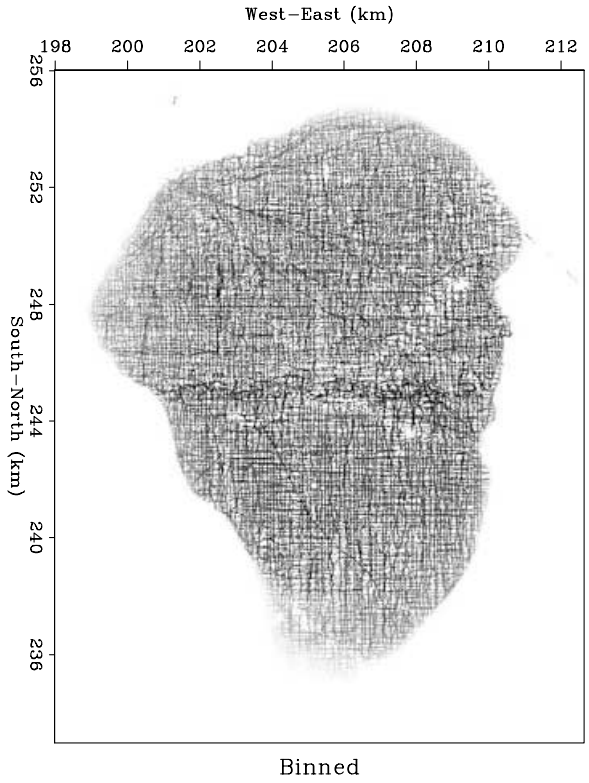

Figure 9 The Sea of Galilee data set after nearest-neighbour binning. The binned data are used as input for the missing data interpolation program.

Regarding the finite-difference operators as 2D autocorrelations and applying the Wilson–Burg method of spectral factorization, we obtain 2D minimum-phase filters suitable for inverse filtering. The exact filters contain many coefficients, which rapidly decrease in magnitude at a distance from the first coefficient. For reasons of efficiency, it is advisable to restrict the shape of the filter so that it contains only the significant coefficients. Keeping all the coefficients that are 1000 times smaller in magnitude than the leading coefficient creates a 53-point filter for $\lambda = 0$ and a 35-point filter for $\lambda = 1$, with intermediate filter lengths for intermediate values of λ. Keeping only the coefficients that are 200 times smaller than the leading coefficient, we obtain 25- and 16-point filters for $\lambda = 0$ and $\lambda = 1$, respectively. The restricted filters do not factorize the autocorrelation exactly but provide an effective approximation of the exact factors. As outputs of the Wilson–Burg spectral factorization process, they obey the minimum-phase condition.

Figure 8 shows the 2D filters for different values of λ and illustrates inverse recursive filtering, which is the essence of the helix method (Claerbout 1998). The case of $\lambda = 1$ leads

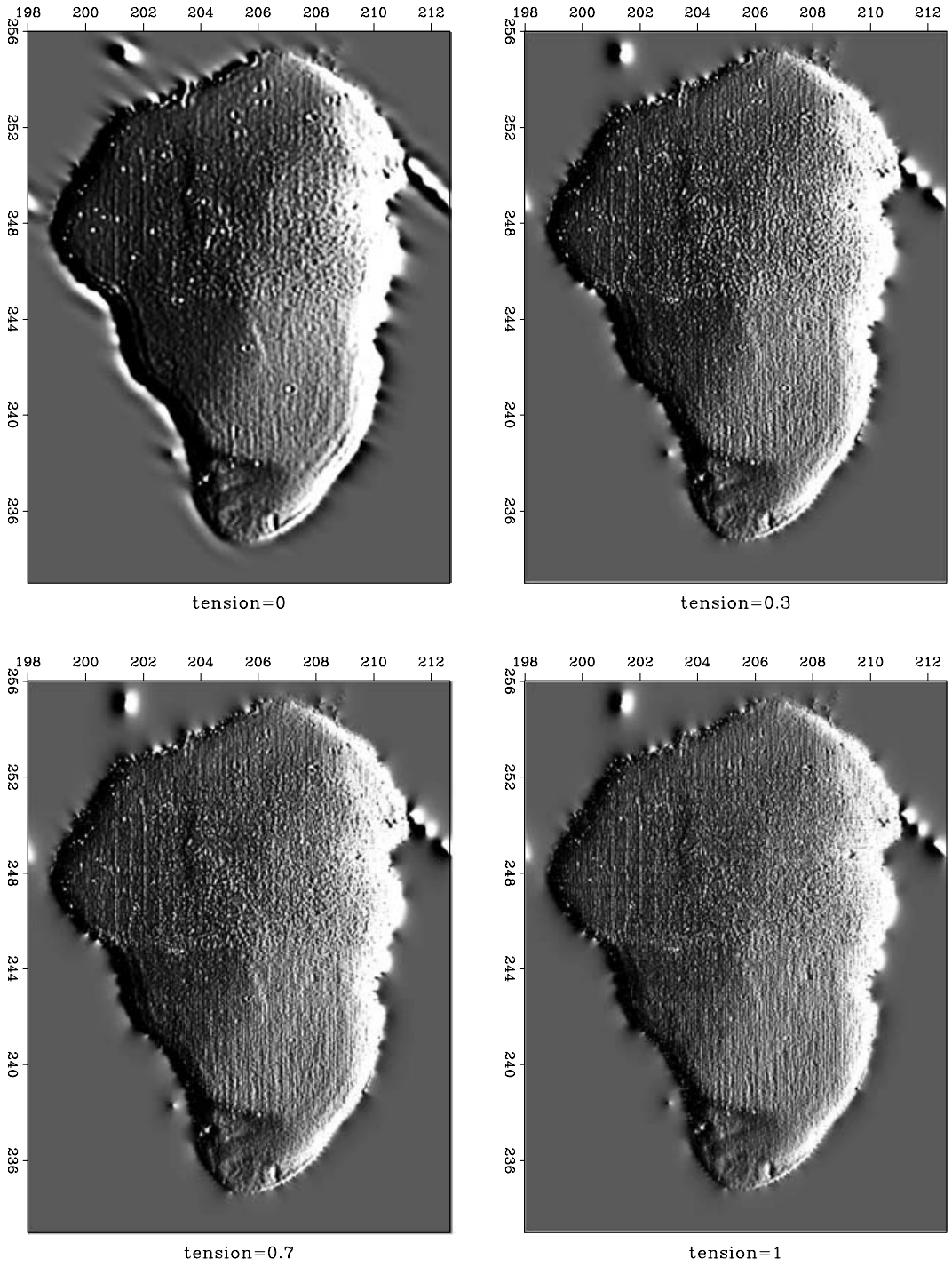

Figure 10 The Sea of Galilee data set after missing data interpolation with helical preconditioning. Different plots correspond to different values of the tension parameter. An east–west derivative filter was applied to illuminate the surface.

to the filter known as a *helix derivative* (Claerbout 2003). The filter values are spread mostly in two columns. The other boundary case ($\lambda = 0$) leads to a three-column filter, which serves as the minimum-phase version of the Laplacian. This filter is similar to the one shown in Fig. 3. As expected from the theory, the inverse impulse response of this filter is noticeably smoother and wider than the inverse response of the helix derivative. Filters corresponding to intermediate values

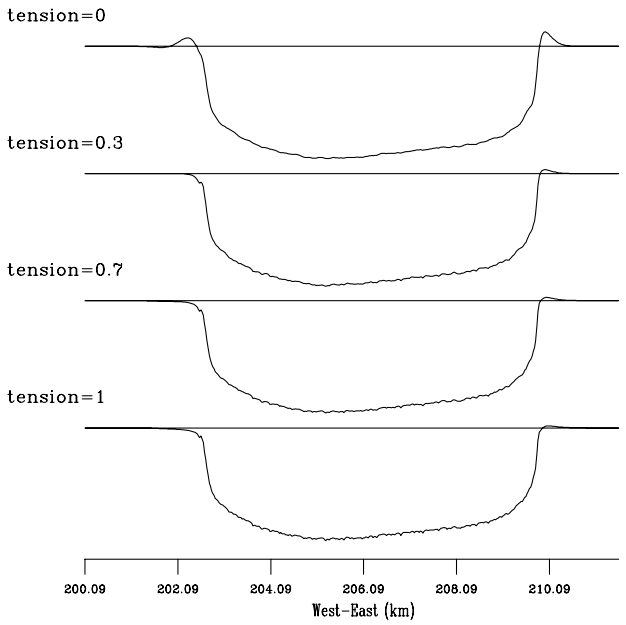

Figure 11 Cross-sections of the Sea of Galilee data set after missing-data interpolation with helical preconditioning. Different plots correspond to different values of the tension parameter.

of λ exhibit intermediate properties. Theoretically, the inverse impulse response of the filter corresponds to the Green's function of (6). The theoretical Green's function for the case of $\lambda = 1$ is

$$G = \frac{1}{2\pi} \ln r, \tag{10}$$

where r is the distance from the impulse: $r = \sqrt{(x - x_k)^2 + (y - y_k)}$. In the case of $\lambda = 0$, the Green's function is smoother at the origin:

$$G = \frac{1}{8\pi} r^2 \ln r. \tag{11}$$

The expression for the theoretical Green's function for an arbitrary value of λ is unknown (Mitášová and Mitáš (1993) derived an analytical Green's function for a different model of tension splines using special functions), but we can assume that its smoothness lies between the two boundary conditions.

In the next subsection, we illustrate an application of helical inverse filtering to a 2D interpolation problem.

Regularization example

We chose an environmental data set (Claerbout 2003) for a simple illustration of smooth data regularization. The data were collected on a bottom-sounding survey of the Sea of Galilee in Israel (Ben-Avraham *et al.* 1990). The data contain a number of noisy, erroneous and inconsistent measurements, which present a challenge for the traditional estimation methods (Fomel and Claerbout 1995).

Figure 9 shows the data after a nearest-neighbour binning to a regular grid. The data were then passed to an interpolation program to fill the empty bins. The results (for different values of λ) are shown in Figs 10 and 11. Interpolation with the minimum-phase Laplacian ($\lambda = 0$) creates a relatively smooth interpolation surface but plants artificial 'hills' around the edge of the sea. This effect is caused by large gradient changes and is similar to the sidelobe effect in the 1D example (Fig. 5). It is clearly seen in the cross-section plots in Fig. 11. The abrupt gradient change is a typical case of a shelf break. It is caused by a combination of sedimentation and active rifting. Interpolation with the helix derivative ($\lambda = 1$) is free from the sidelobe artefacts, but it also produces an undesirable non-smooth behaviour in the middle part of the image. As in the 1D example, intermediate tension allows us to achieve a compromise: smooth interpolation in the middle and constrained behaviour at the sides of the sea bottom.

CONCLUSIONS

The Wilson–Burg spectral factorization method, presented here, enables stable recursive filters to be constructed. The method appears to have attractive computational properties and can be significantly more efficient than alternative spectral factorization algorithms. It is particularly suitable for the multidimensional case, where recursive filtering is enabled by the helix transform.

We have illustrated an application of the Wilson–Burg method for efficient smooth data regularization. A constrained approach to smooth data regularization leads to splines in tension. The constraint is embedded in a user-specified tension parameter. The two boundary values of tension correspond to cubic and linear interpolation. By applying the method of spectral factorization on a helix, we have been able to define a family of 2D minimum-phase filters, which correspond to the spline interpolation problem with different values of tension. We have used these filters for accelerating data-regularization problems with smooth surfaces by recursive preconditioning. In general, they are applicable for preconditioning acceleration in various estimation problems with smooth models.

ACKNOWLEDGEMENTS

This paper owes a great deal to John P. Burg. We thank him and Francis Muir for many useful and stimulating discussions. S.F. also thanks Jim Berryman for explaining the variational derivation of the biharmonic and tension-spline equations. Ralf Ferber and Ali Özbek provided helpful reviews.

The financial support for this work was provided by the sponsors of the Stanford Exploration Project.

REFERENCES

Bauer F.L. 1955. Ein direktes Iterations verfahren zur Hurwitz-Zerlegung eines Polynoms. *Archiv für Elektronik und Übertragungstechnik* **9**, 285–290.

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.

de Boor C. 1978. *A Practical Guide to Splines*. Springer-Verlag, Inc.

Briggs I.C. 1974. Machine contouring using minimum curvature. *Geophysics* **39**, 39–48.

Claerbout J.F. 1976. *Fundamentals of Geophysical Data Processing*. Blackwell Publishing.

Claerbout J. 1998. Multidimensional recursive filters via a helix. *Geophysics* **63**, 1532–1541.

Claerbout J. 2003. *Image estimation by example: Geophysical soundings image construction*. Stanford Exploration Project, http://sepwww.stanford.edu/sep/prof/.

Clapp R.G., Biondi B.L., Fomel S.B. and Claerbout J.F. 1998. Regularizing velocity estimation using geologic dip information. 68th SEG Meeting, New Orleans, USA, Expanded Abstracts, 1851–1854.

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

Fomel S. and Claerbout J. 2003. Multidimensional recursive filter preconditioning in geophysical estimation problems. *Geophysics* **68**, 577–588.

Fung Y.C. 1965. *Foundations of Solid Mechanics*. Prentice–Hall, Inc.

Goodman T.N.T., Micchelli C.A., Rodriguez G. and Seatzu S. 1997. Spectral factorization of Laurent polynomials. *Advances in Computational Mathematics* **7**, 429–445.

Kolmogoroff A.N. 1939. Sur l'interpolation et extrapolation des suites stationnaires. *Comptes Rendus de l'Académie des Sciences* **208**, 2043–2045.

Laurie D.P. 1980. Efficient implementation of Wilson's algorithm for factorizing a self-reciprocal polynomial. *BIT* **20**, 257–259.

Mersereau R.M. and Dudgeon D.E. 1974. The representation of two-dimensional sequences as one-dimensional sequences. *IEEE Transactions on Acoustics, Speech and Signal Processing* **ASSP-22**, 320–325.

Mitášová H. and Mitáš L. 1993. Interpolation by regularized splines with tension: I. Theory and implementation. *Mathematical Geology* **25**, 641–655.

Oppenheim A.V. and Shafer R.W. 1989. *Discrete-Time Signal Processing*. Prentice–Hall, Inc.

Ozdemir A., Ozbek A., Ferber R. and Zerouk K. 1999a. *f-xy* projection filtering using helical transformation. 69th SEG Meeting, Houston, USA, Expanded Abstracts, 1231–1234.

Ozdemir A.K., Ozbek A., Ferber R. and Zerouk K. 1999b. *f-xy* projection via helical transformation – application to noise attenuation. 61st EAGE Conference, Helsinki, Finland, Extended Abstracts, Session 6039.

Rickett J. 2000. Efficient 3D wavefield extrapolation with Fourier finite-differences and helical boundary conditions. 62nd EAGE Conference, Glasgow, Scotland, Extended Abstracts, Session P0145.

Rickett J. and Claerbout J. 1999. Acoustic daylight imaging via spectral factorization: Helioseismology and reservoir monitoring. 69th SEG Meeting, Houston, USA, Expanded Abstracts, 1675–1678.

Rickett J., Claerbout J. and Fomel S.B. 1998. Implicit 3D depth migration by wavefield extrapolation with helical boundary conditions. 68th SEG Meeting, New Orleans, USA, Expanded Abstracts, 1124–1127.

Rickett J., Guitton A. and Gratwick D. 2001. Adaptive multiple subtraction with non-stationary helical shaping filters. 63rd EAGE Conference, Amsterdam, The Netherlands, Extended Abstracts, Session P167.

Sandwell D.T. 1987. Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data. *Geophysical Research Letters* **14**, 139–142.

Sayed A.H. and Kailath T. 2001. A survey of spectral factorization methods. *Numerical Linear Algebra Applications* **8**, 467–496.

Schur I. 1917. Über potenzreihen die im inheren des einheitskreises beschränkt sind. *Journal für die Reine und Angewandte Mathematik* **147**, 205–232.

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**, 293–305.

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.

Timoshenko S. and Woinowsky-Krieger S. 1968. *Theory of Plates and Shells*. McGraw–Hill Book Co.

Wilson G. 1969. Factorization of the covariance generating function of a pure moving average process. *SIAM Journal of Numerical Analysis* **6**, 1–7.

Woodward M.J., Farmer P., Nichols D. and Charles S. 1998. Automated 3D tomographic velocity analysis of residual moveout in prestack depth migrated common image point gathers. 68th SEG Meeting, New Orleans, USA, Expanded Abstracts, 1218–1221.

Zhang G. and Shan G. 2001. Helical scheme for 2-D prestack migration based on double-square-root equation. 71st SEG Meeting, San Antonio, USA, Expanded Abstracts, 1057–1060.

Zhang G., Zhang Y. and Zhou H. 2000. Helical finite-difference schemes for 3D depth migration. 70th SEG Meeting, Calgary, Canada, Expanded Abstracts, 862–865.